Latent Force Models: Combining Probabilistic and Mechanistic Modelling

Neil D. Lawrence

(work with Magnus Rattray, **Mauricio Álvarez**, Pei Gao, Antti Honkela, David Luengo, Guido Sanguinetti, Michalis Titsias, Jennifer Withers)

University of Sheffield

University of Sheffield
Seminar at University of Oxford
Robotics Research Group Seminar

13th February 2012

Outline

Outline

Styles of Machine Learning

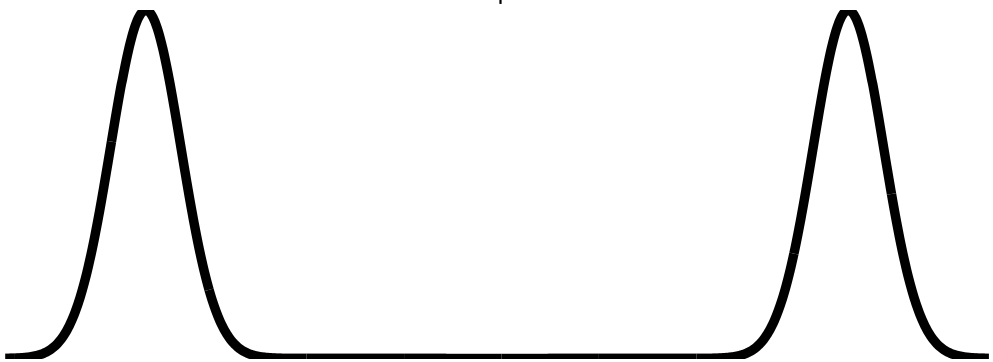
Background: interpolation is easy, extrapolation is hard

- 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*



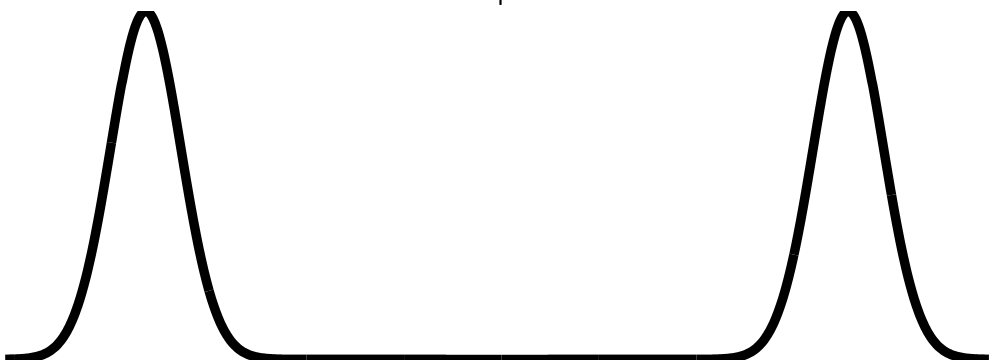
General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

mechanistic modeling



General Approach

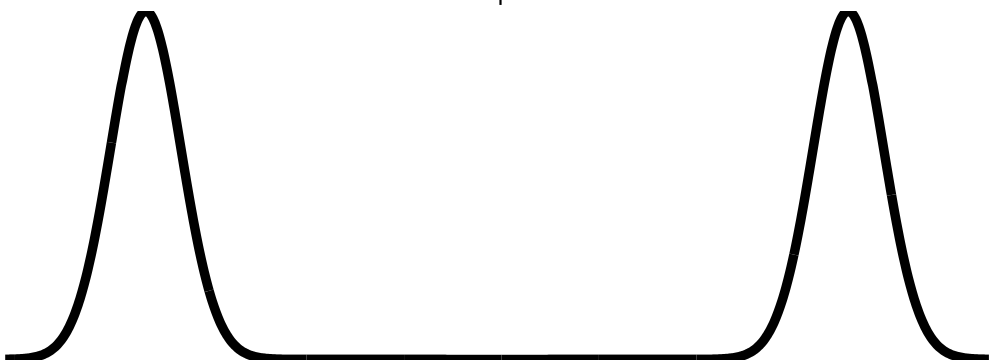
Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

mechanistic modeling

impose physical laws



General Approach

Broadly Speaking: Two approaches to modeling

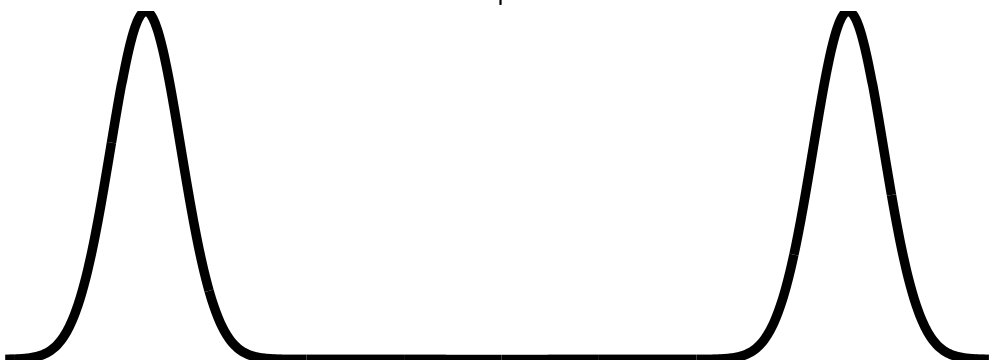
data modeling

let the data "speak"

data driven

mechanistic modeling

impose physical laws



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

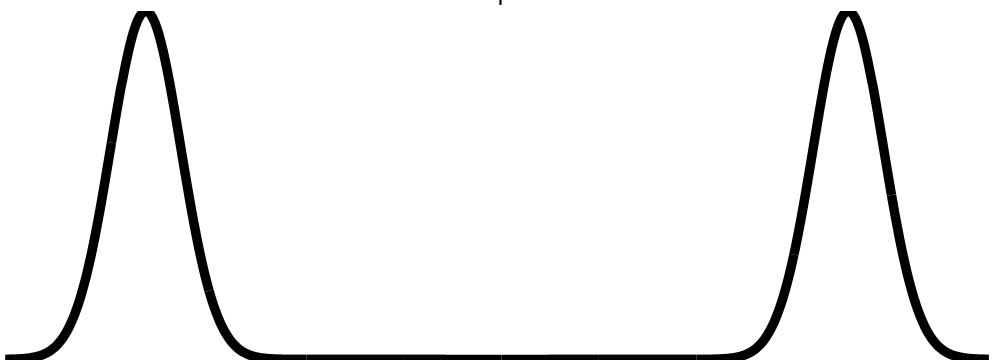
let the data "speak"

data driven

mechanistic modeling

impose physical laws

knowledge driven



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

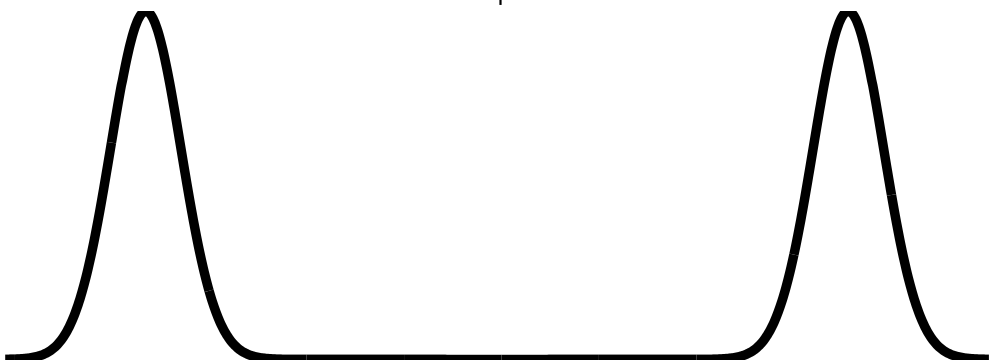
data driven

adaptive models

mechanistic modeling

impose physical laws

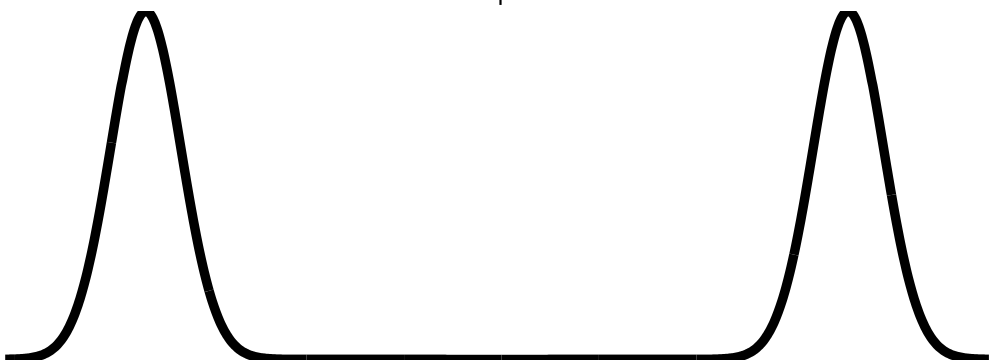
knowledge driven



General Approach

Broadly Speaking: Two approaches to modeling

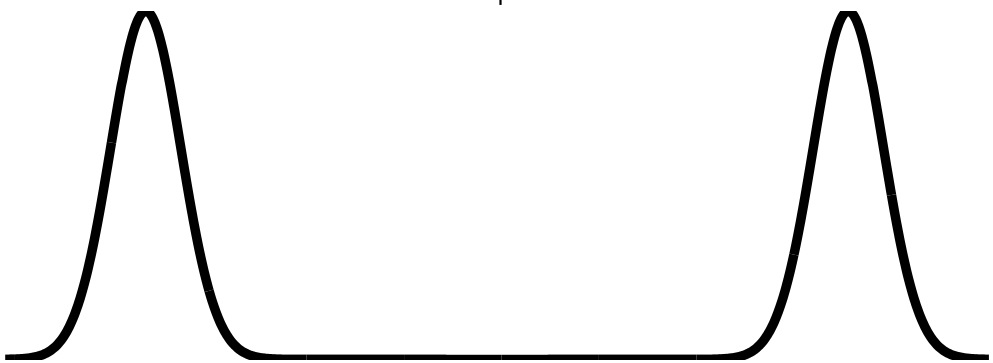
data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations



General Approach

Broadly Speaking: Two approaches to modeling

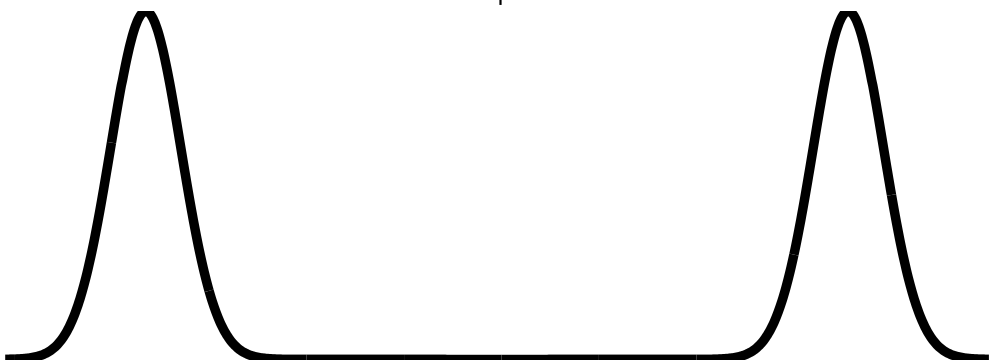
data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	



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

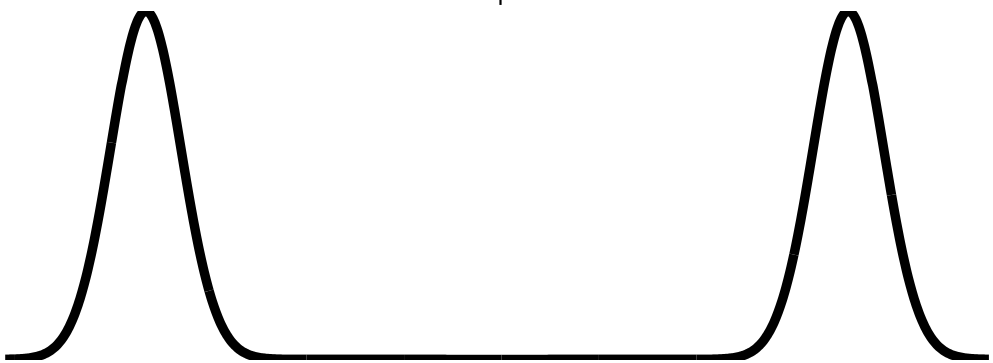


General Approach

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

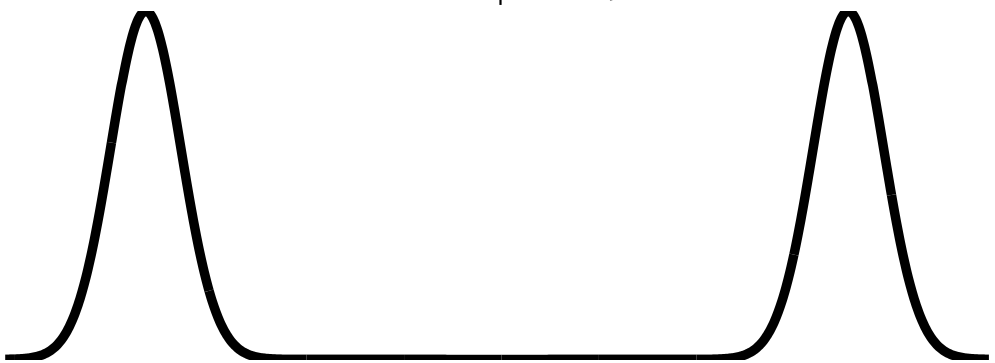
mechanistic modeling

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.

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,

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

Joint Gaussian Process

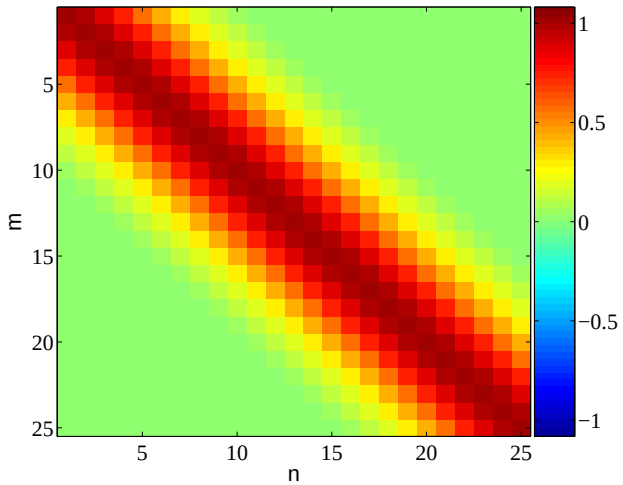
- Given the covariance functions for $\{f_i(t)\}$ we have an implied covariance function across all $\{x_i(t)\}$—(ML: semi-parametric latent factor model (Teh et al., 2005), Geostatistics: linear model of coregionalization).
- Rauch-Tung-Striebel smoother has been preferred
  - linear computational complexity 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 Process: Exponentiated Quadratic Covariance

- Take, for example, exponentiated quadratic form for covariance.

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

- Gaussian process over latent functions.



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: $\mathbf{W} = \mathbf{S}\mathbf{D}^{-1}$.

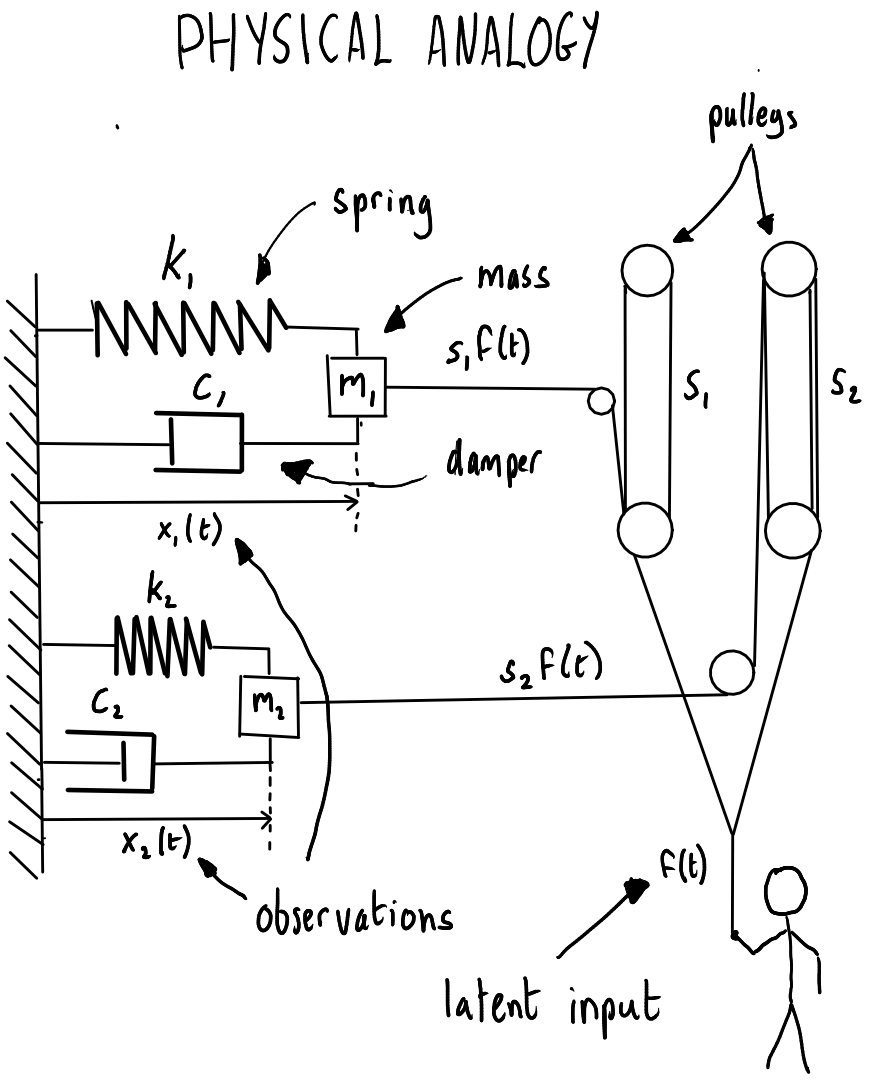
Extend Model

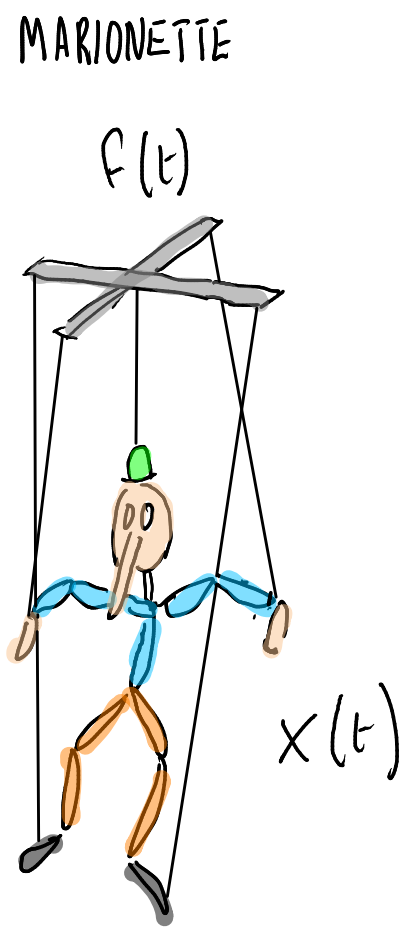
- Add a damper and give the system mass.

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

- 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

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

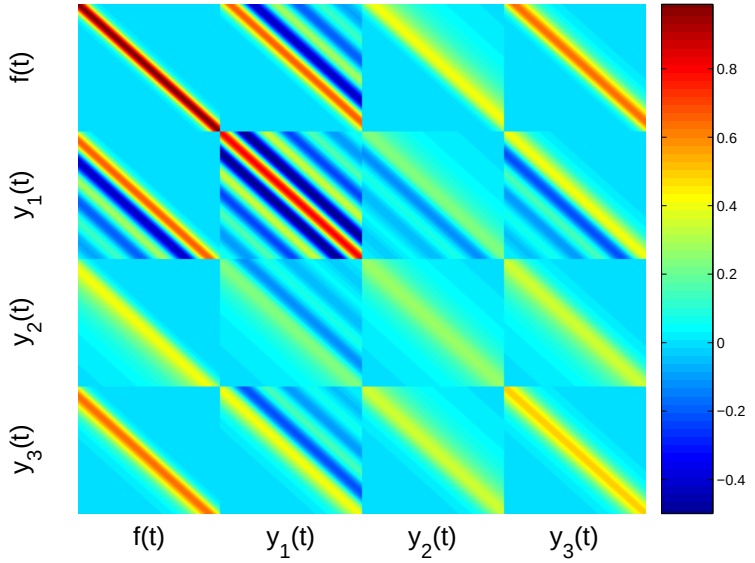
Covariance for ODE Model

- 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



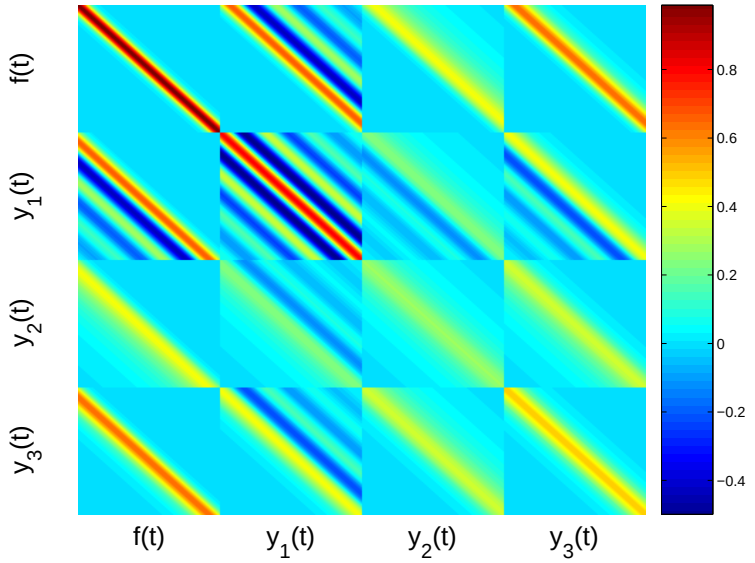
Covariance for ODE Model

- 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



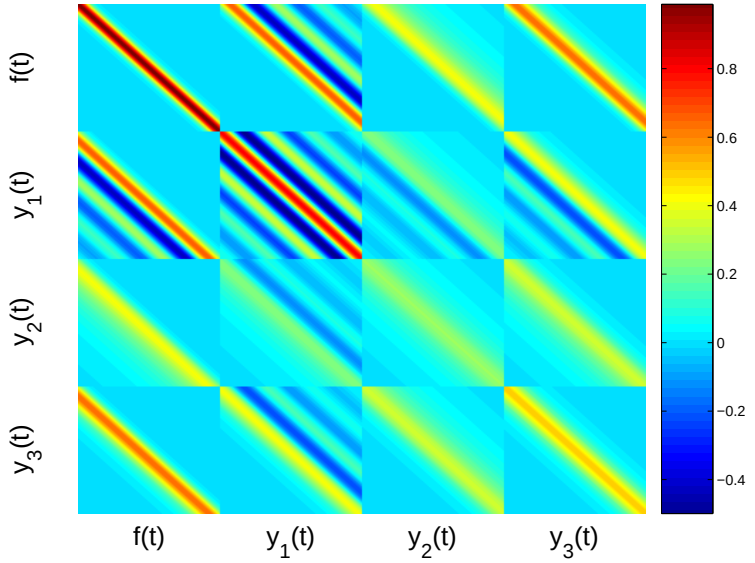
Covariance for ODE Model

- 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`

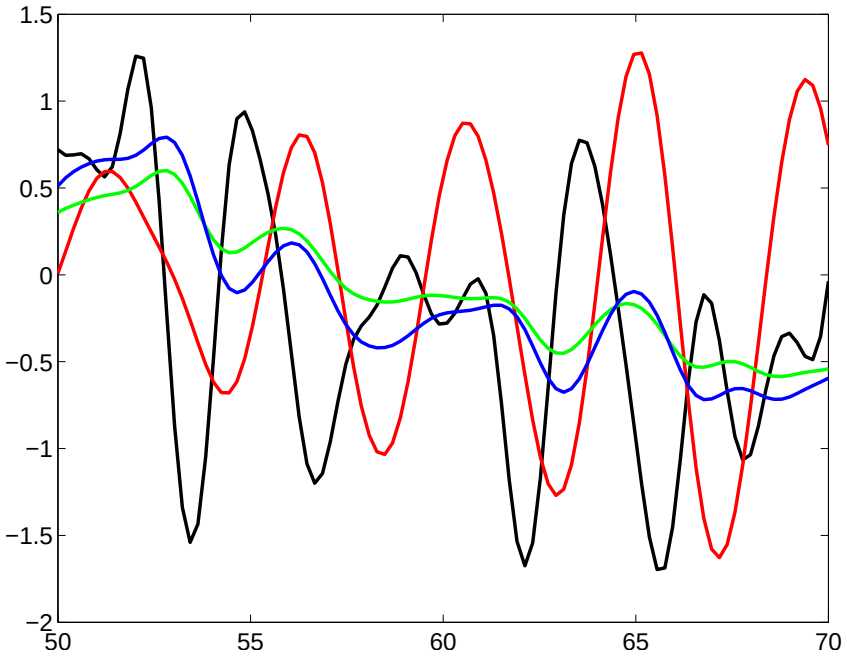


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

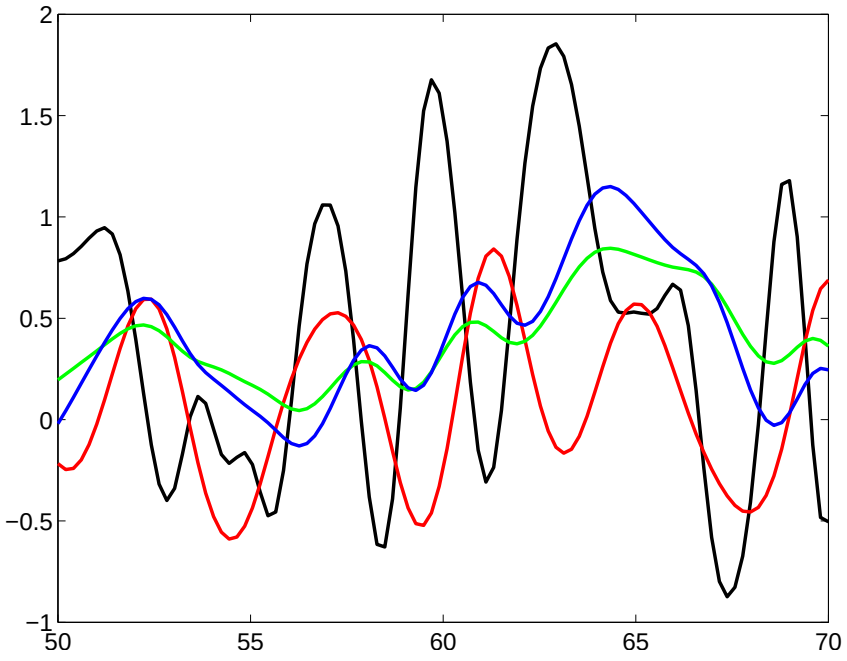


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

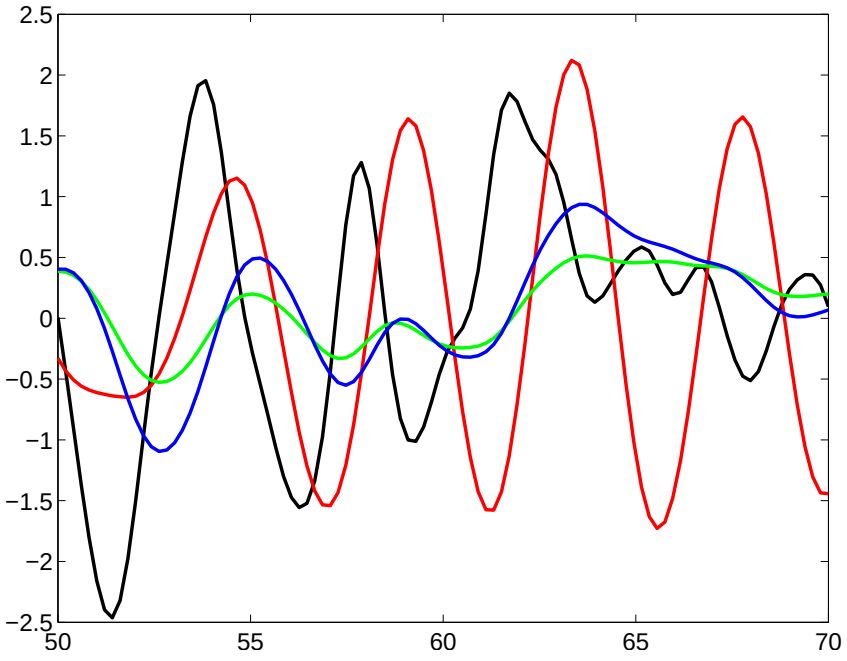


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

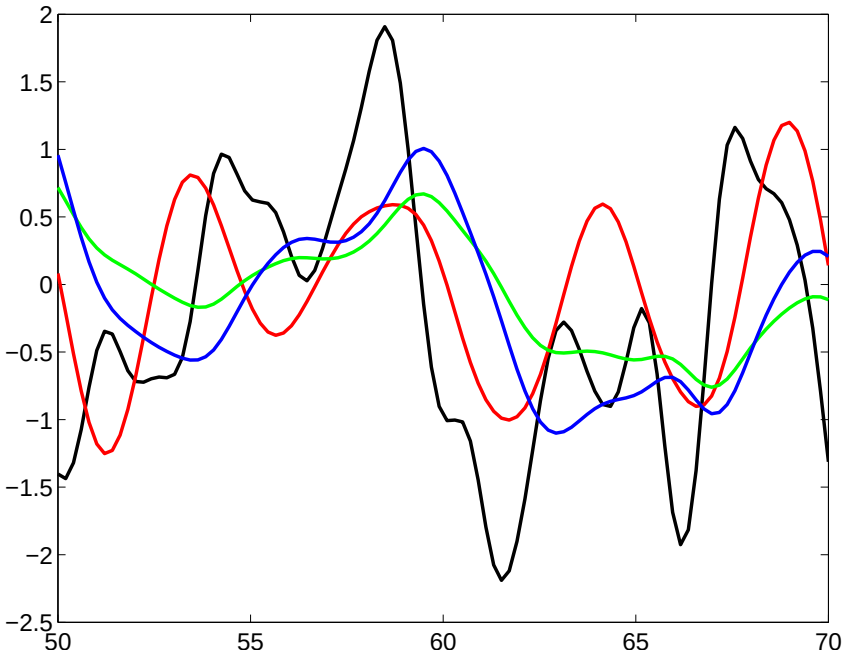


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

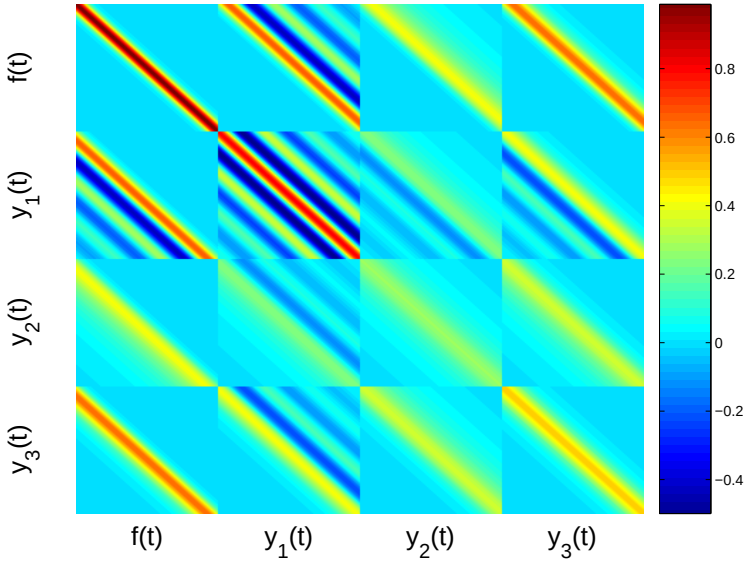
Covariance for ODE

- 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



Outline

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009, 2011a)

- 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, 2011a)

- 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, 2011a)

- 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, 2011a)

- 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

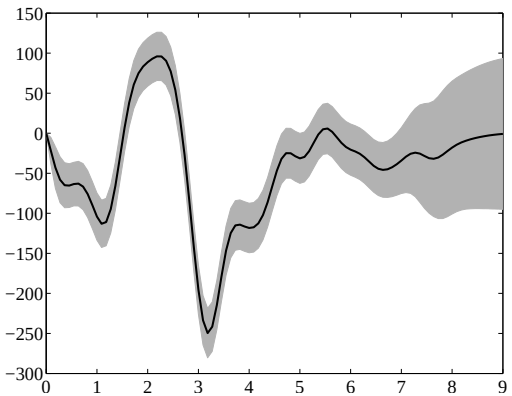
- Model left arm only.
- 3 balancing motions (18, 19, 20) from subject 49.
- 18 and 19 are similar, 20 contains more dramatic movements.
- Train on 18 and 19 and testing on 20
- Data was down-sampled by 32 (from 120 fps).
- Reconstruct motion of left arm for 20 given other movements.
- Compare with GP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

Mocap Results

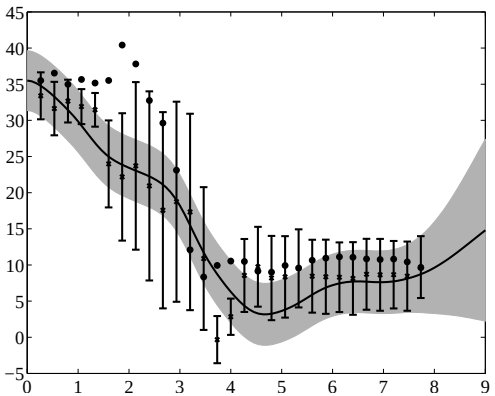
Table: Root mean squared (RMS) angle error for prediction of the left arm's configuration in the motion capture data. Prediction with the latent force model outperforms the prediction with regression for all apart from the radius's angle.

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

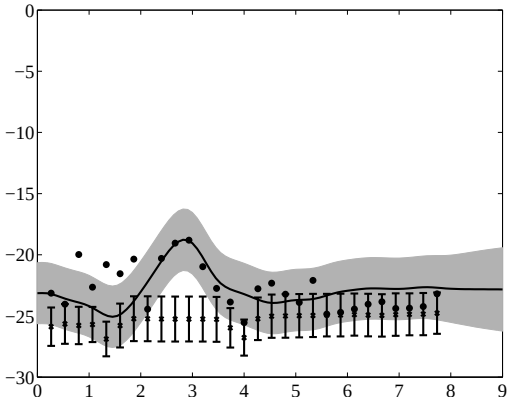
Mocap Results II



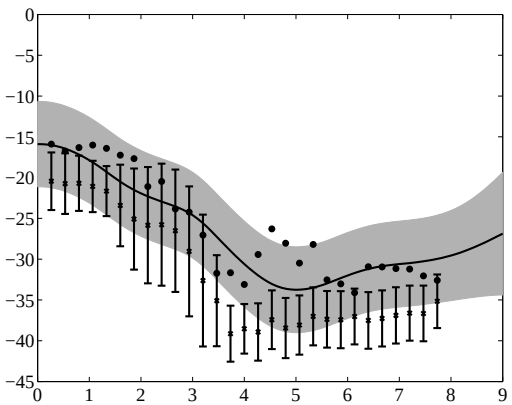
(a) Inferred Latent Force



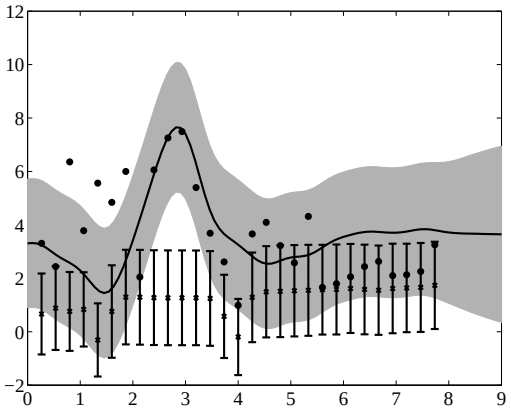
(b) Wrist



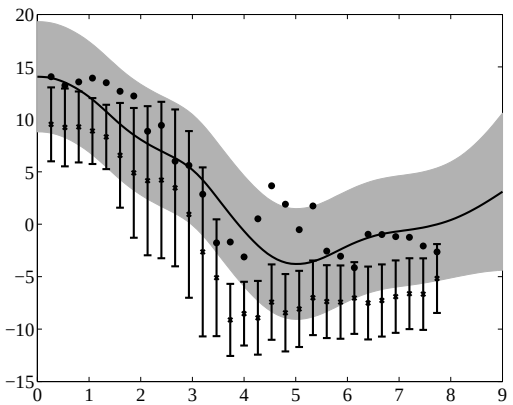
(c) Hand X Rotation



(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).

Motion Capture Experiments

- Data set is from the CMU motion capture data base[1].
- Two different types of movements: golf-swing and walking.
- Train on a subset of motions for each movement and test on a different subset.
- This assesses the model's ability to extrapolate.
- For testing: condition on three angles associated to the root nodes and first five and last five frames of the motion.
- Golf-swing use leave one out cross validation on four motions.
- For the walking train on 4 motions and validate on 8 motions.

[1]The CMU Graphics Lab Motion Capture Database was created with funding from NSF EIA-0196217 and is available at `http://mocap.cs.cmu.edu`.

Motion Capture Results

Table: RMSE and R^2 (explained variance) for golf swing and walking

Movement	**Method**	**RMSE**	**R^2 (%)**
Golf swing	IND GP	21.55 ± 2.35	30.99 ± 9.67
	MTGP	21.19 ± 2.18	45.59 ± 7.86
	SLFM	21.52 ± 1.93	49.32 ± 3.03
	LFM	$\mathbf{18.09 \pm 1.30}$	$\mathbf{72.25 \pm 3.08}$
Walking	IND GP	8.03 ± 2.55	30.55 ± 10.64
	MTGP	7.75 ± 2.05	37.77 ± 4.53
	SLFM	7.81 ± 2.00	36.84 ± 4.26
	LFM	$\mathbf{7.23 \pm 2.18}$	$\mathbf{48.15 \pm 5.66}$

Outline

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?

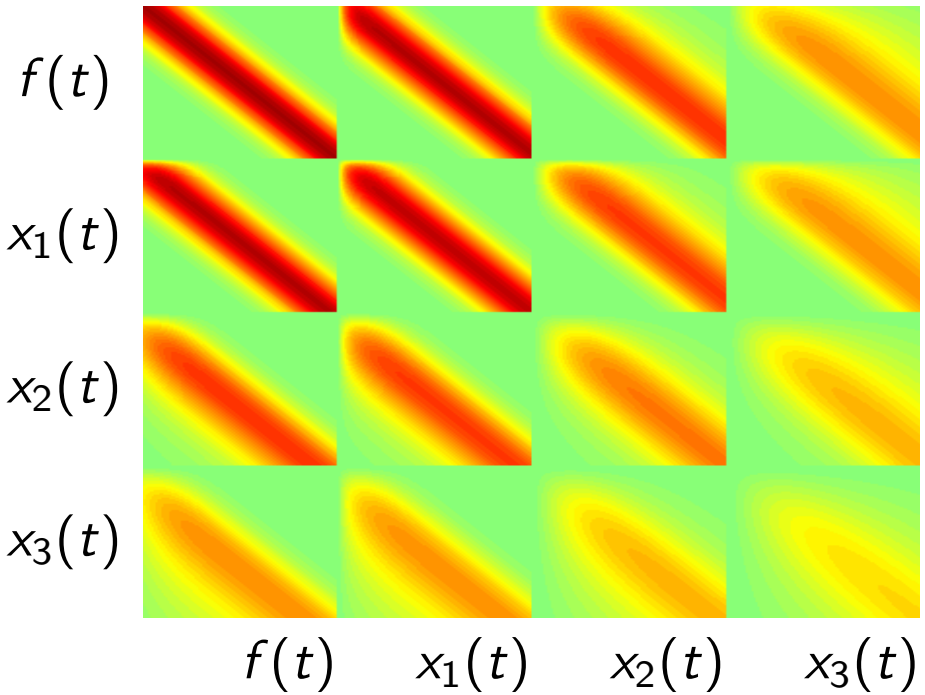
Covariance for Transcription Model

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



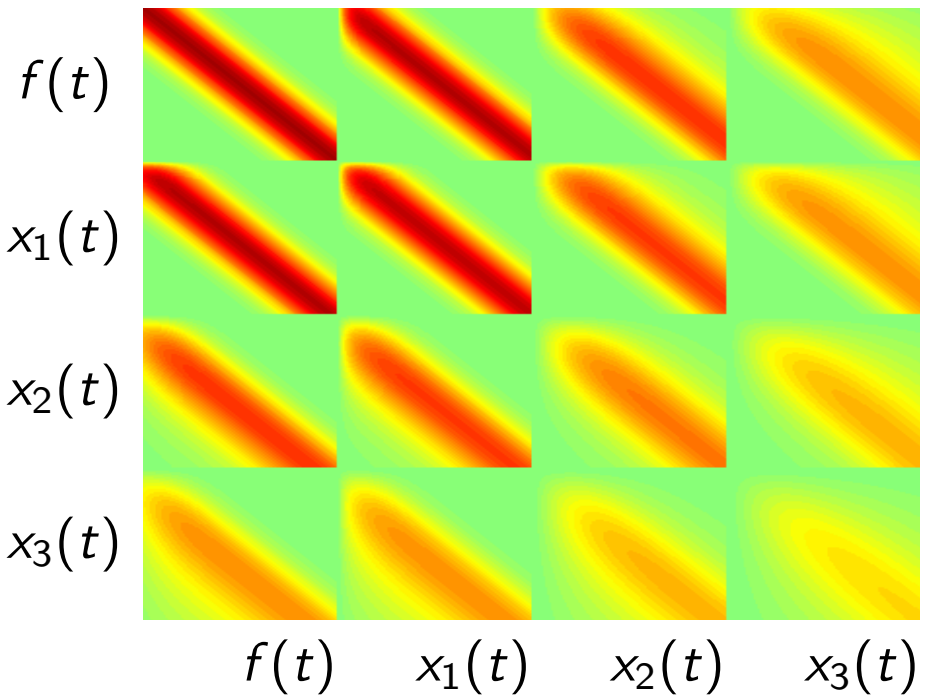
Covariance for Transcription Model

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



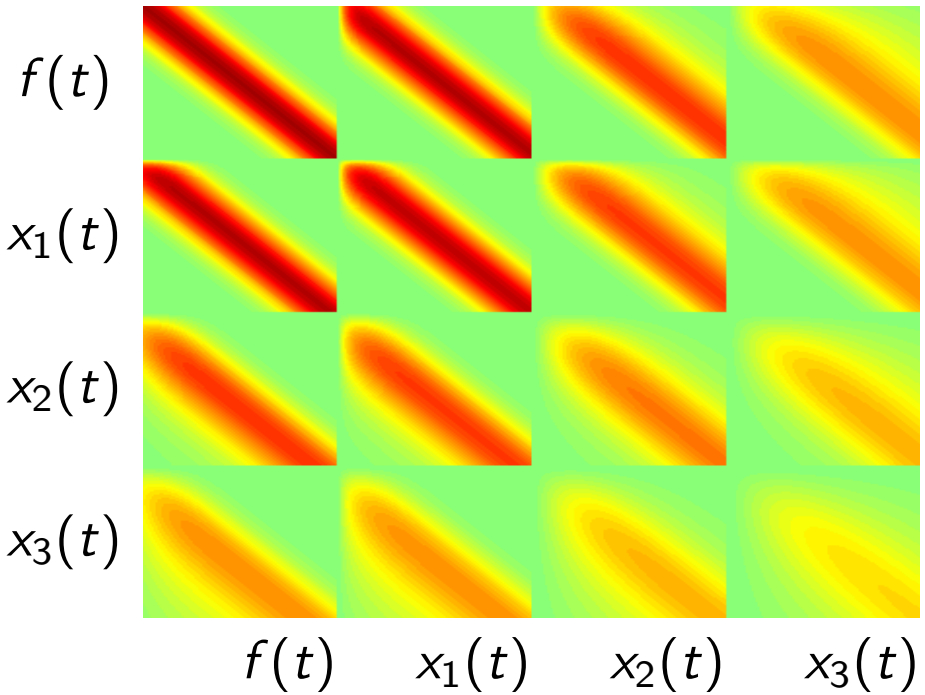
Covariance for Transcription Model

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`

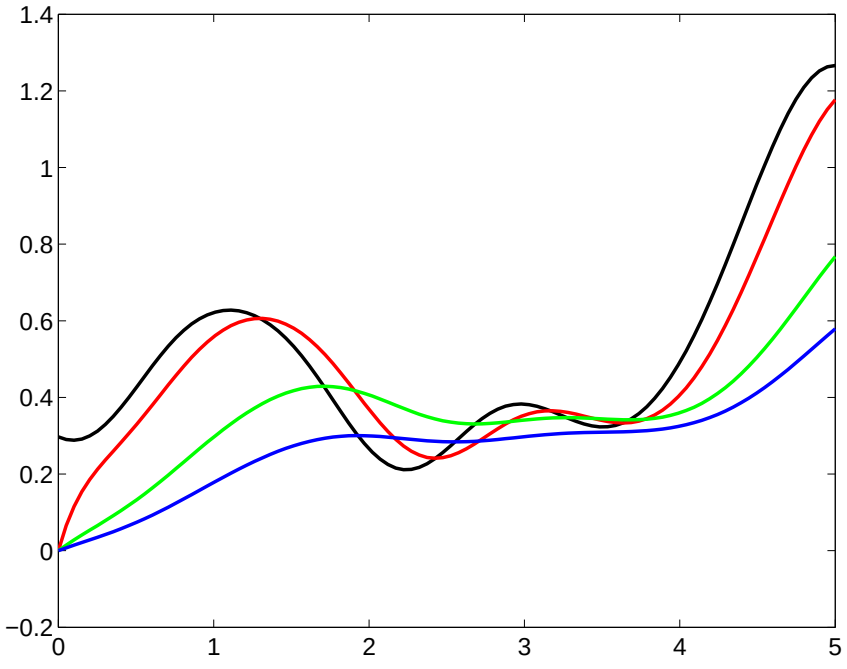


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

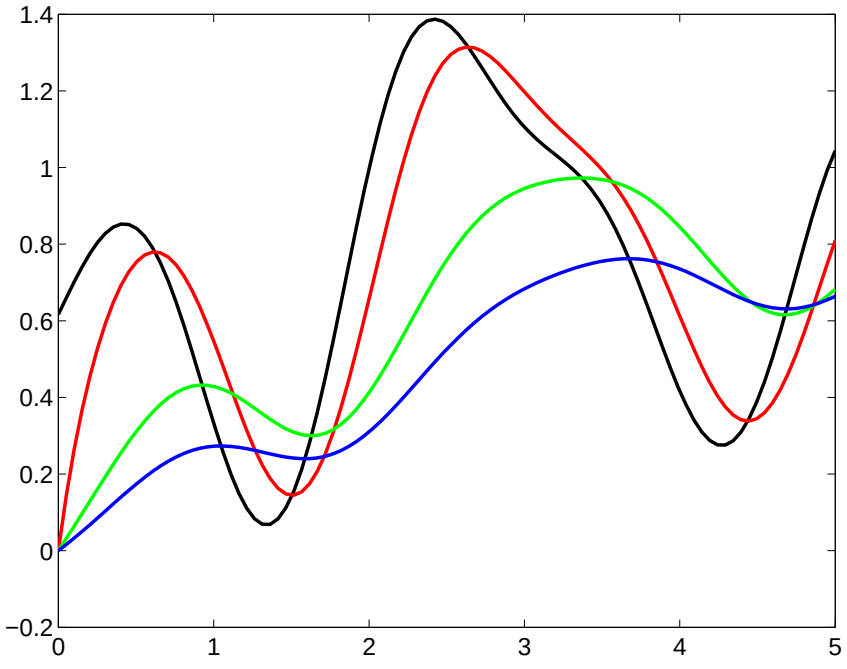


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

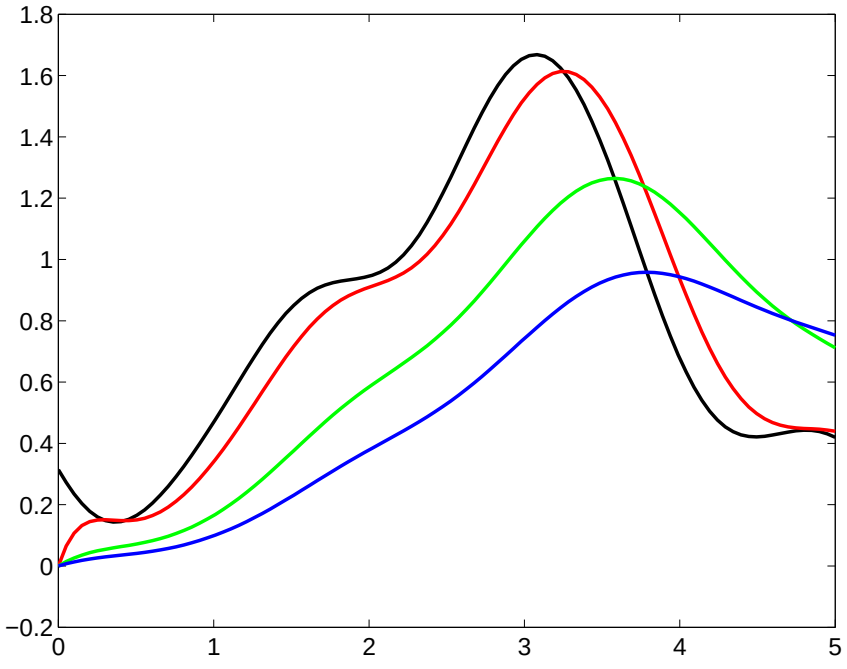


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

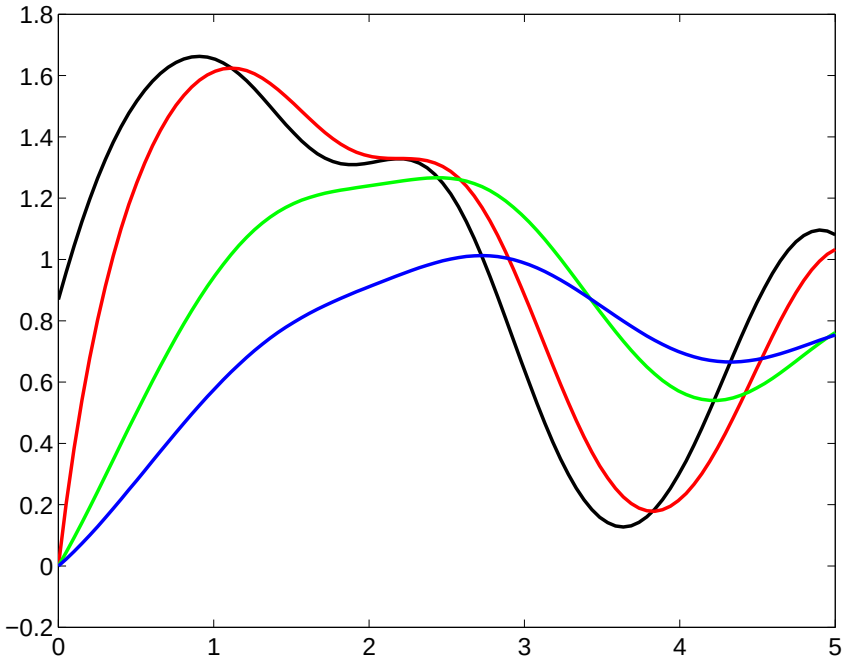
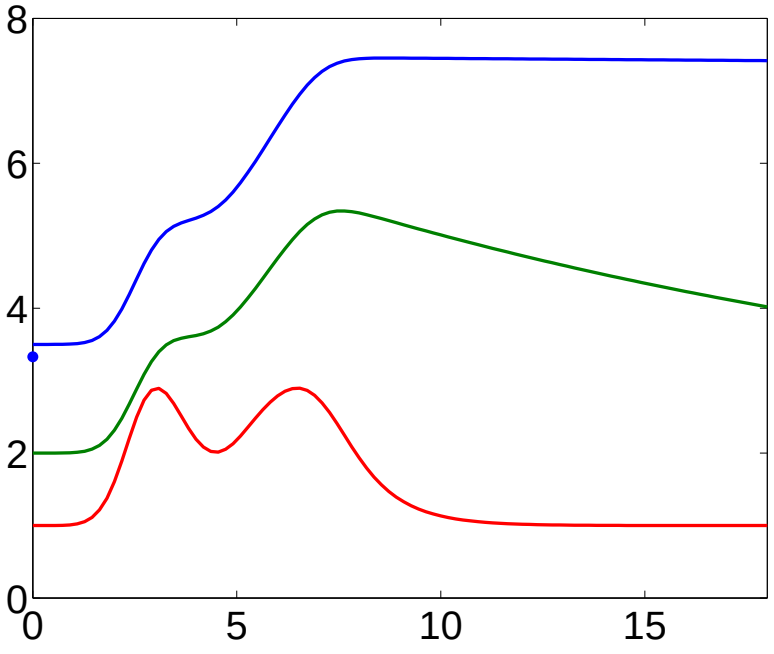


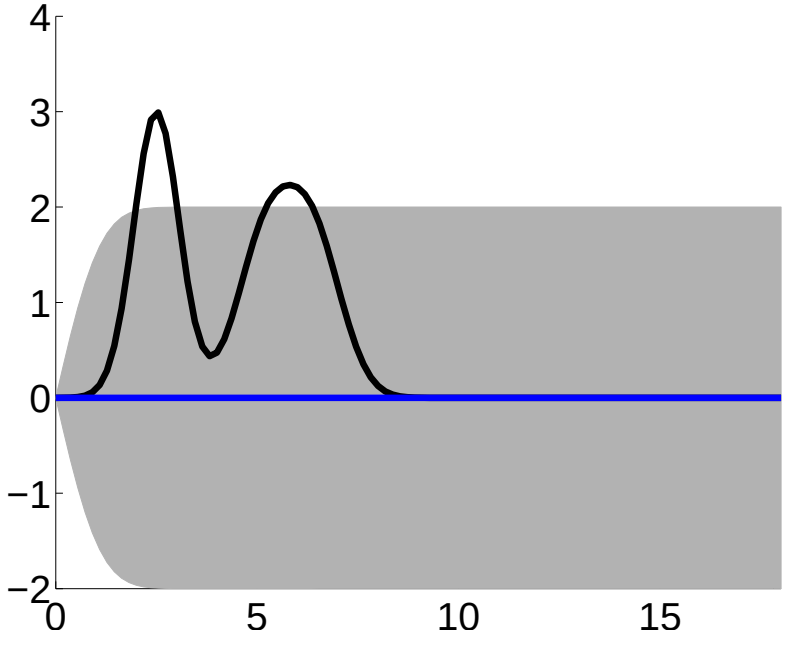
Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



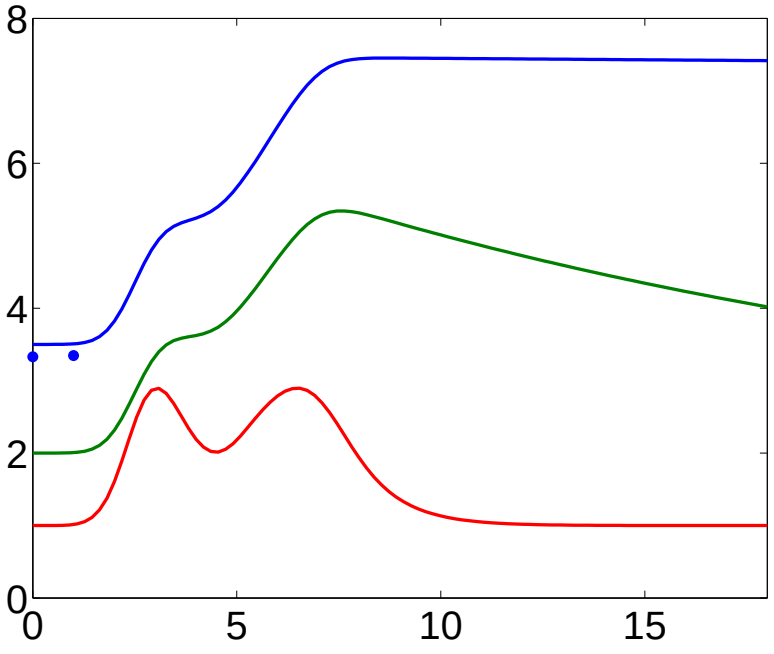
True "gene profiles" and noisy observations.



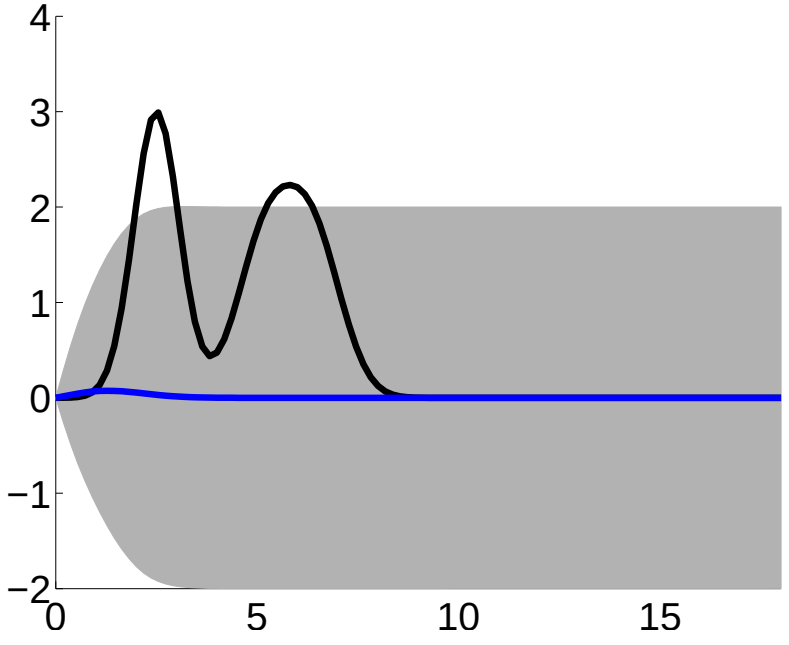
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



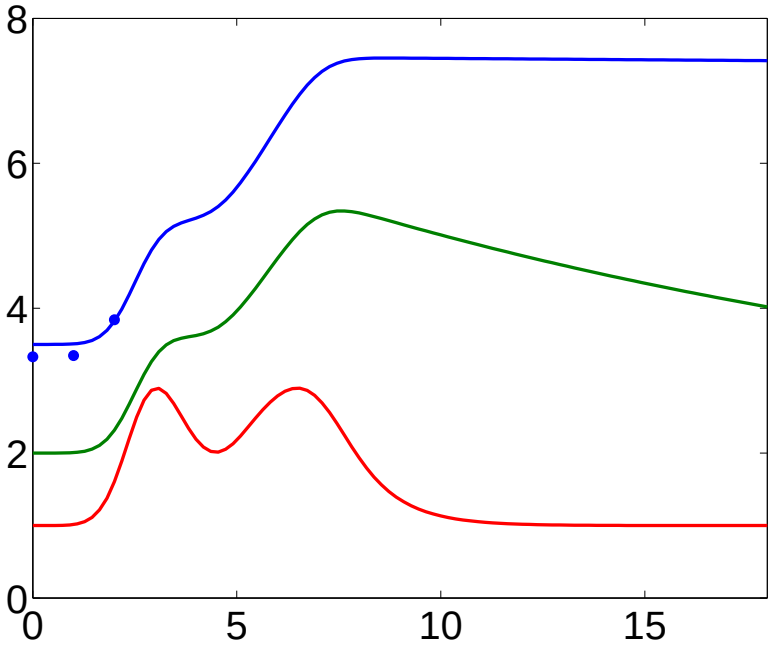
True "gene profiles" and noisy observations.



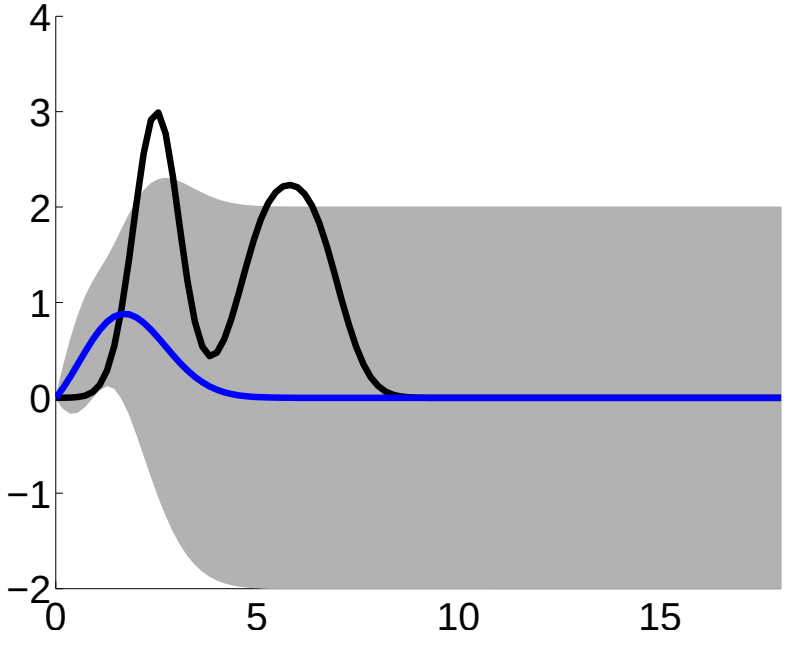
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



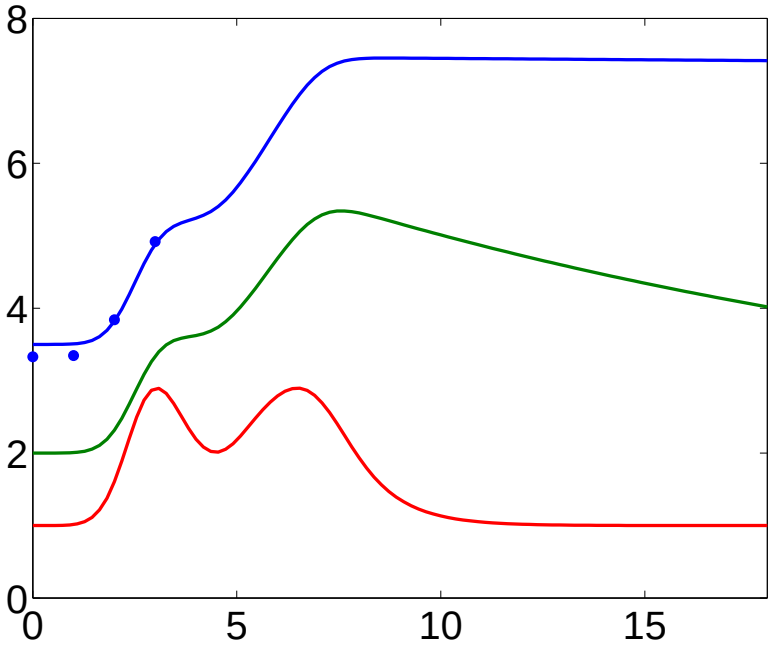
True "gene profiles" and noisy observations.



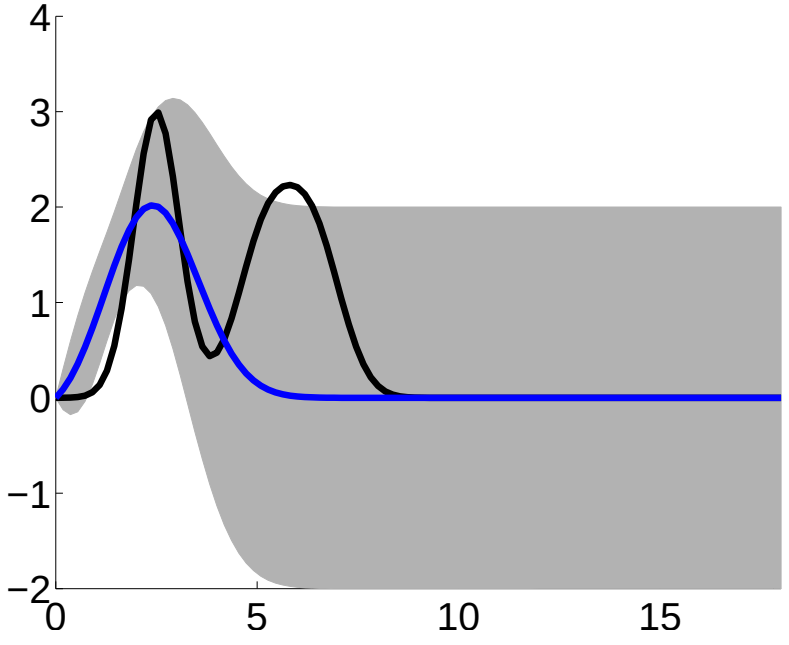
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



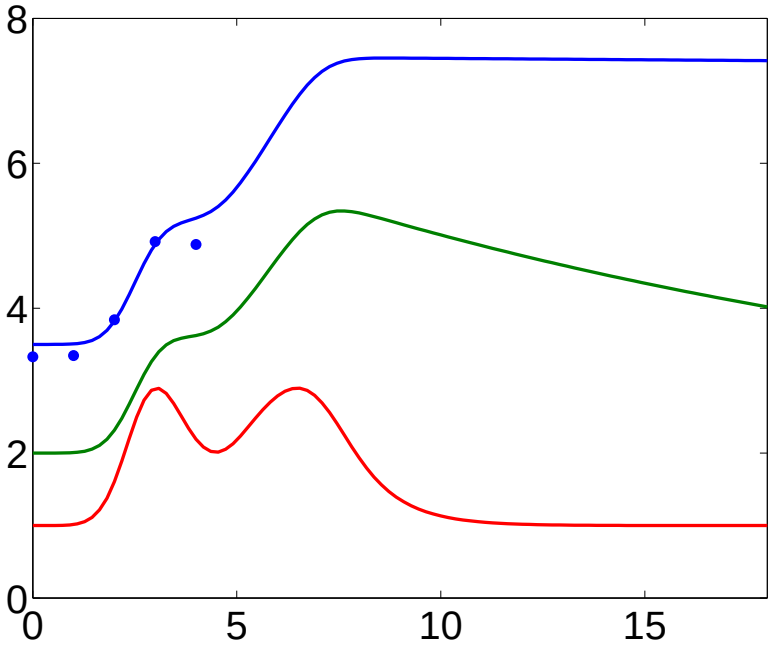
True "gene profiles" and noisy observations.



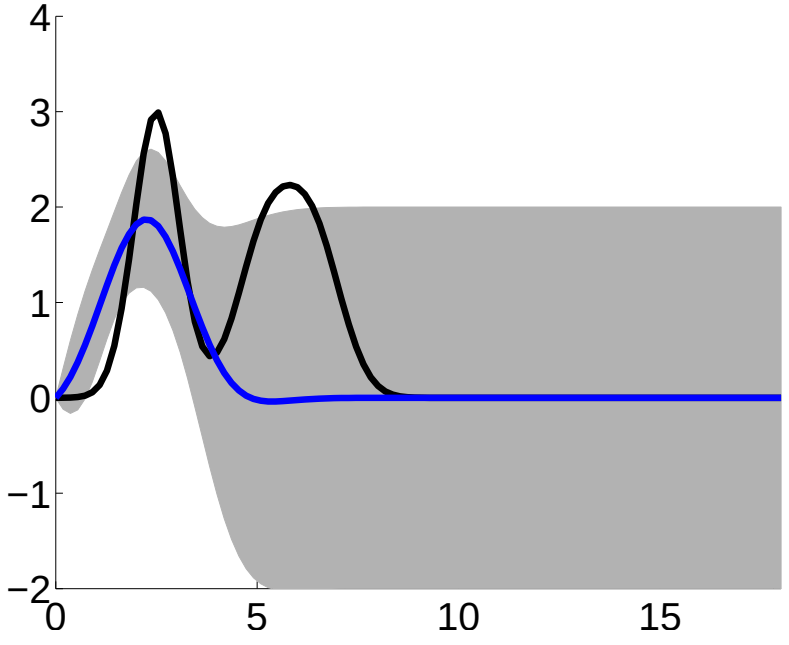
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



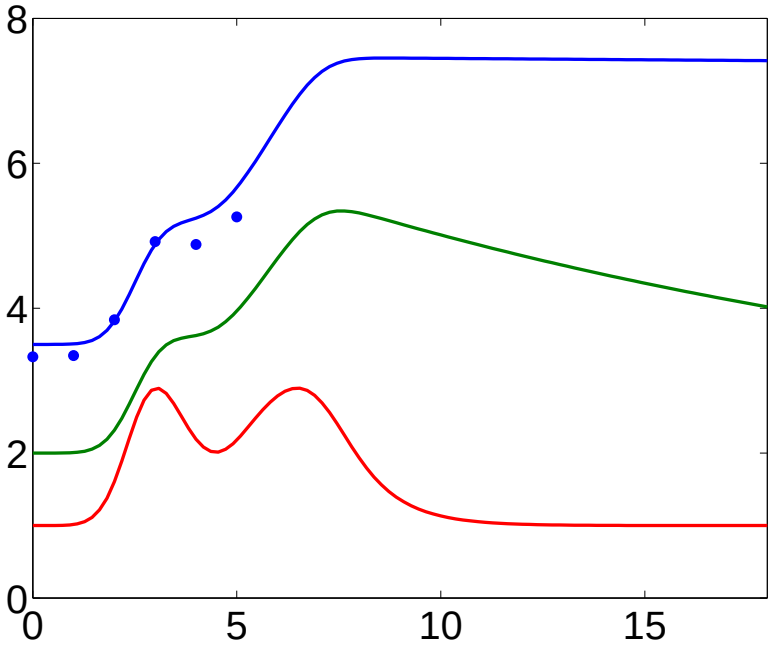
True "gene profiles" and noisy observations.



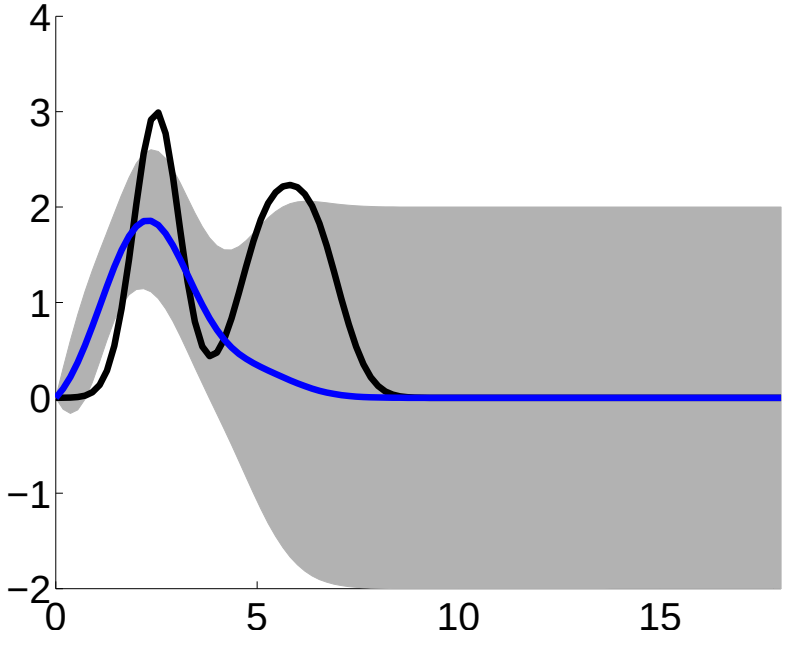
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



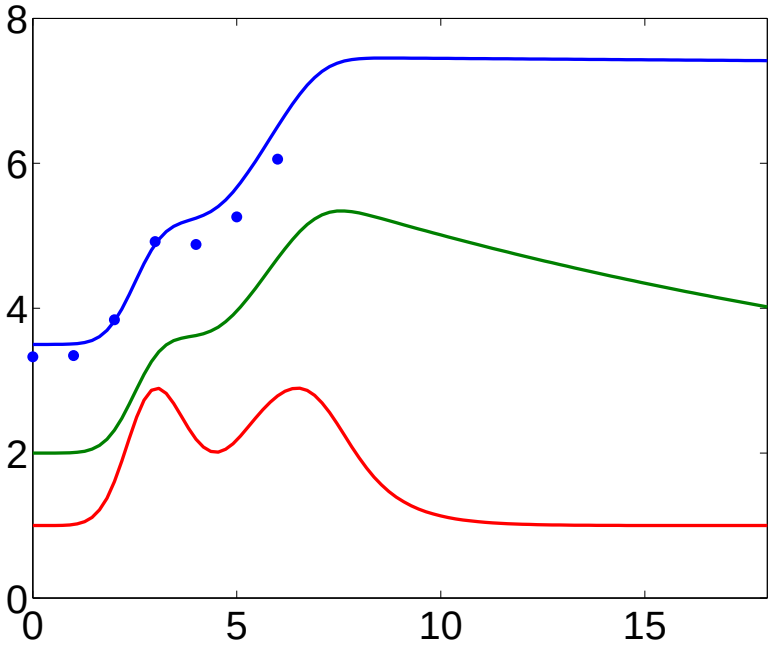
True "gene profiles" and noisy observations.



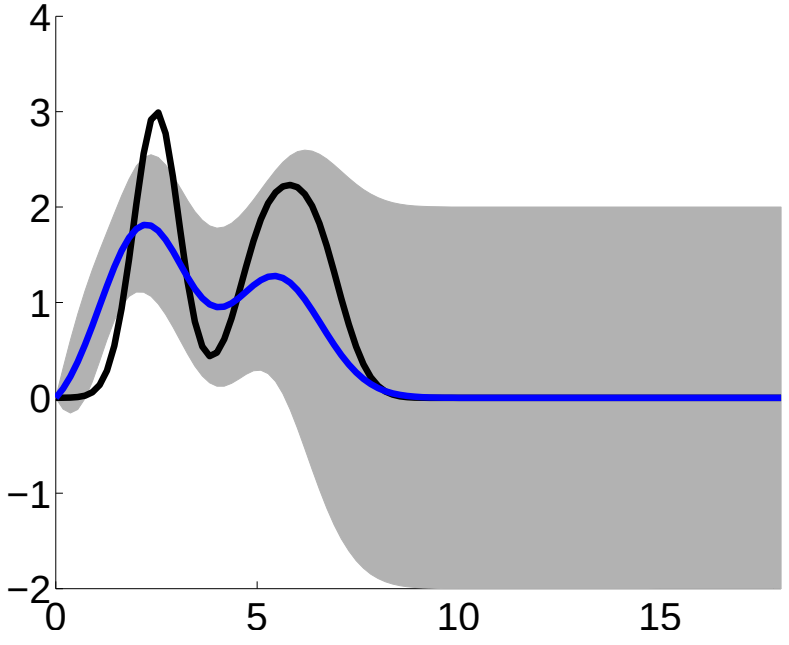
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



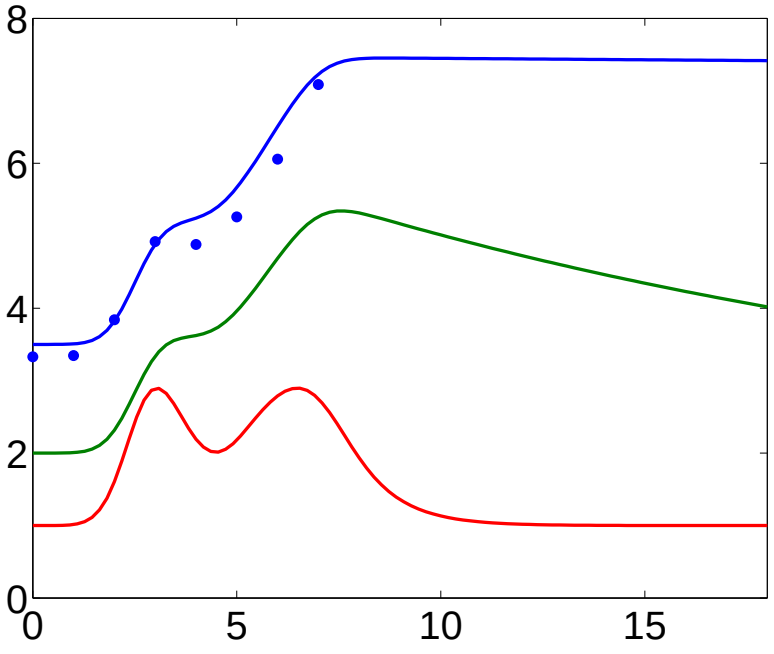
True "gene profiles" and noisy observations.



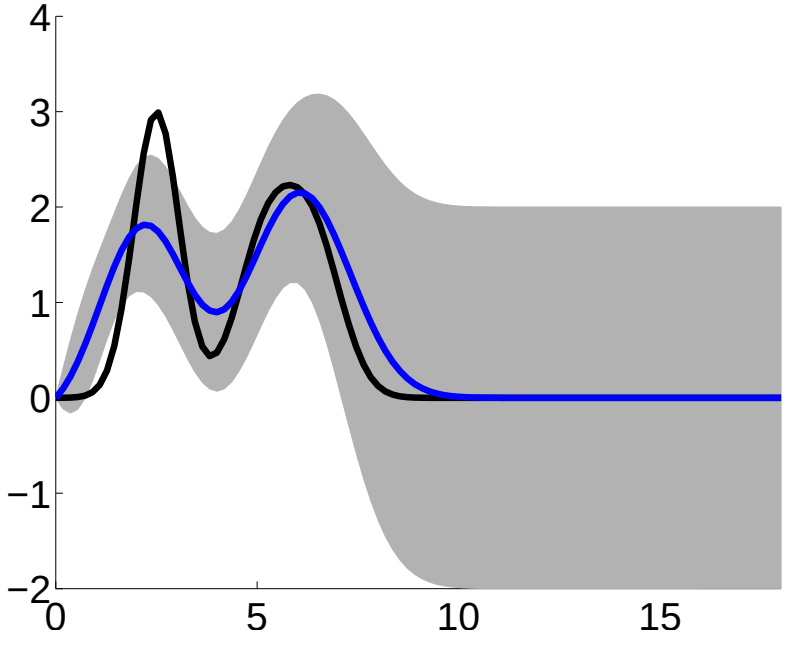
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



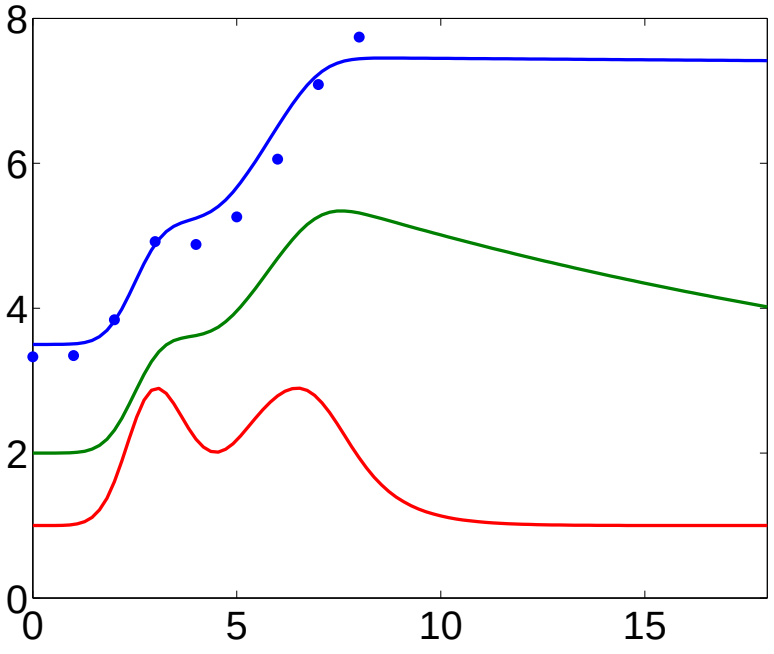
True "gene profiles" and noisy observations.



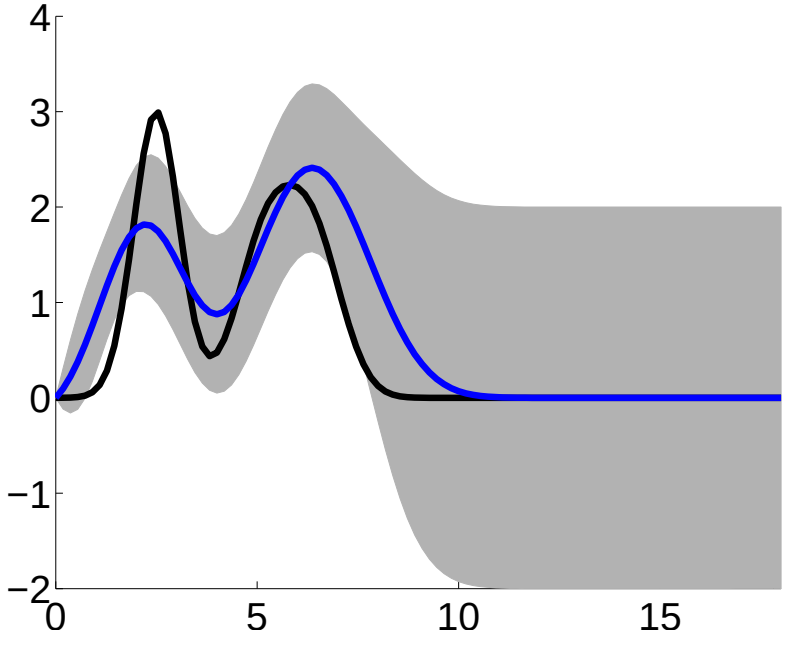
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



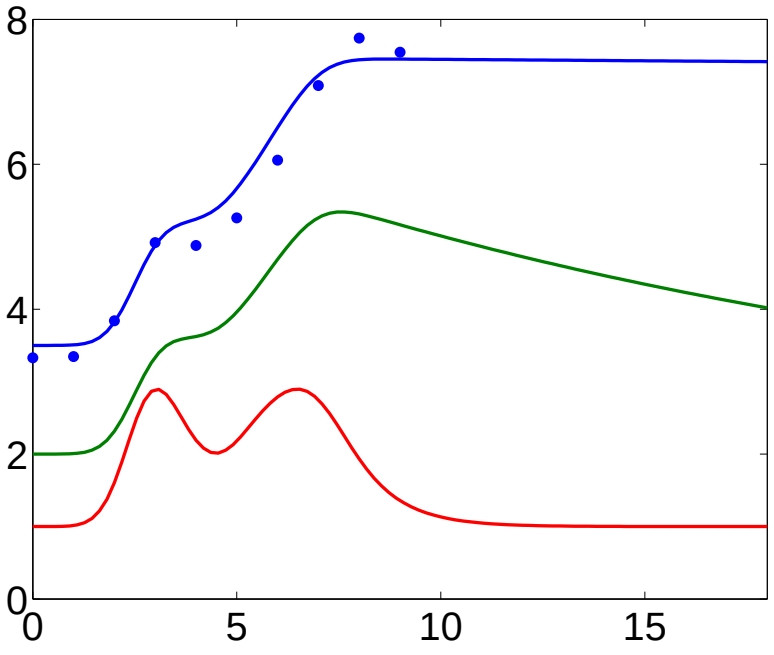
True "gene profiles" and noisy observations.



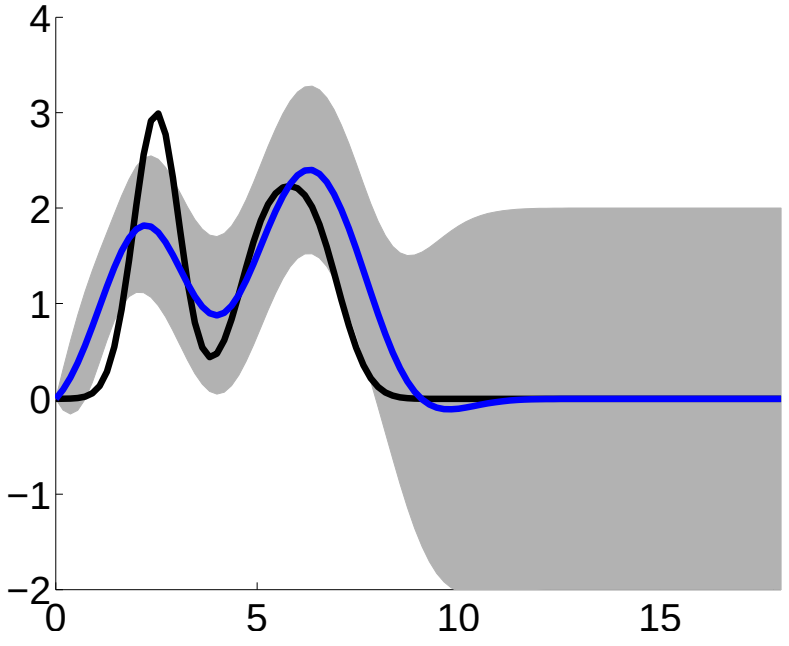
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



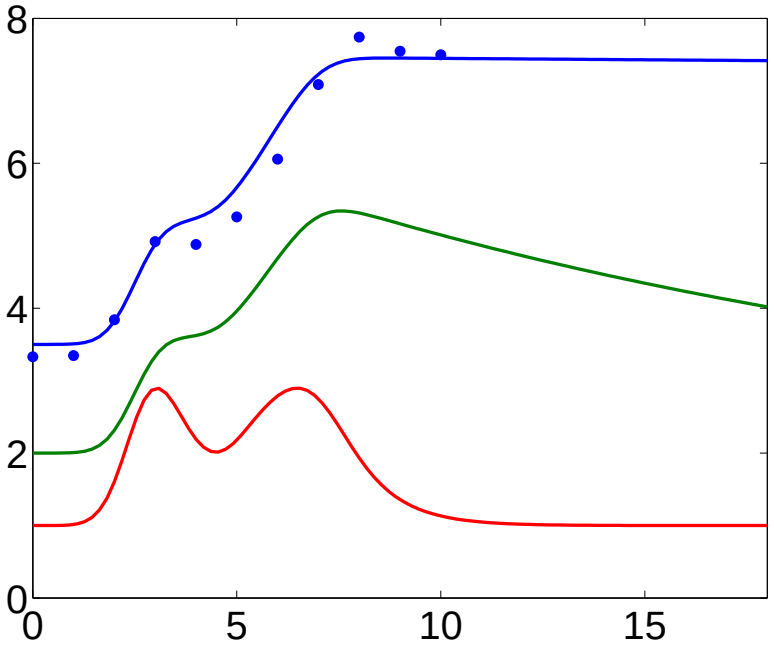
True "gene profiles" and noisy observations.



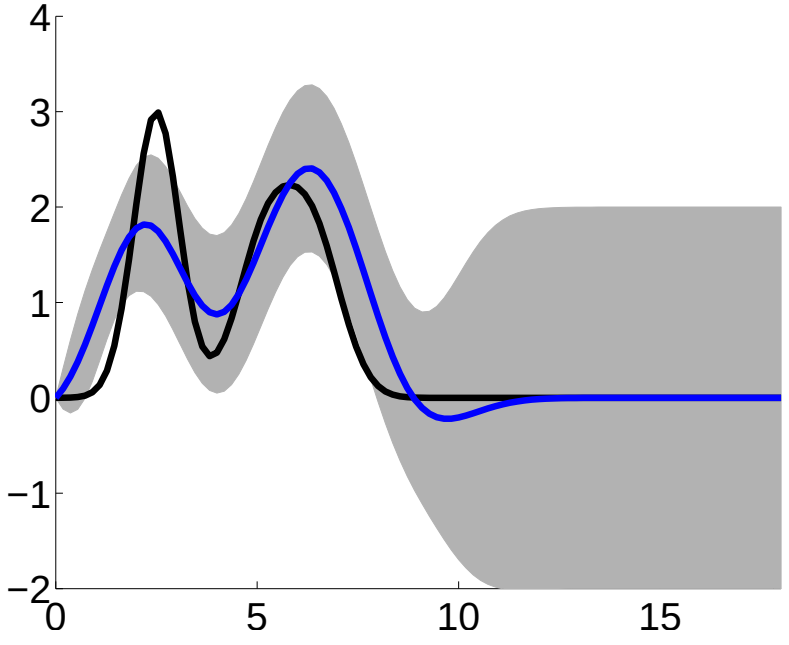
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



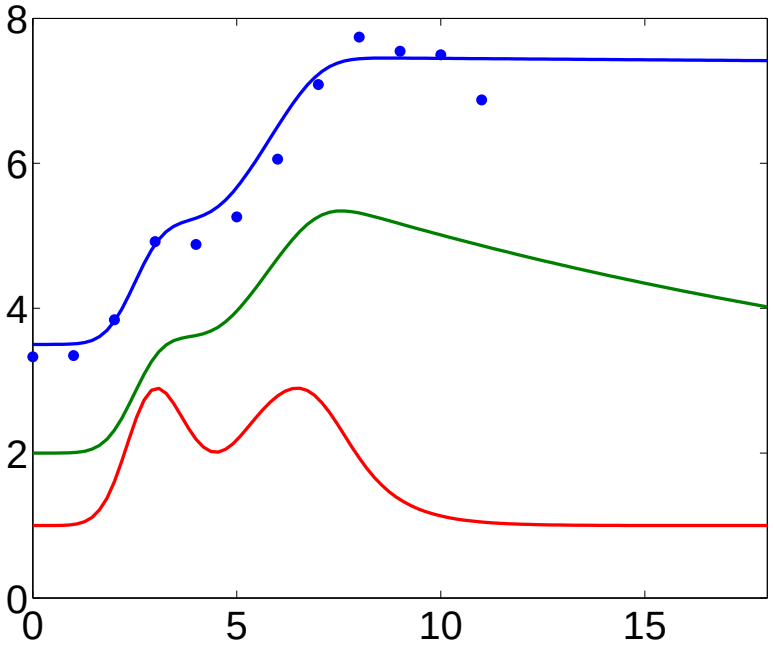
True "gene profiles" and noisy observations.



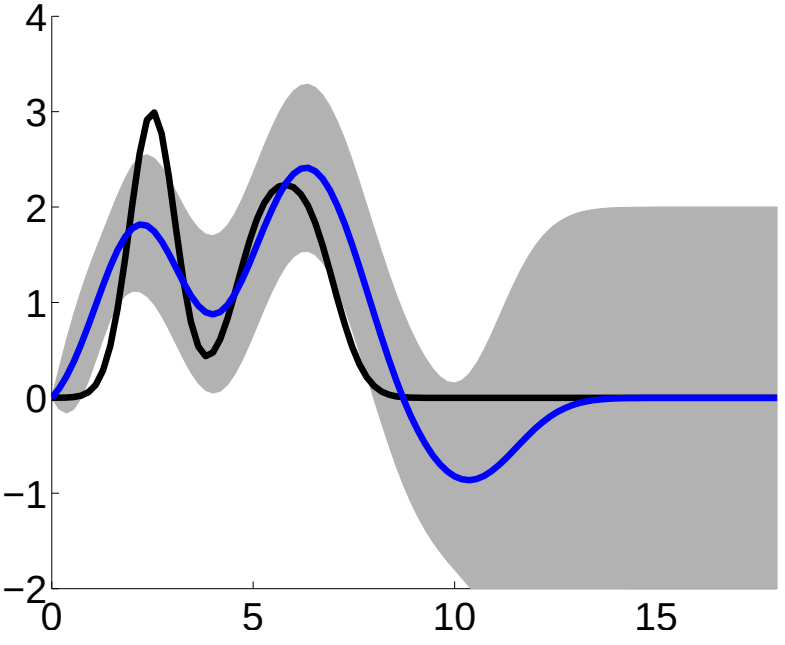
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



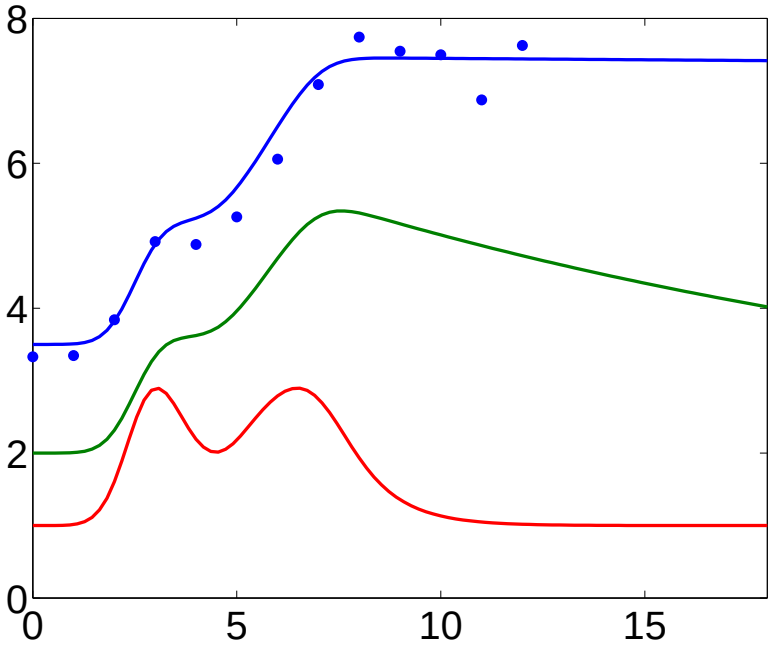
True "gene profiles" and noisy observations.



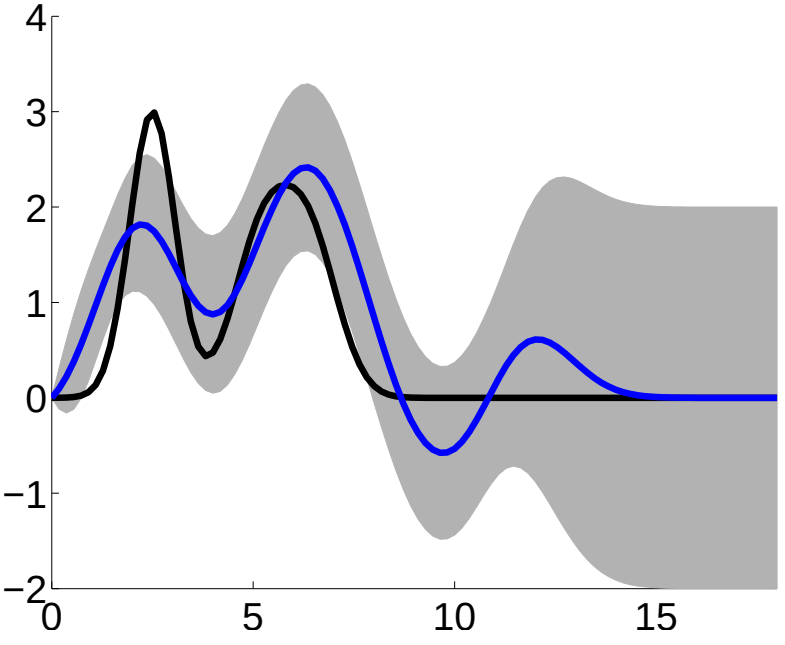
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



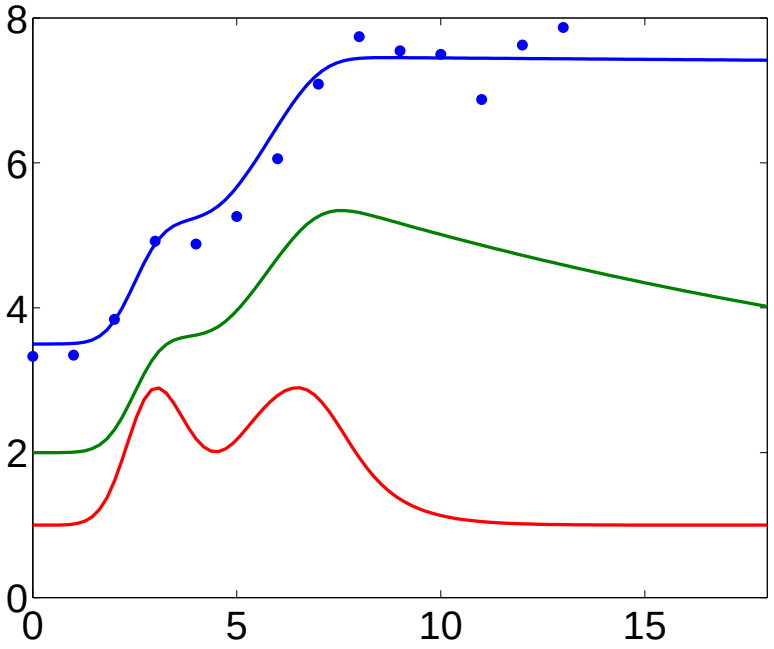
True "gene profiles" and noisy observations.



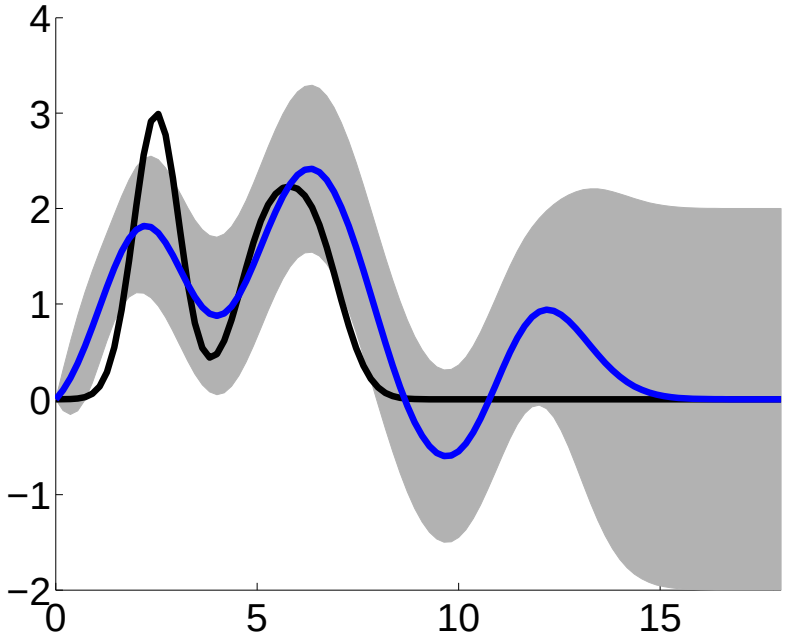
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



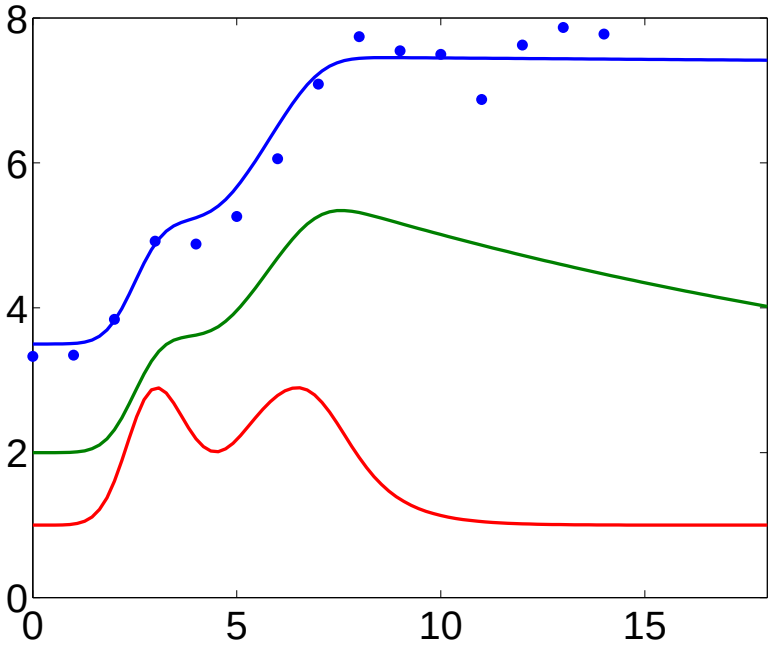
True "gene profiles" and noisy observations.



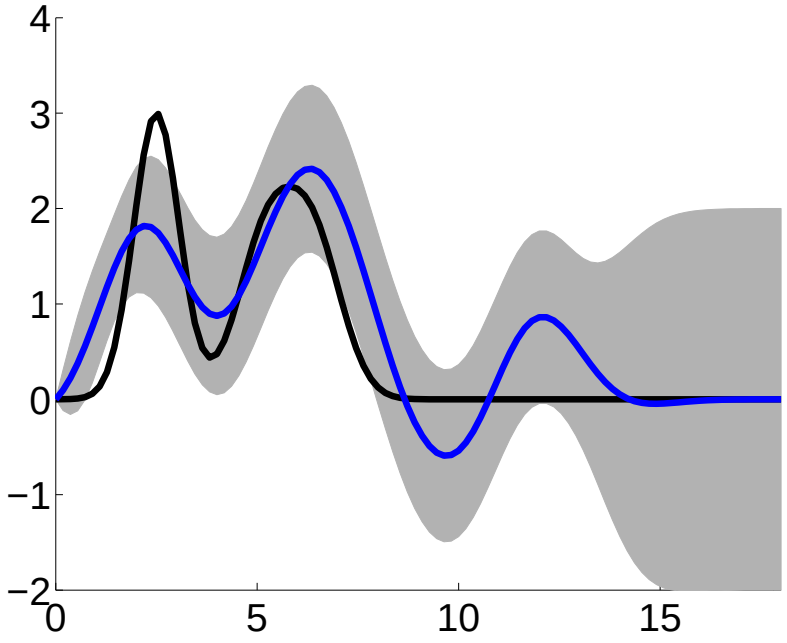
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



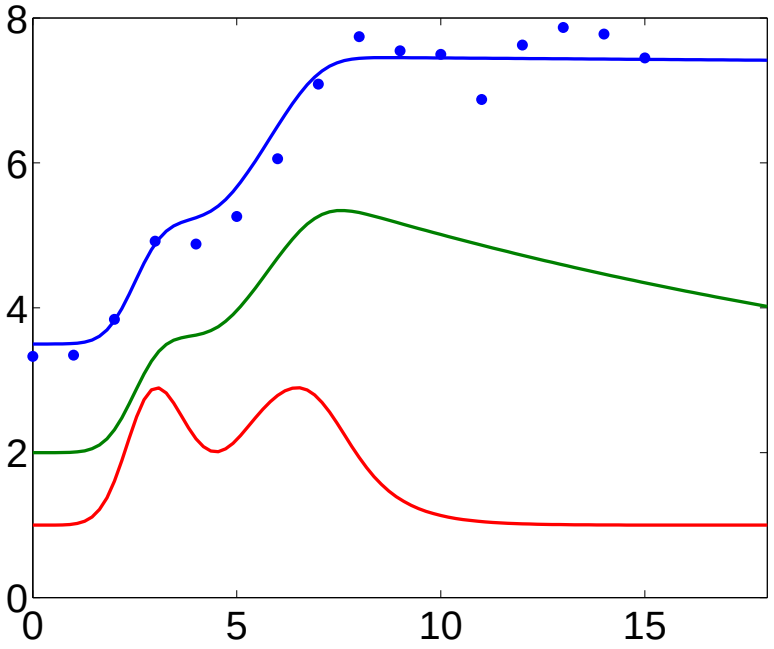
True "gene profiles" and noisy observations.



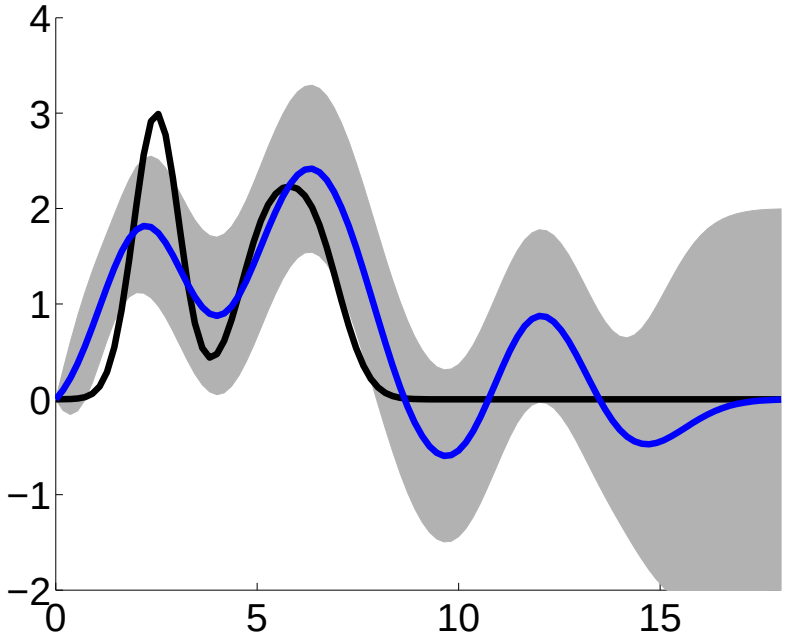
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



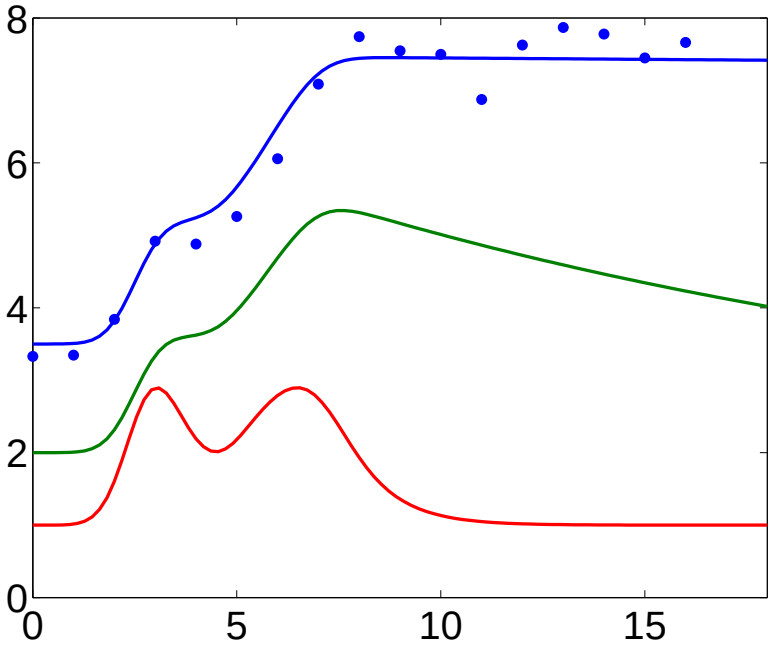
True "gene profiles" and noisy observations.



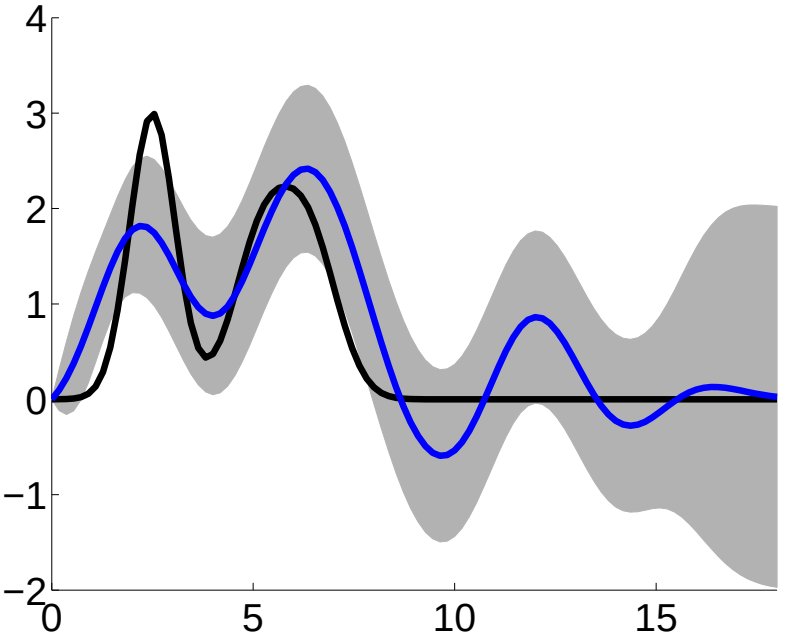
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



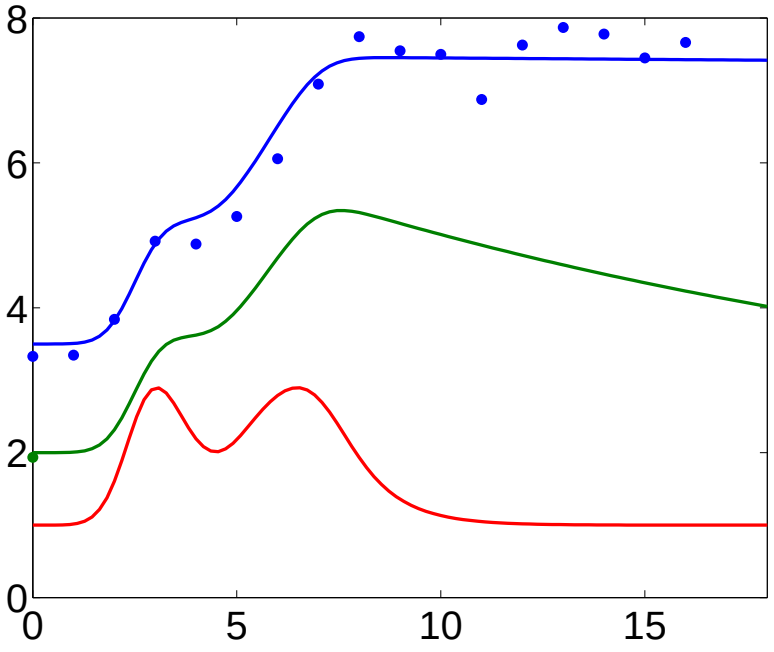
True "gene profiles" and noisy observations.



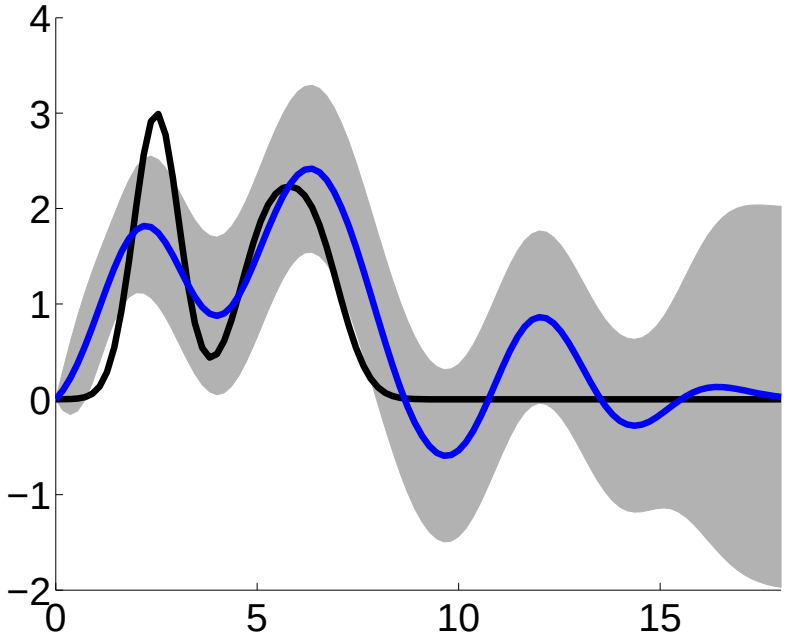
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



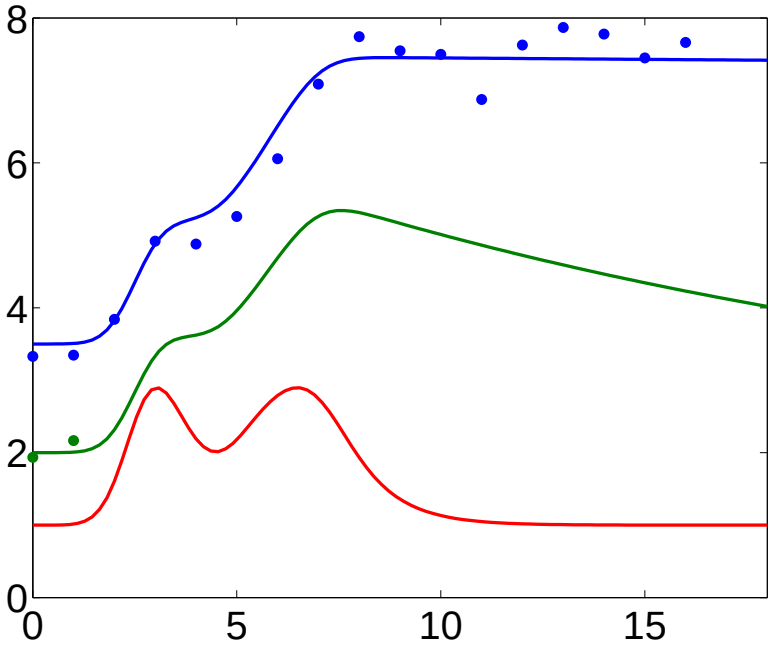
True "gene profiles" and noisy observations.



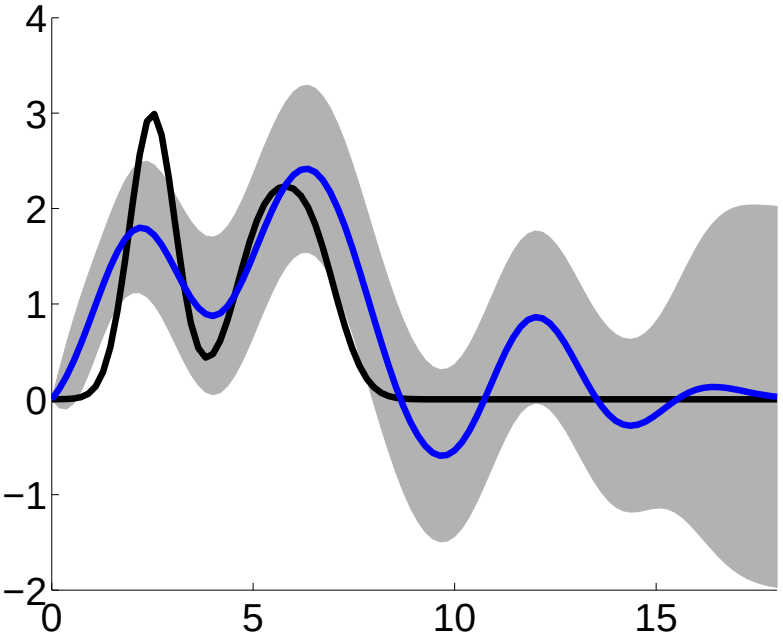
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



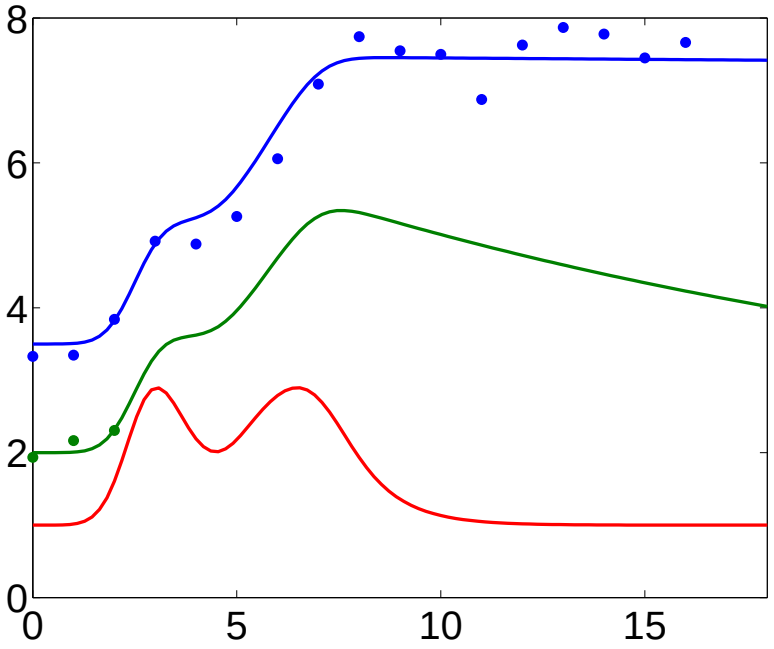
True "gene profiles" and noisy observations.



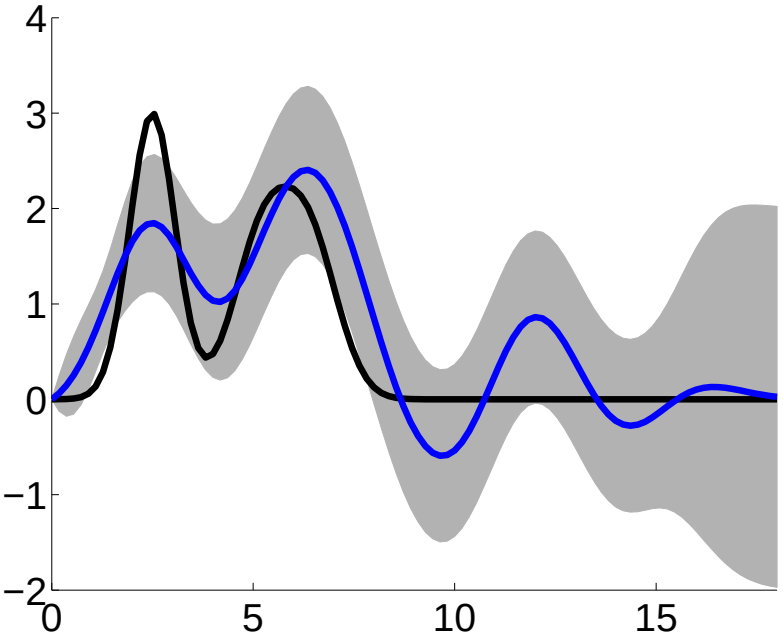
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



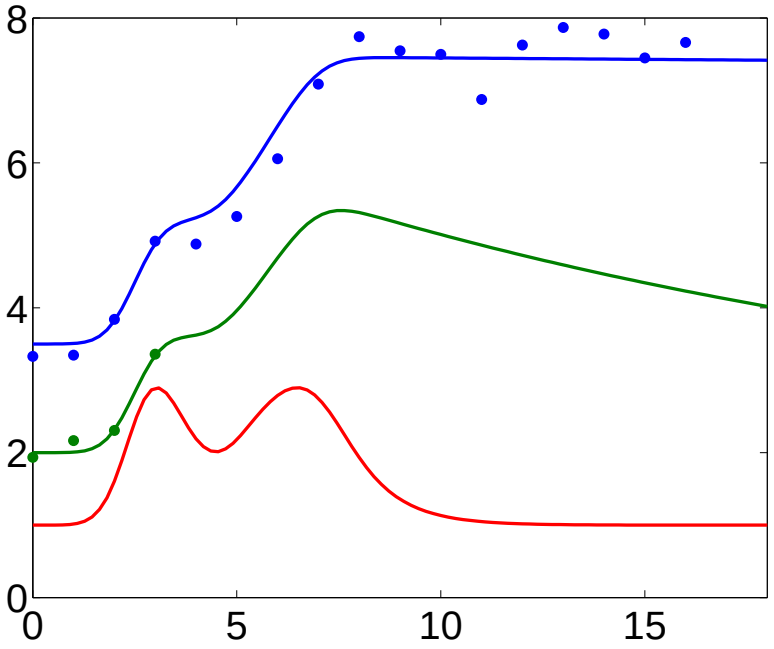
True "gene profiles" and noisy observations.



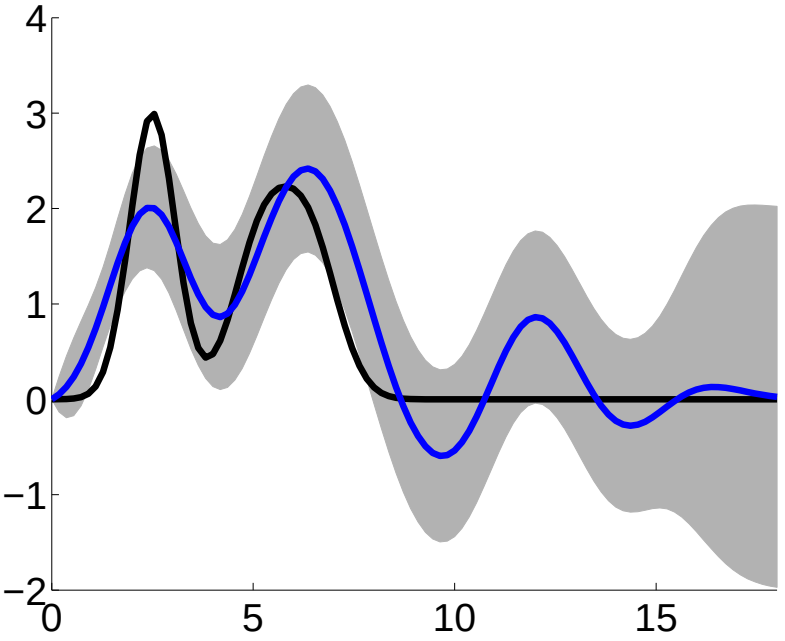
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



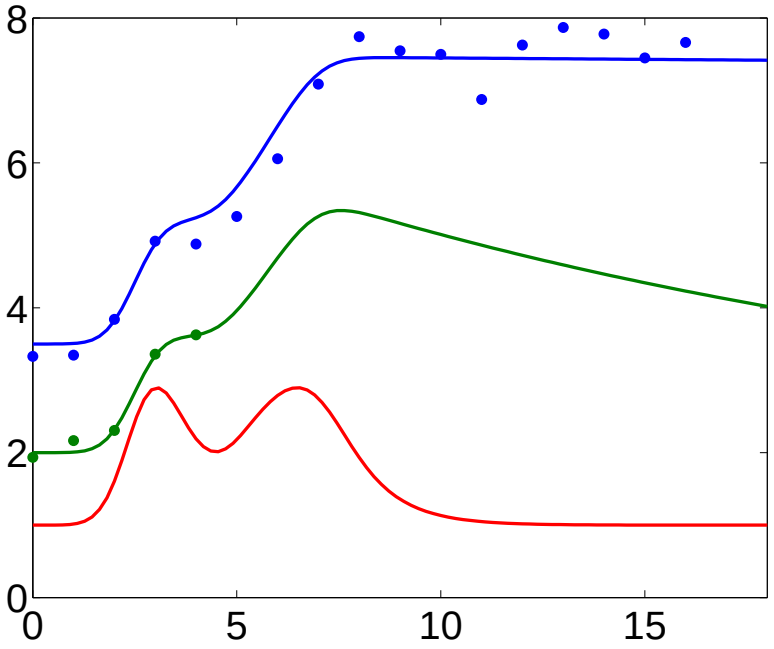
True "gene profiles" and noisy observations.



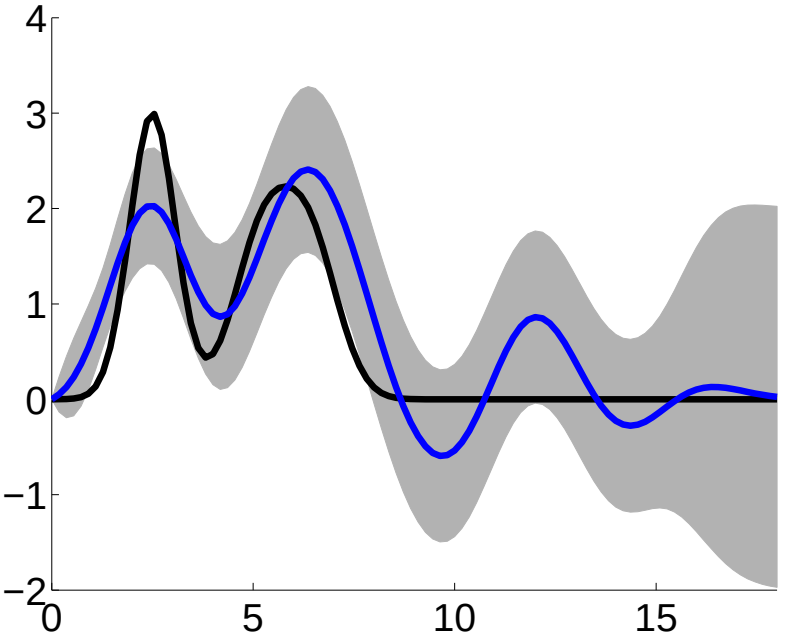
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



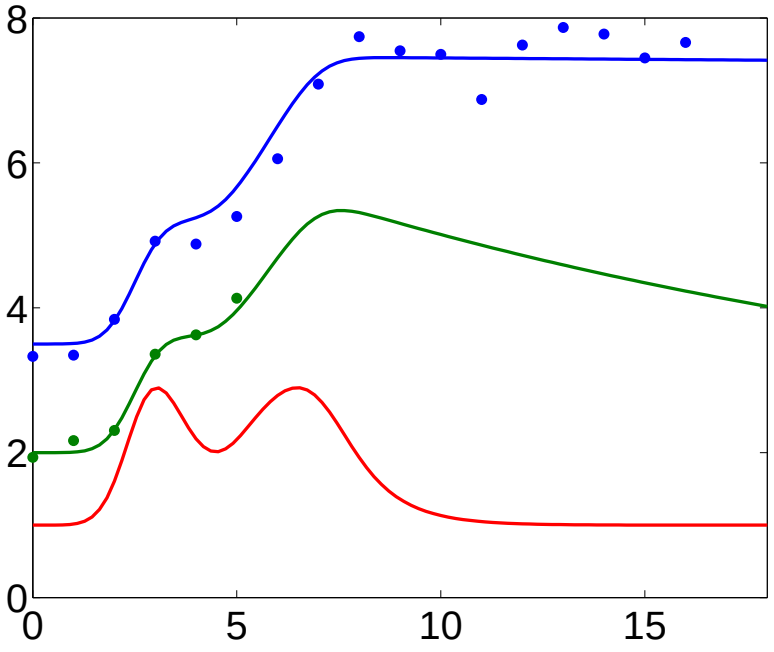
True "gene profiles" and noisy observations.



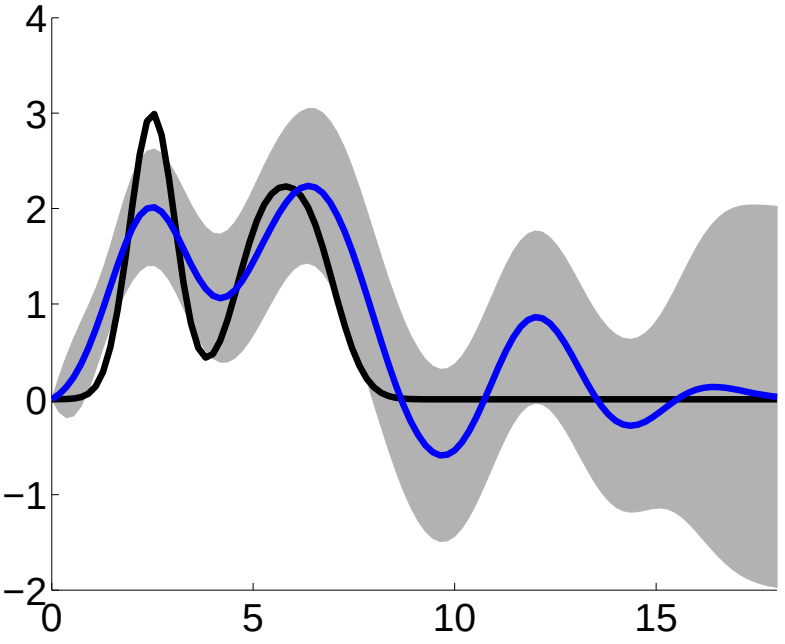
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



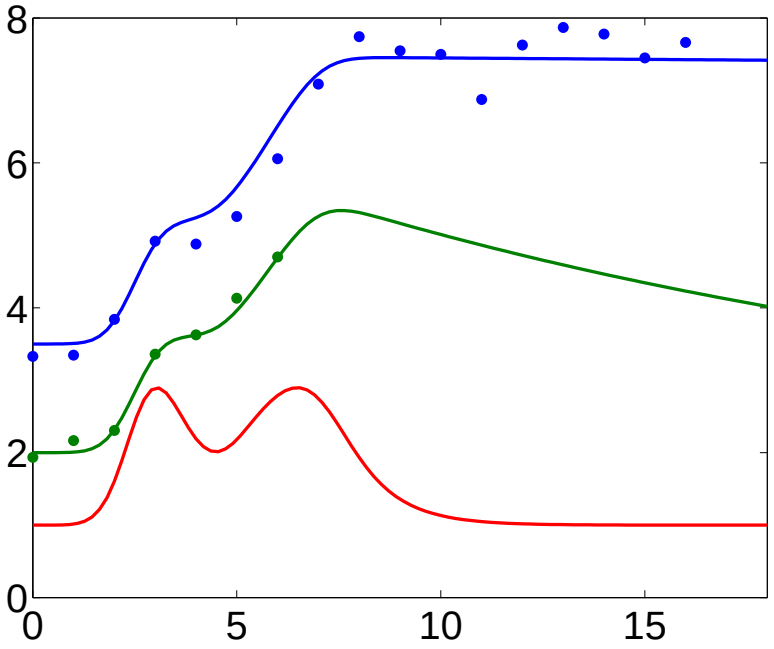
True "gene profiles" and noisy observations.



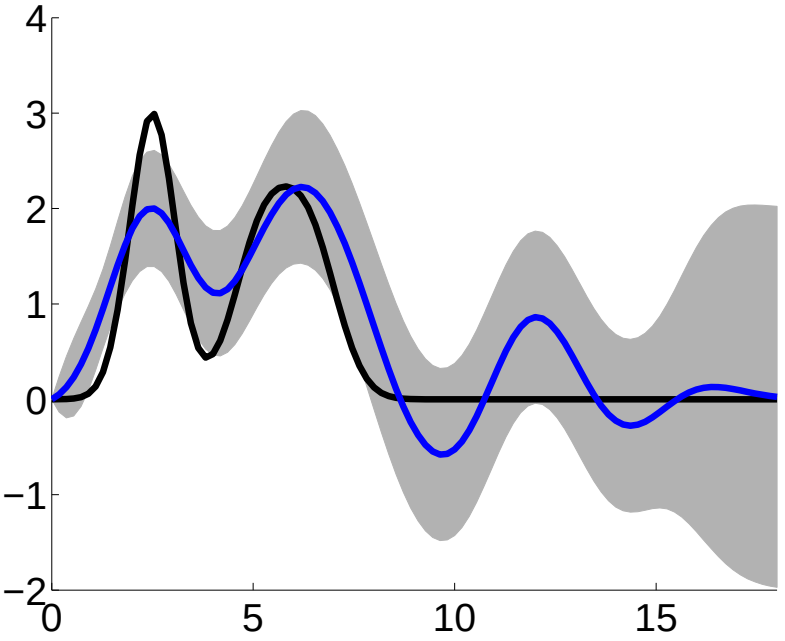
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



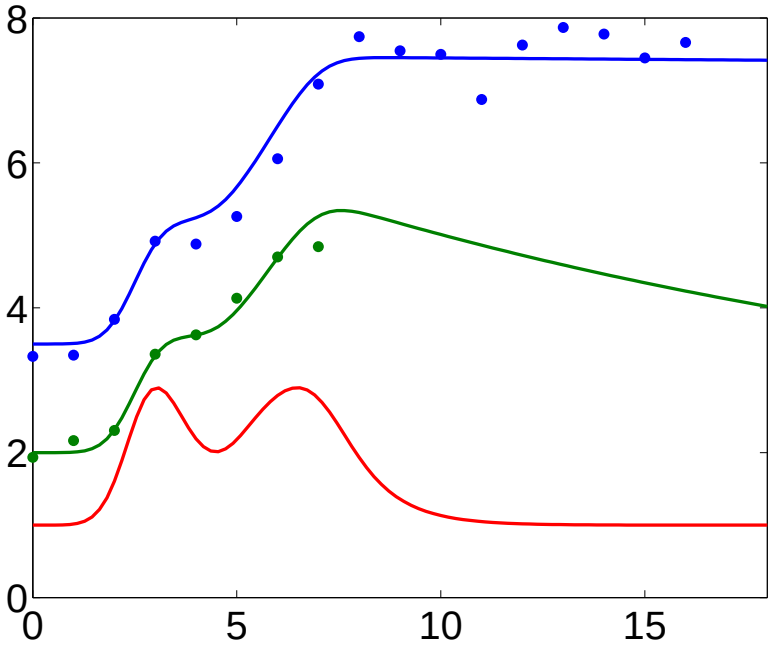
True "gene profiles" and noisy observations.



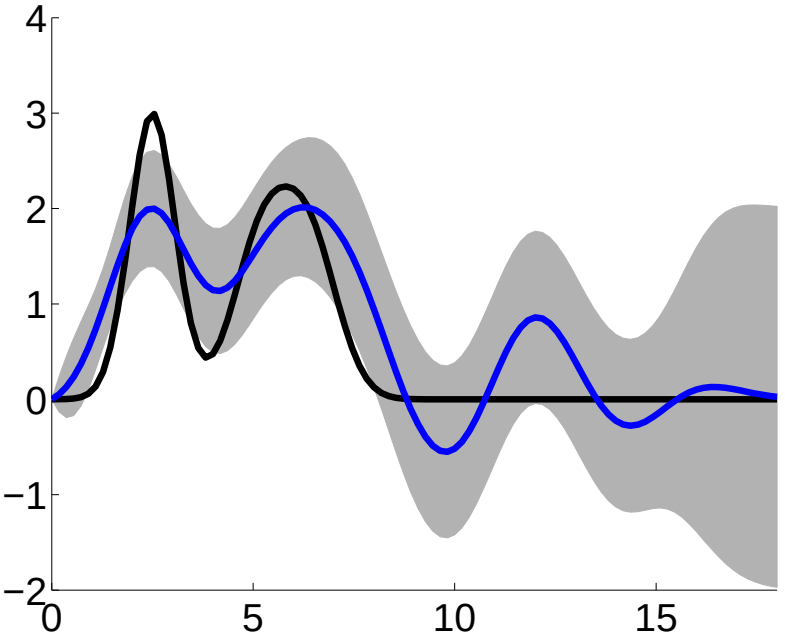
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



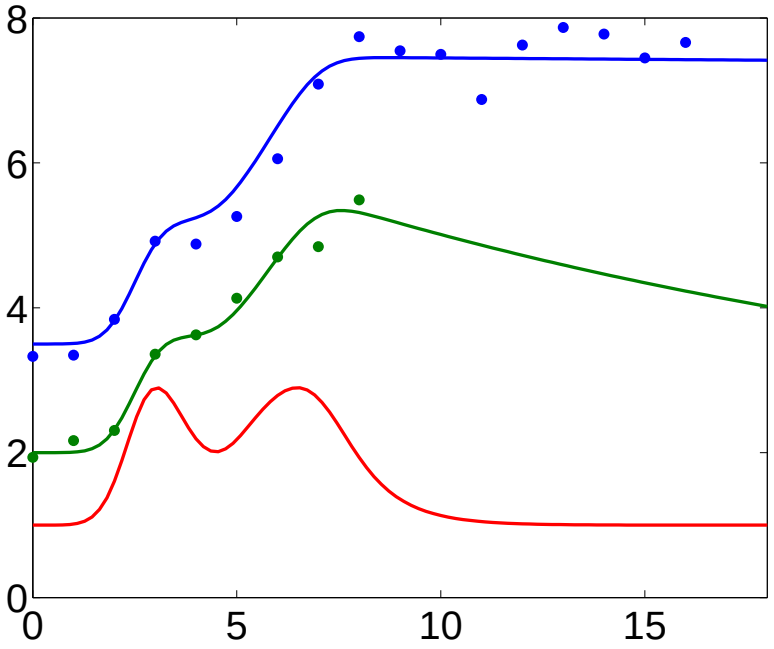
True "gene profiles" and noisy observations.



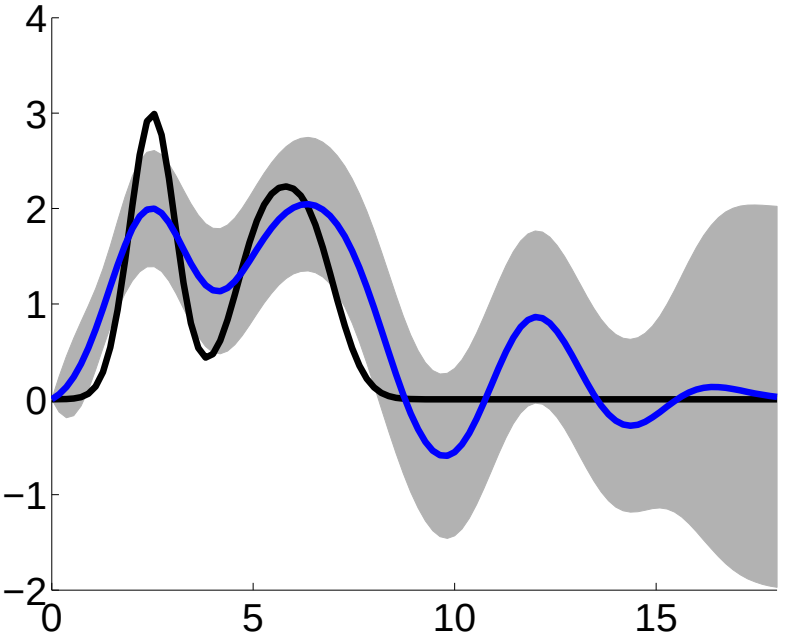
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



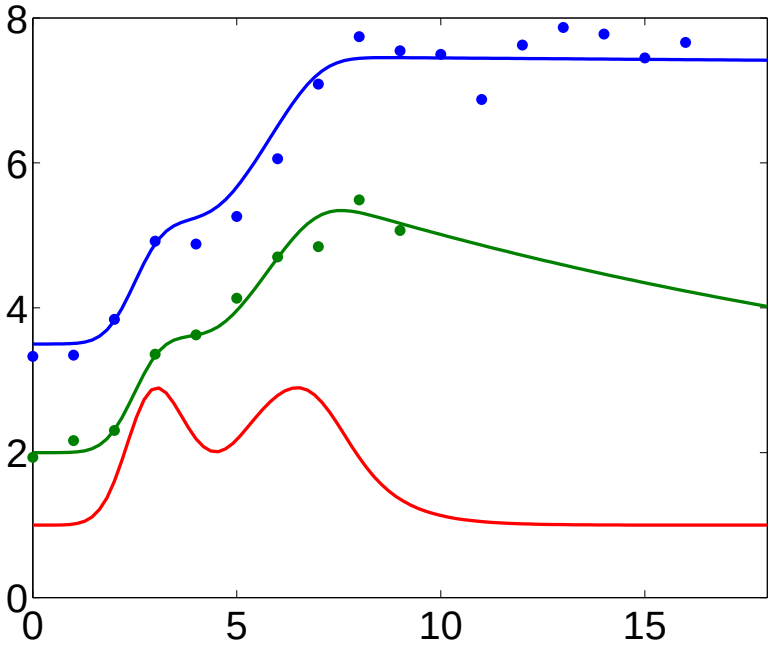
True "gene profiles" and noisy observations.



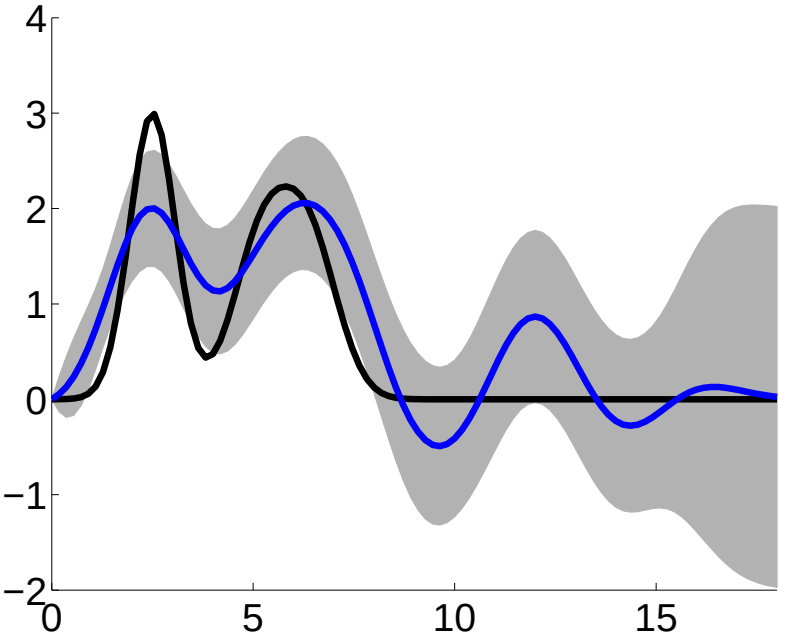
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



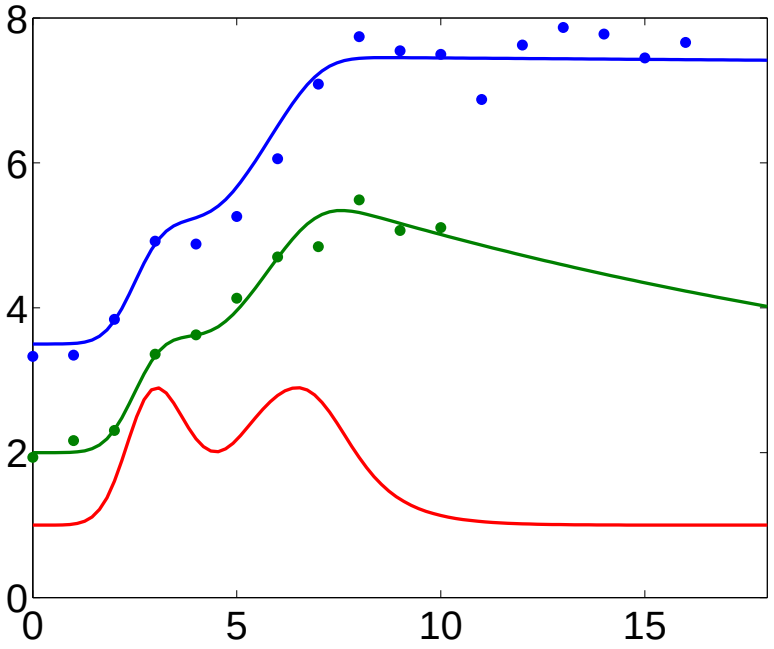
True "gene profiles" and noisy observations.



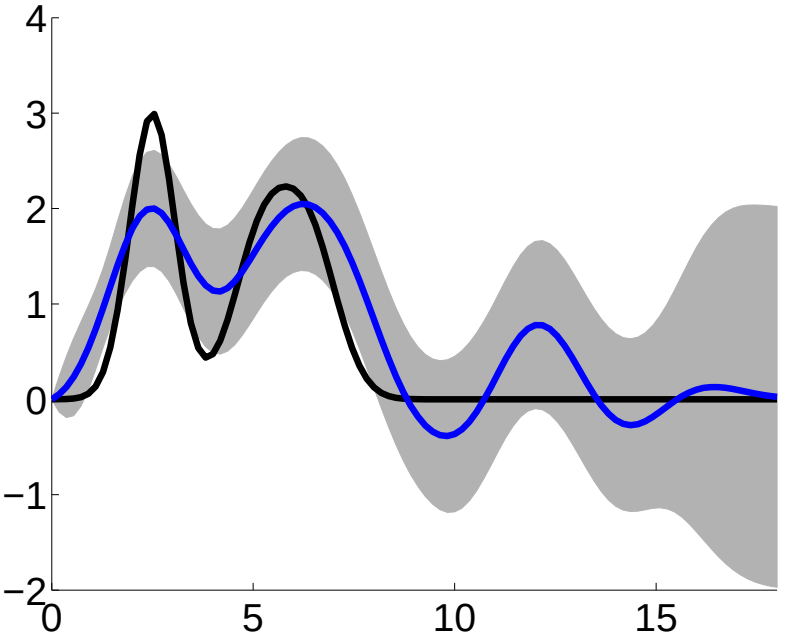
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



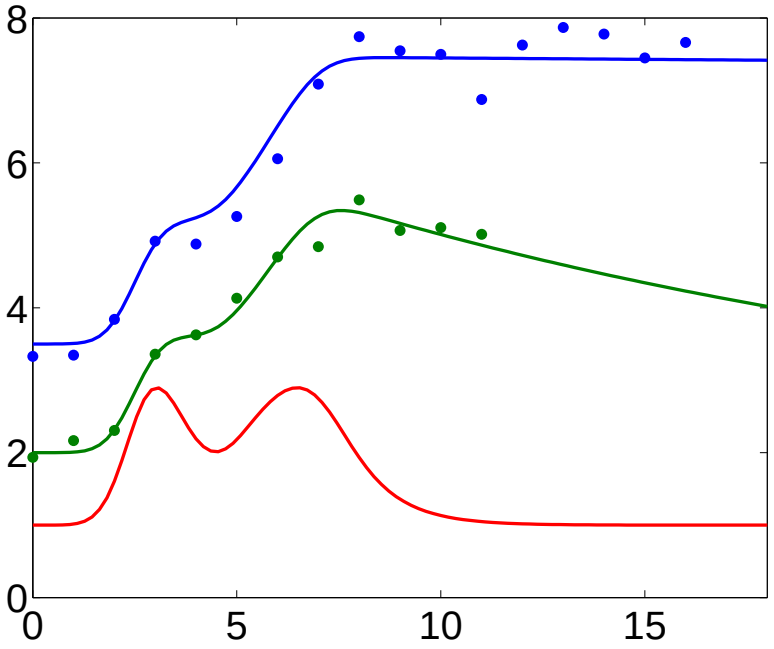
True "gene profiles" and noisy observations.



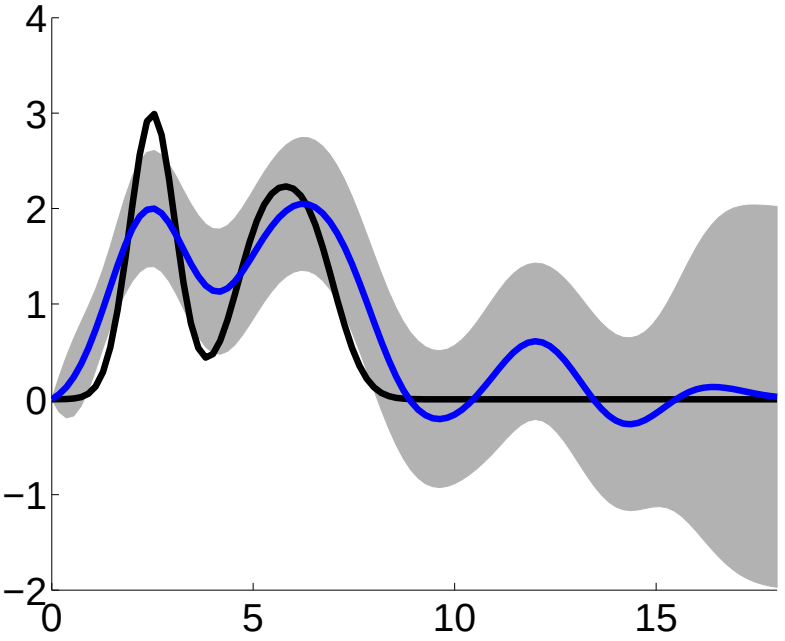
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



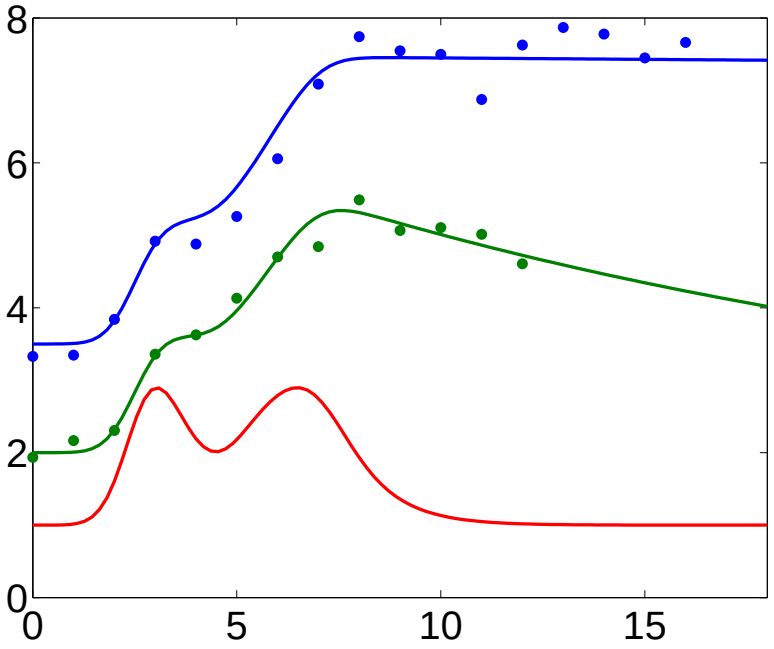
True "gene profiles" and noisy observations.



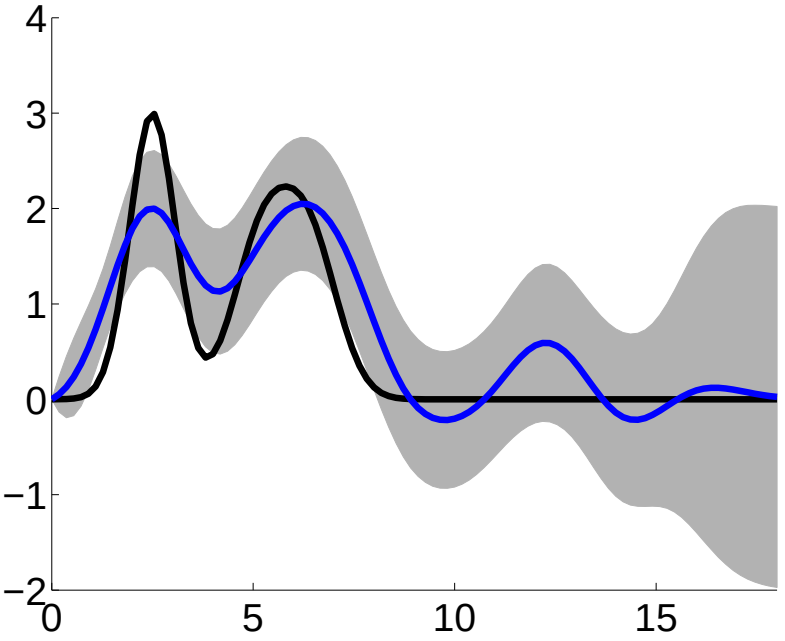
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



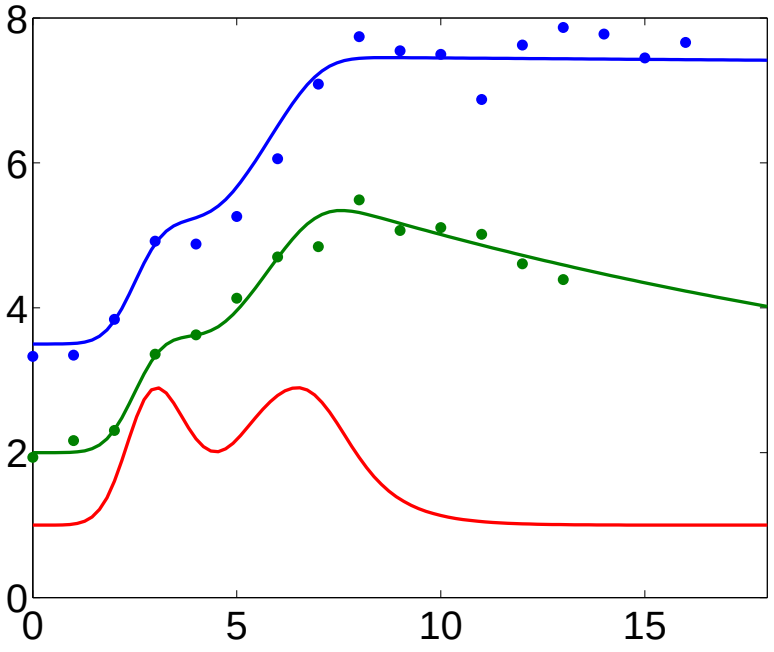
True "gene profiles" and noisy observations.



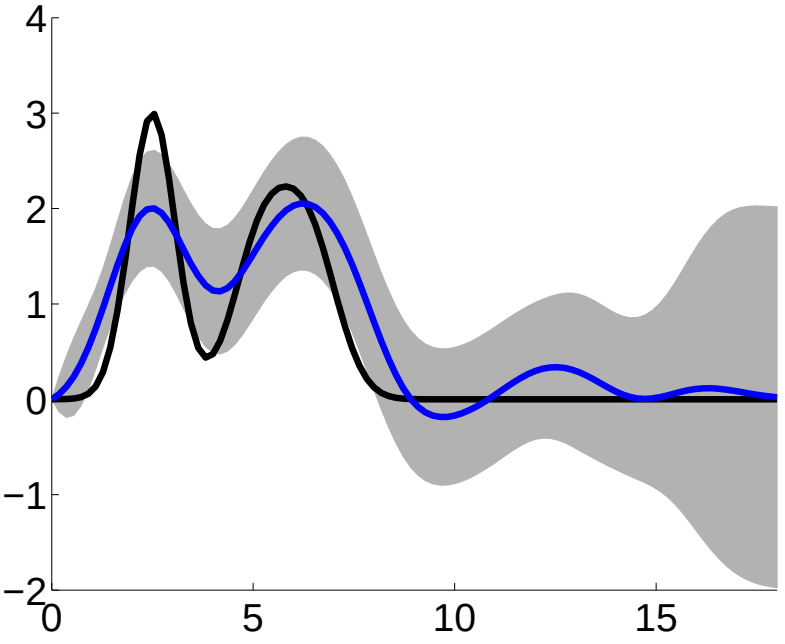
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



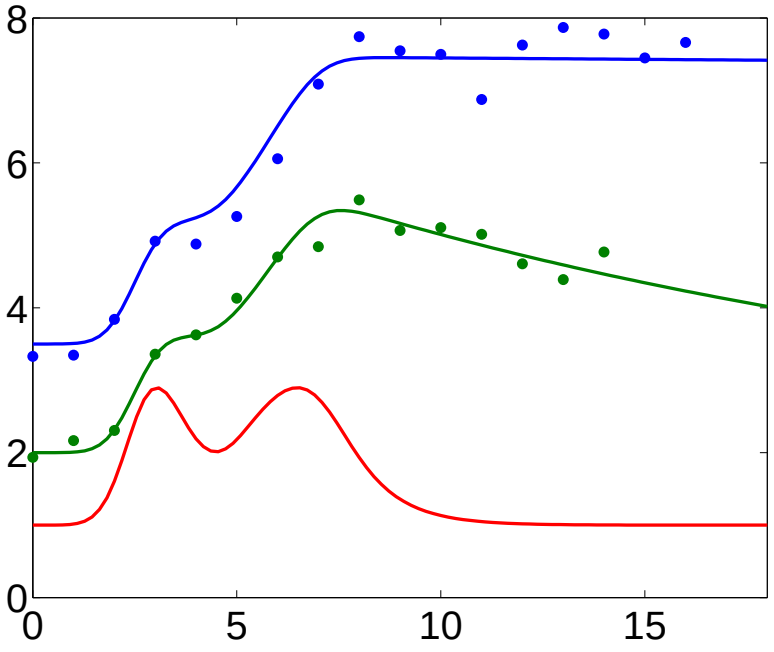
True "gene profiles" and noisy observations.



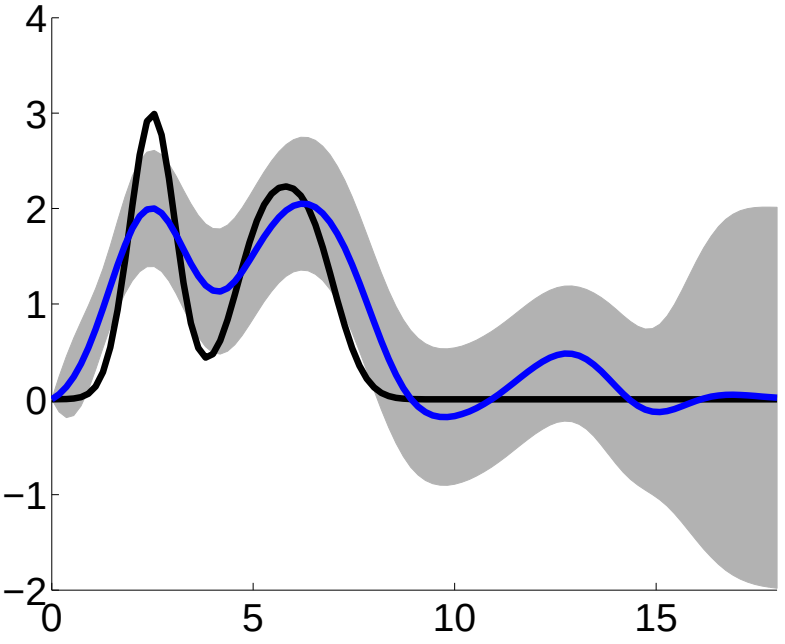
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



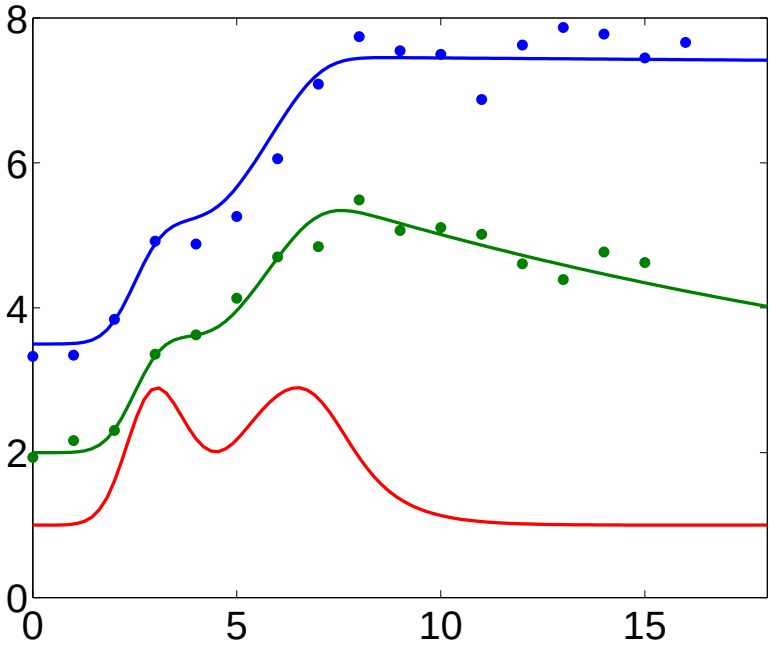
True "gene profiles" and noisy observations.



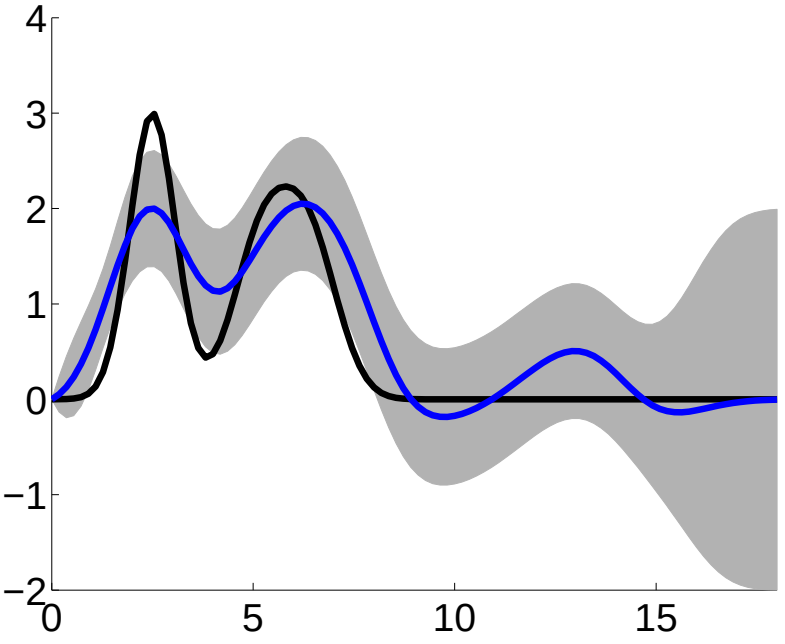
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



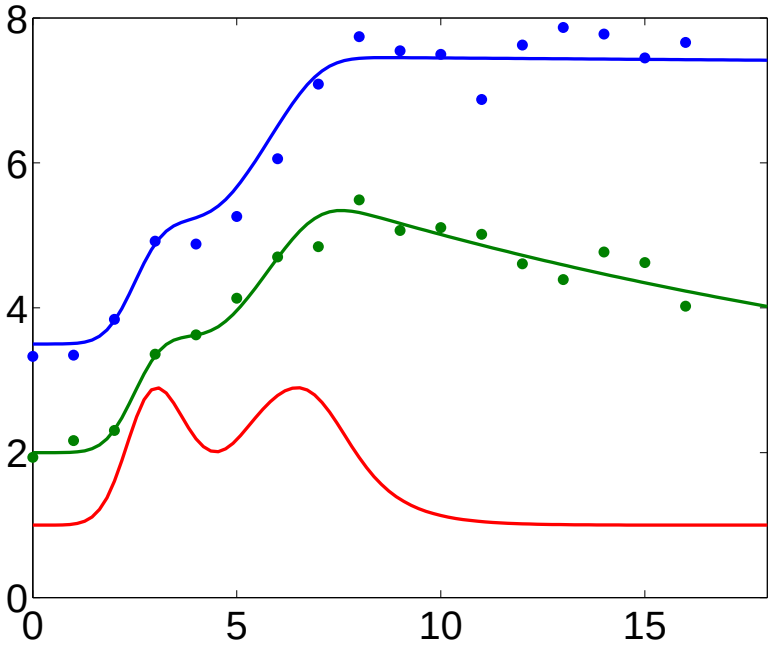
True "gene profiles" and noisy observations.



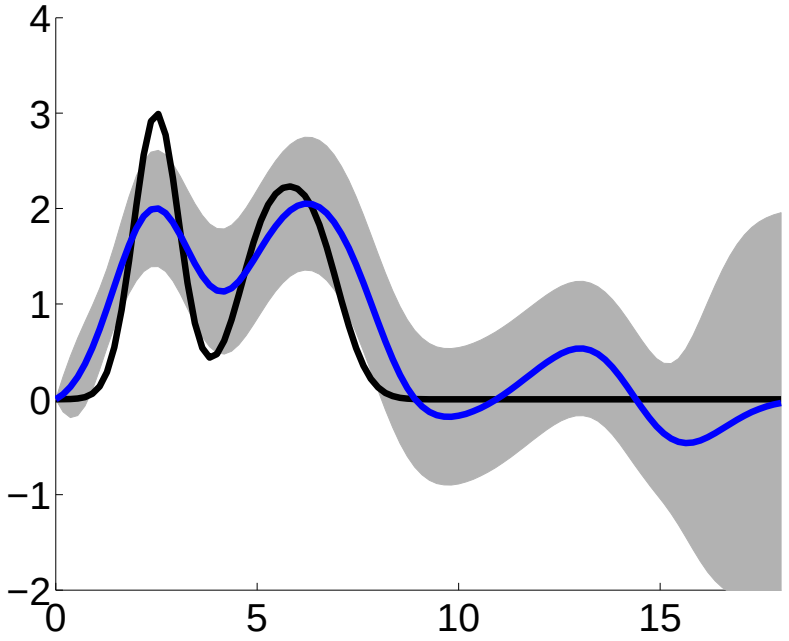
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



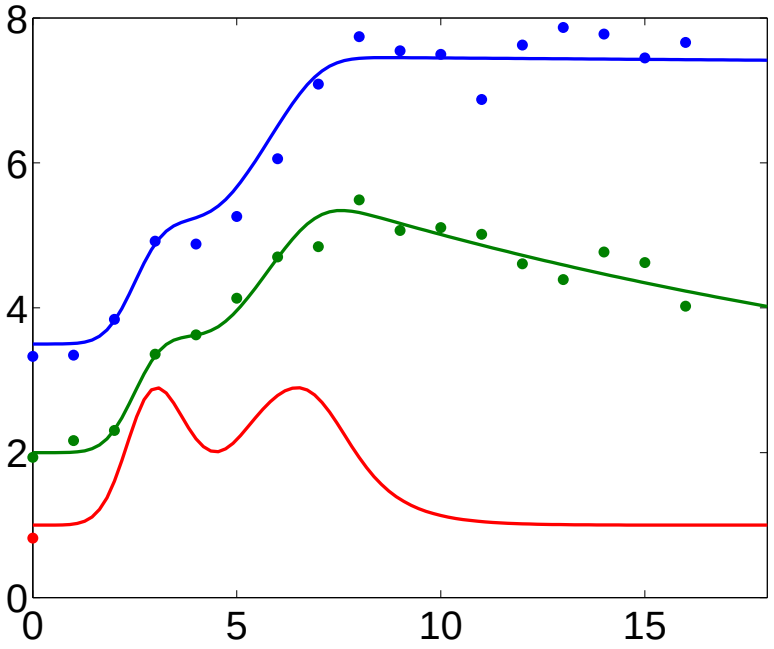
True "gene profiles" and noisy observations.



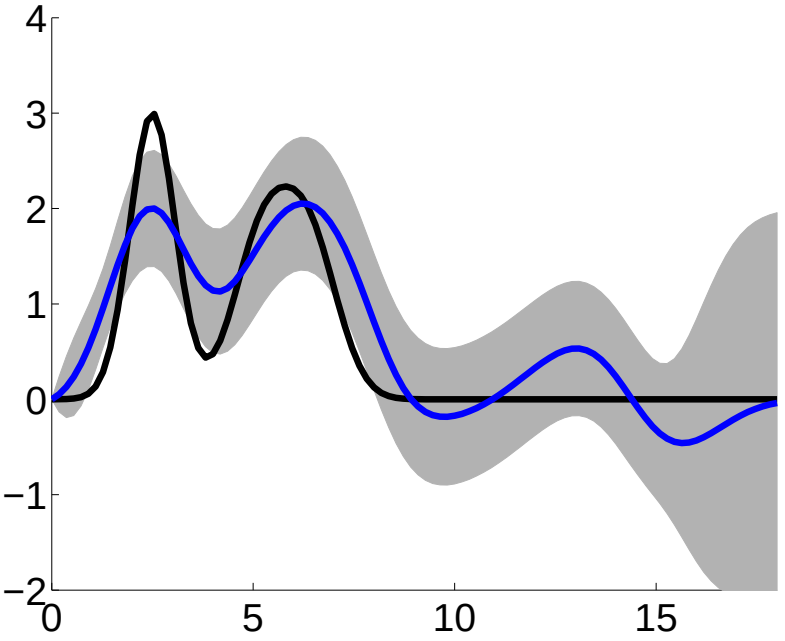
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



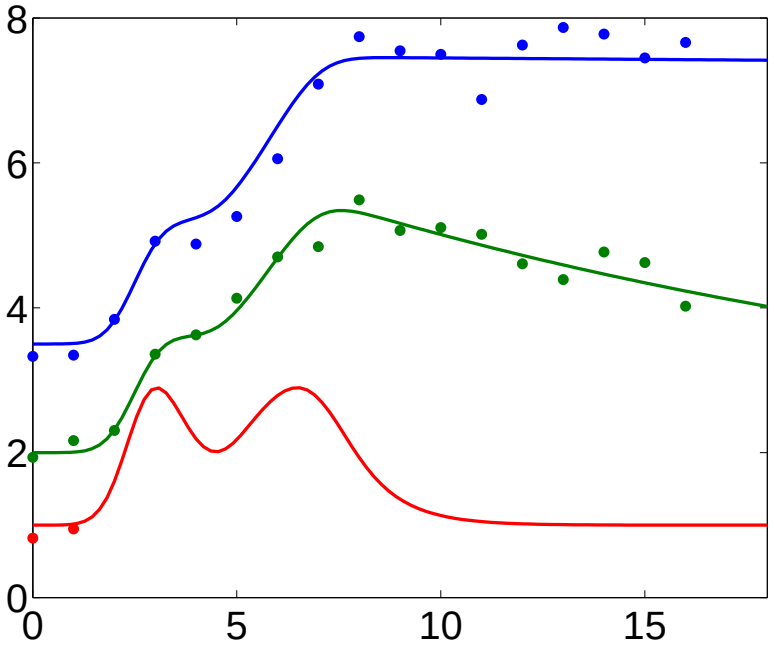
True "gene profiles" and noisy observations.



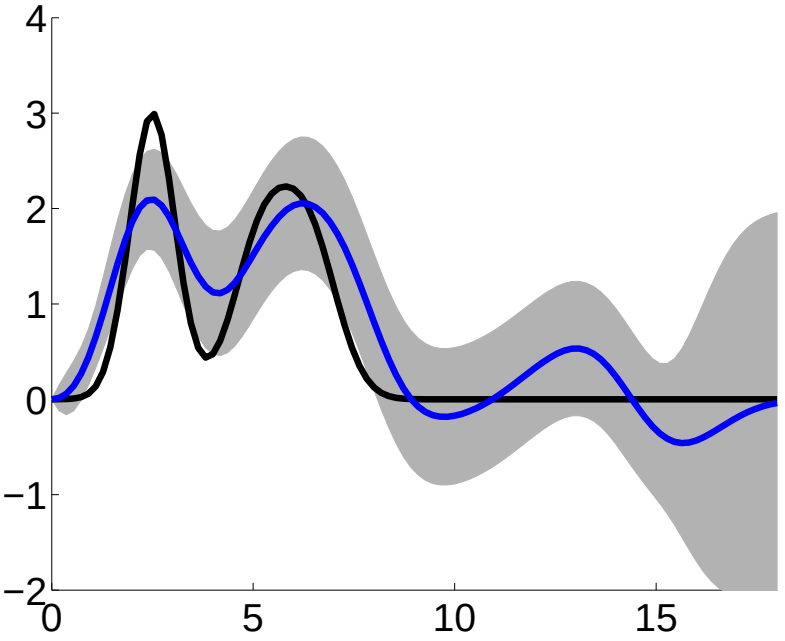
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



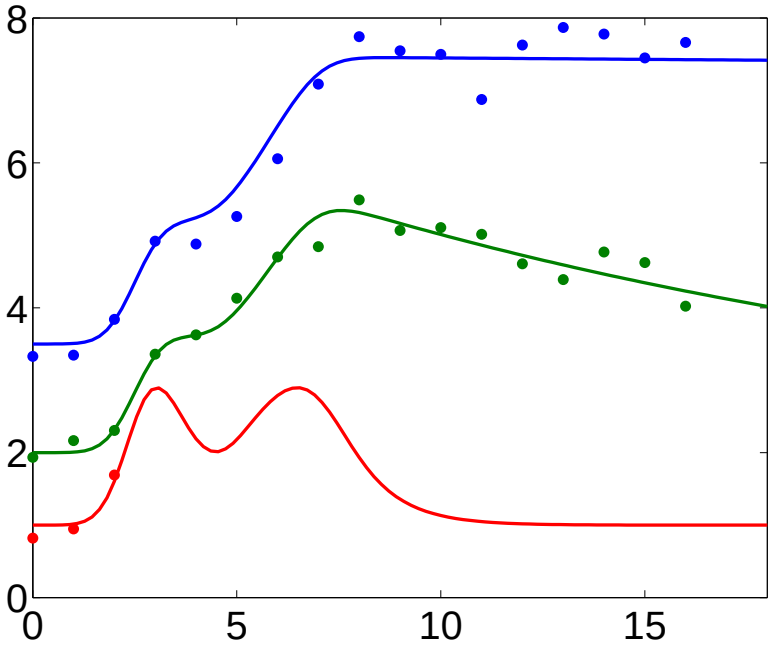
True "gene profiles" and noisy observations.



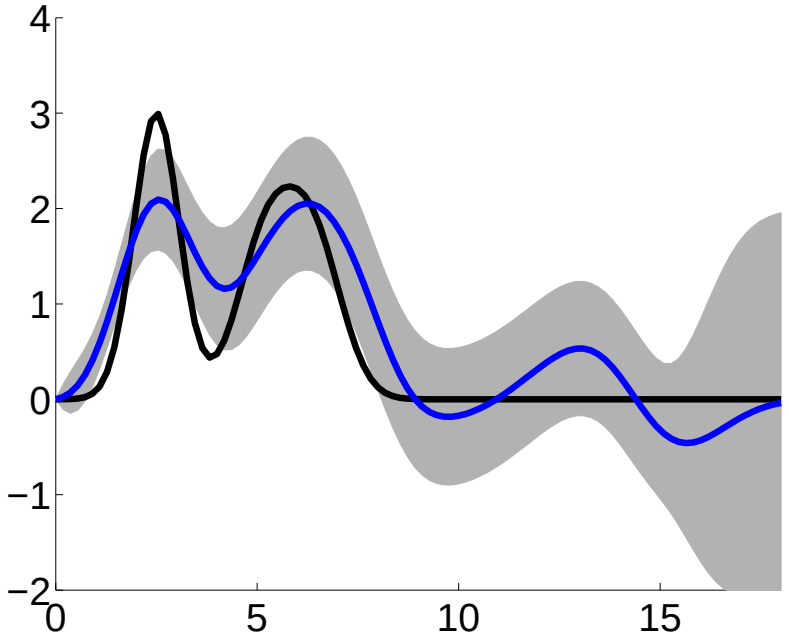
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



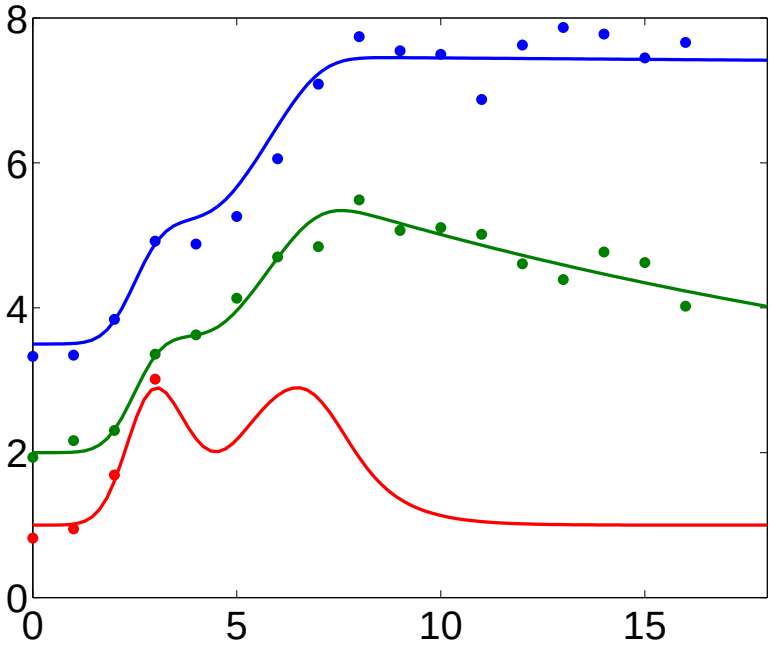
True "gene profiles" and noisy observations.



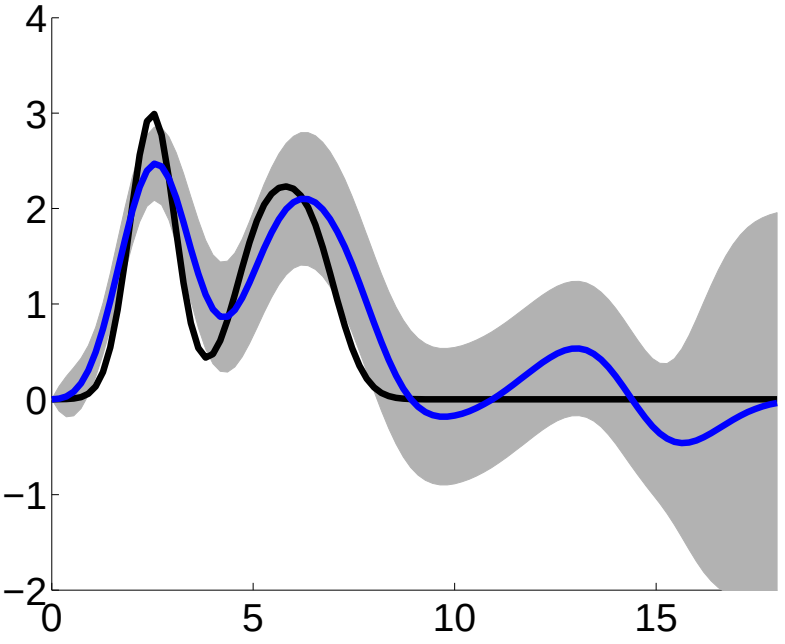
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



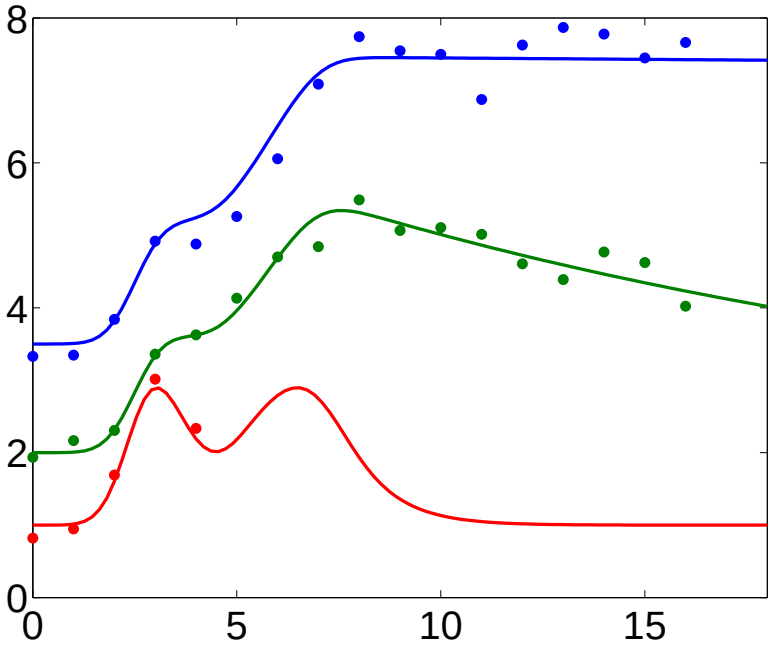
True "gene profiles" and noisy observations.



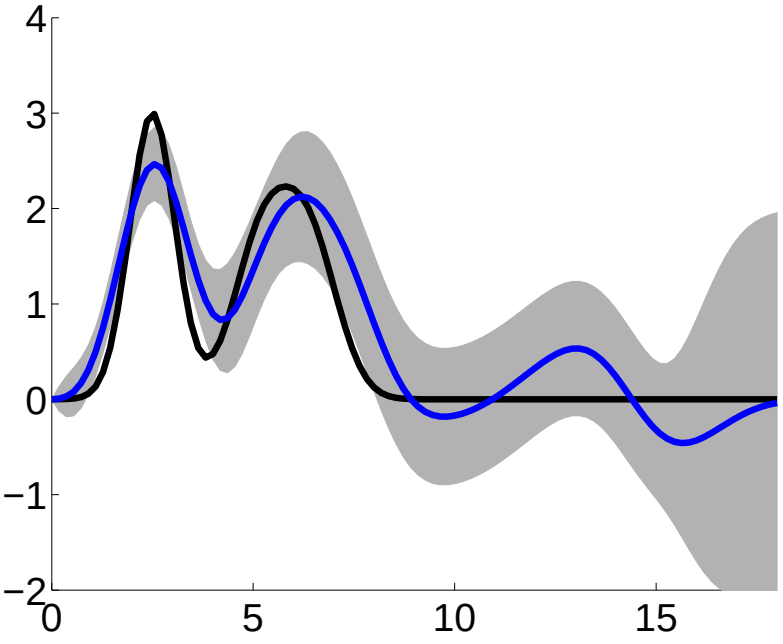
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



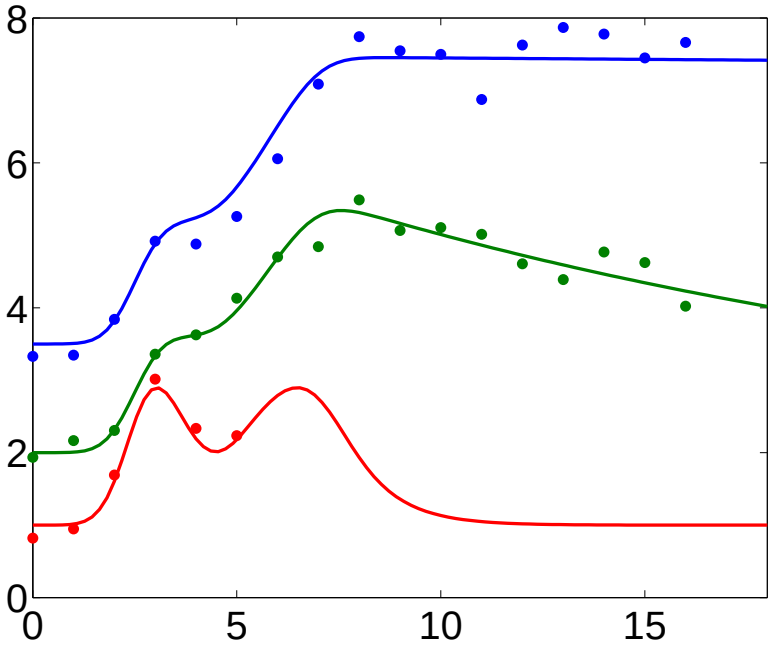
True "gene profiles" and noisy observations.



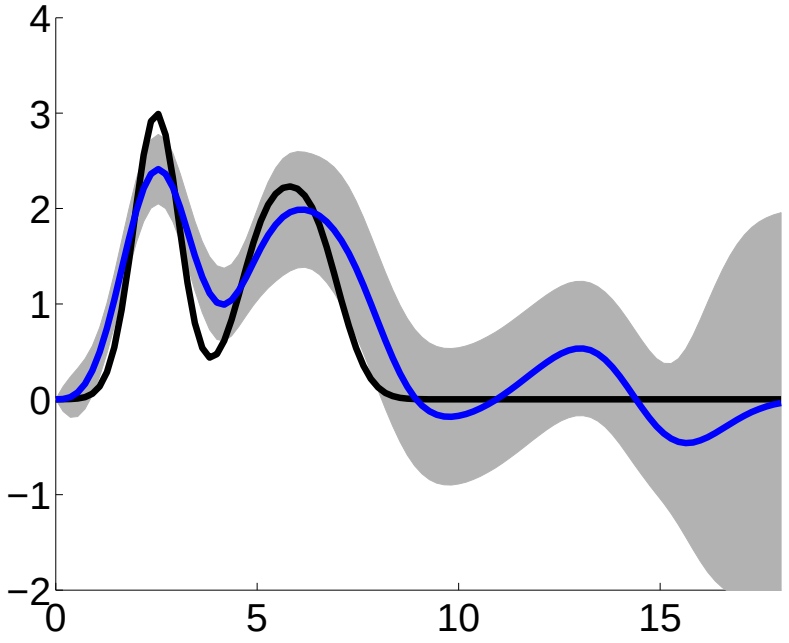
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



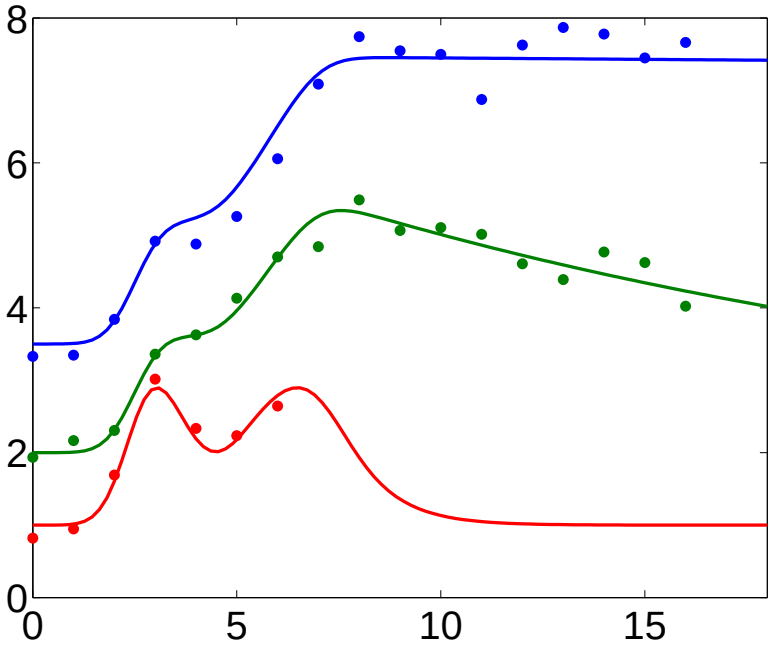
True "gene profiles" and noisy observations.



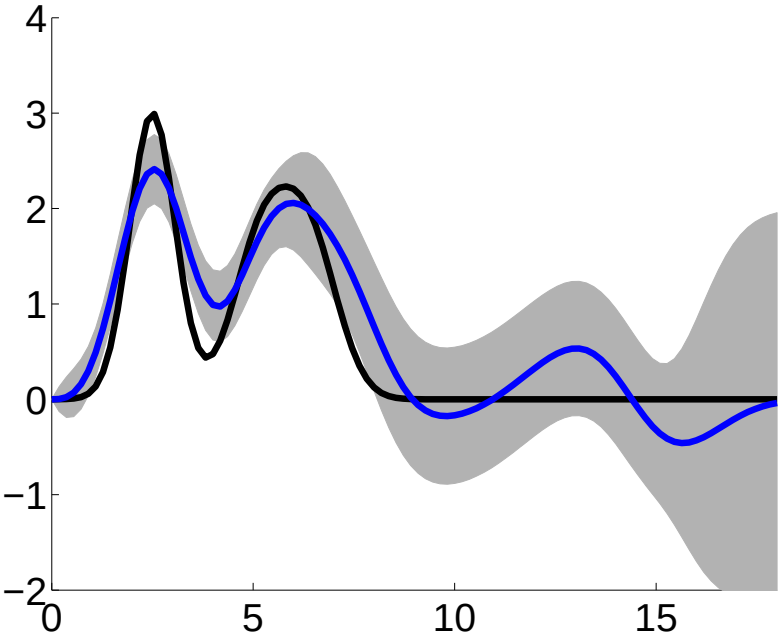
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



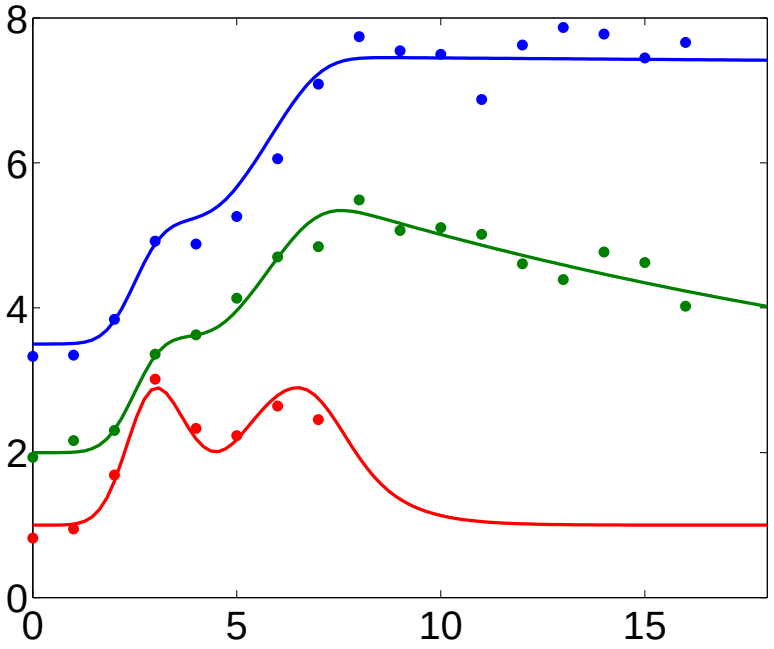
True "gene profiles" and noisy observations.



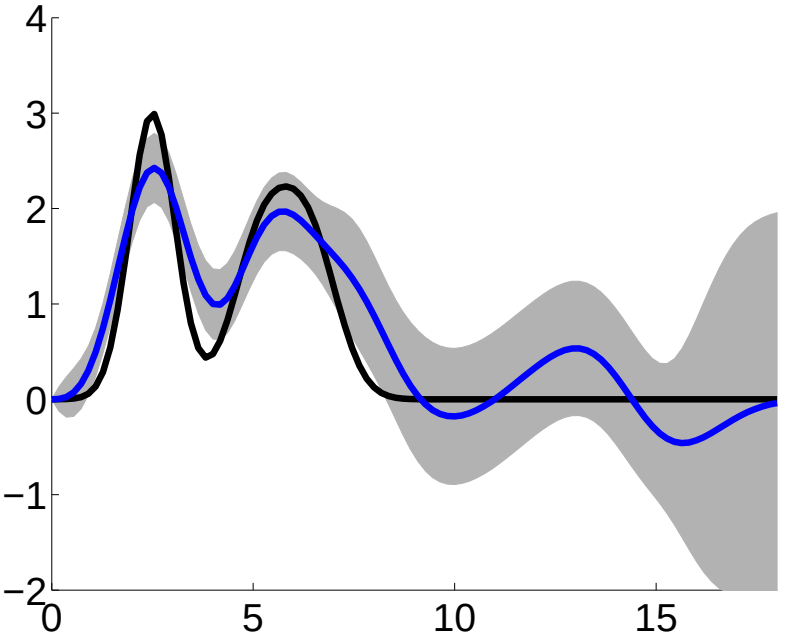
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



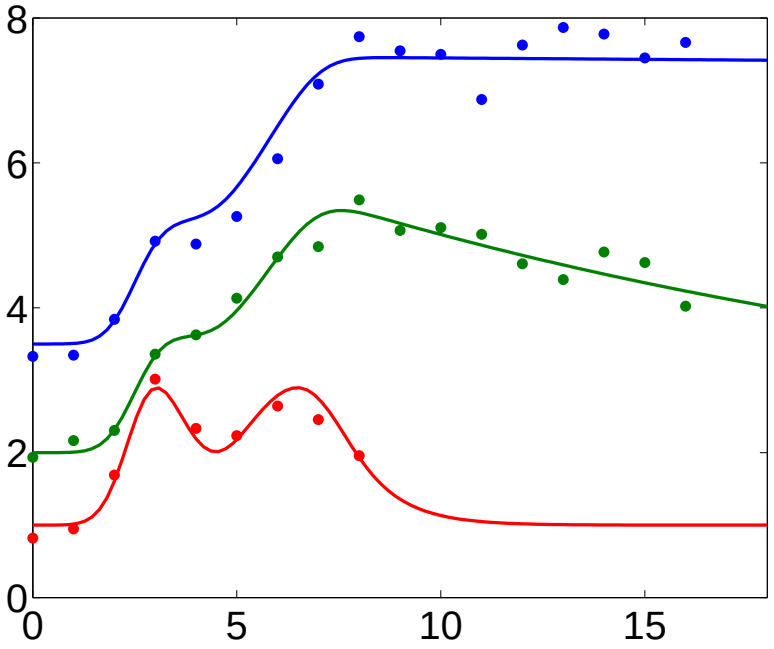
True "gene profiles" and noisy observations.



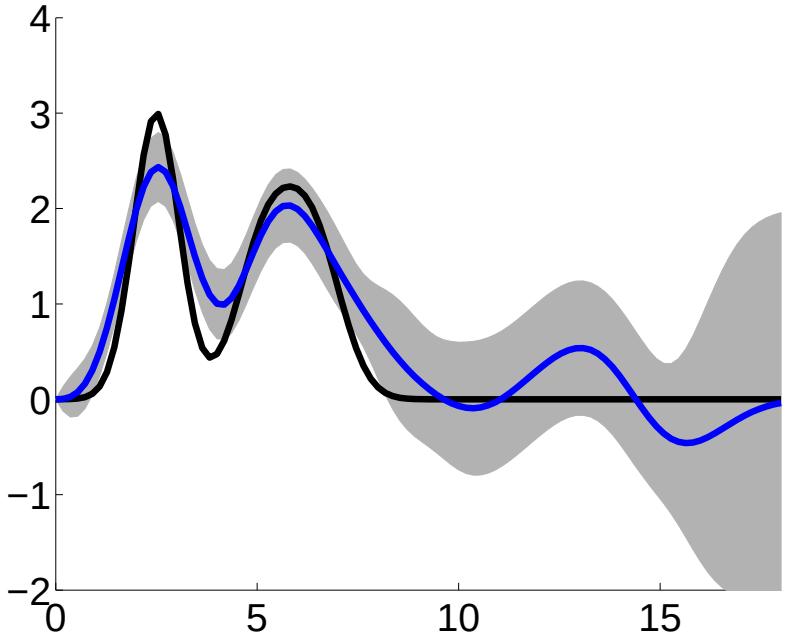
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



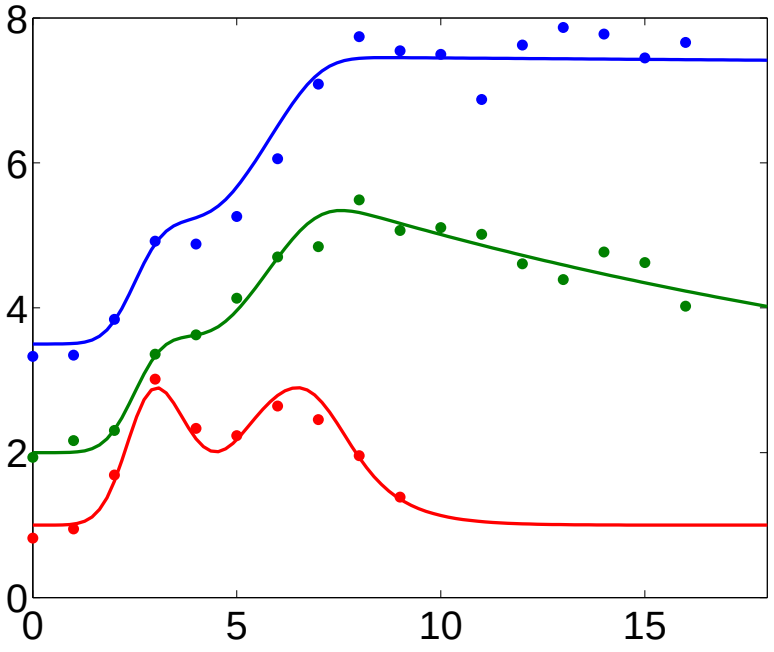
True "gene profiles" and noisy observations.



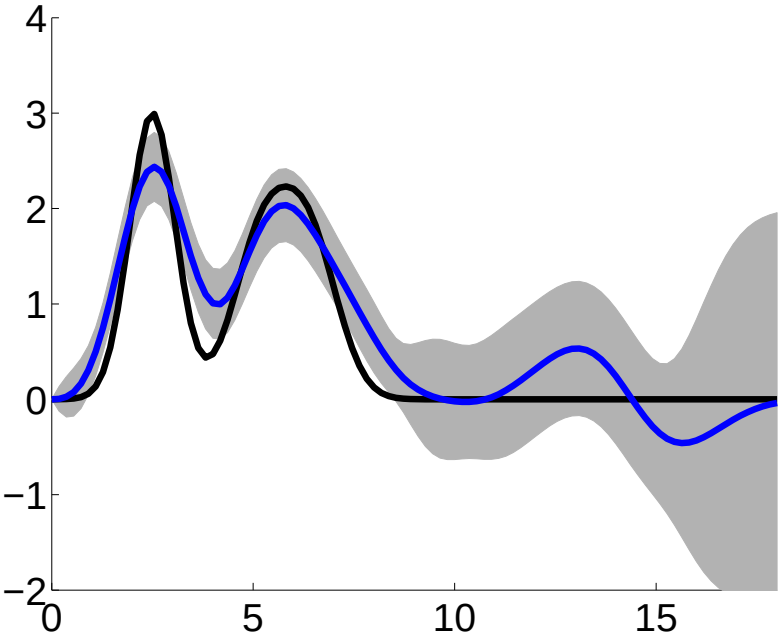
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



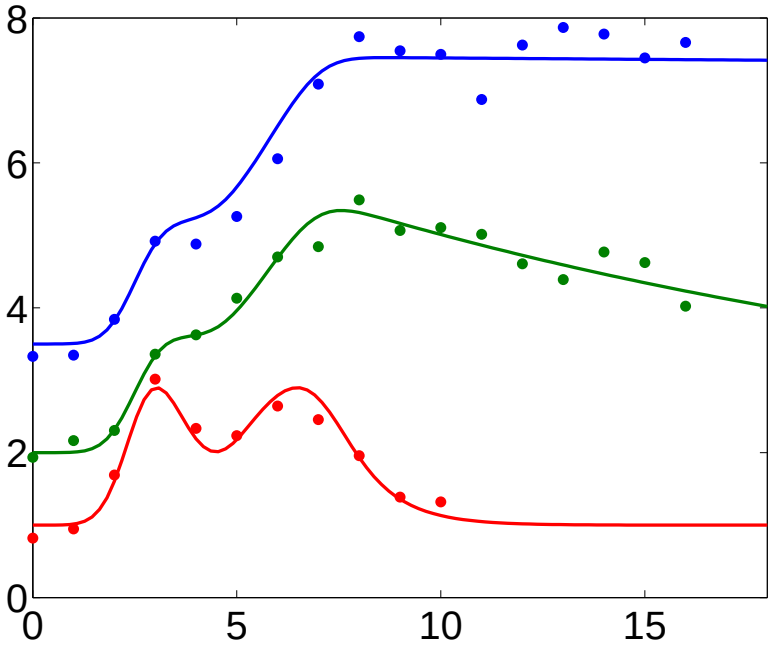
True "gene profiles" and noisy observations.



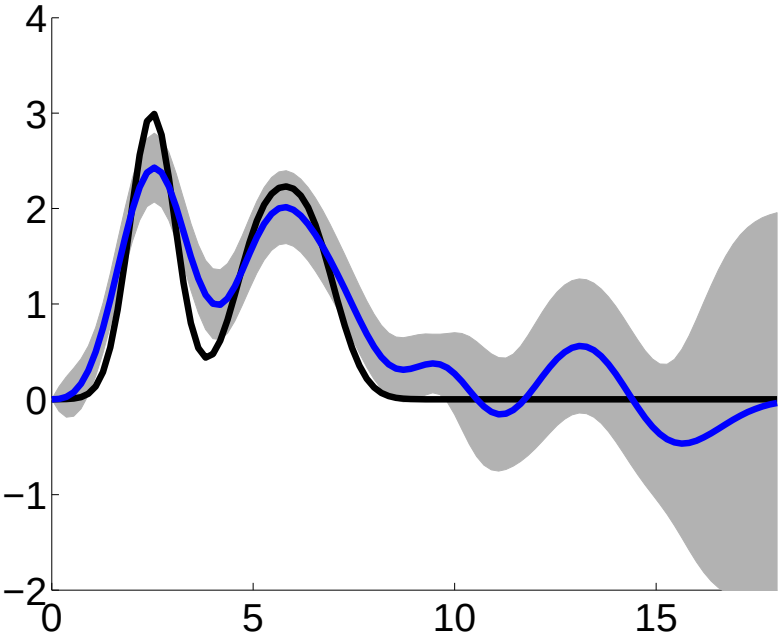
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



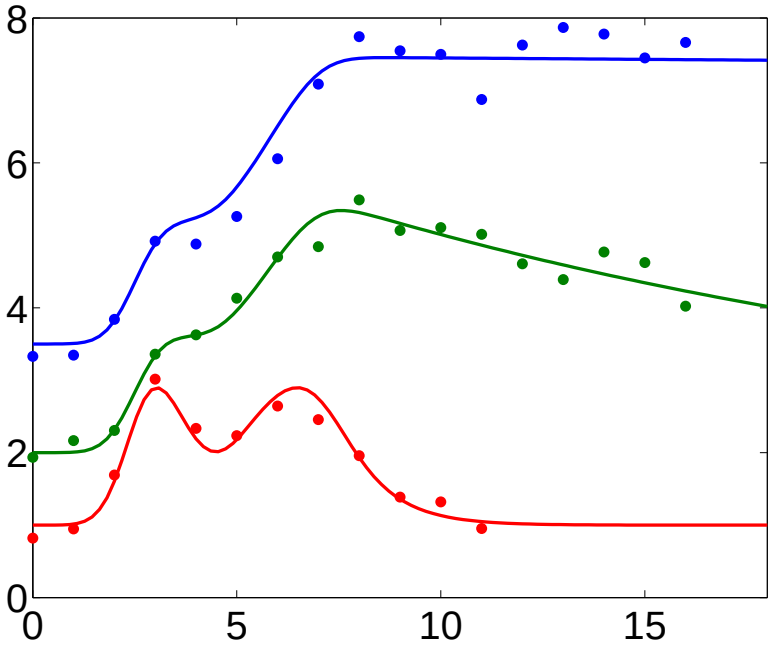
True "gene profiles" and noisy observations.



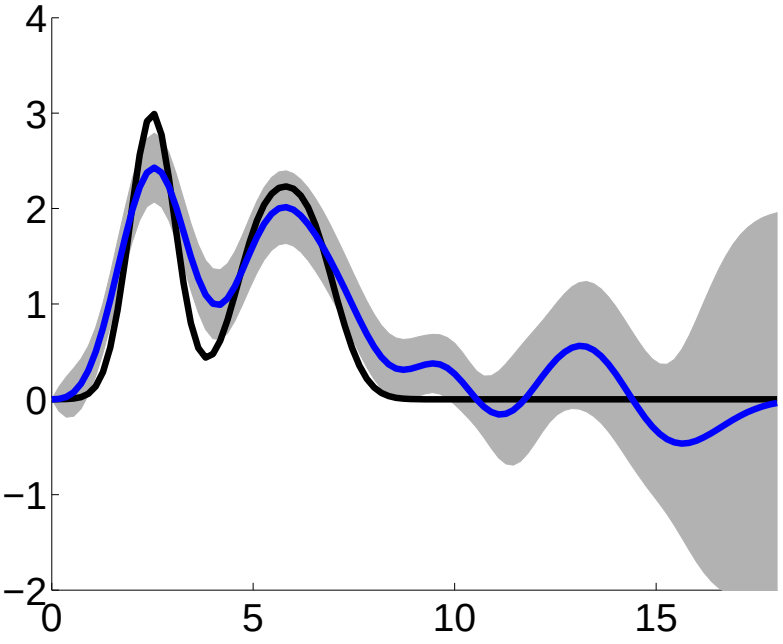
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



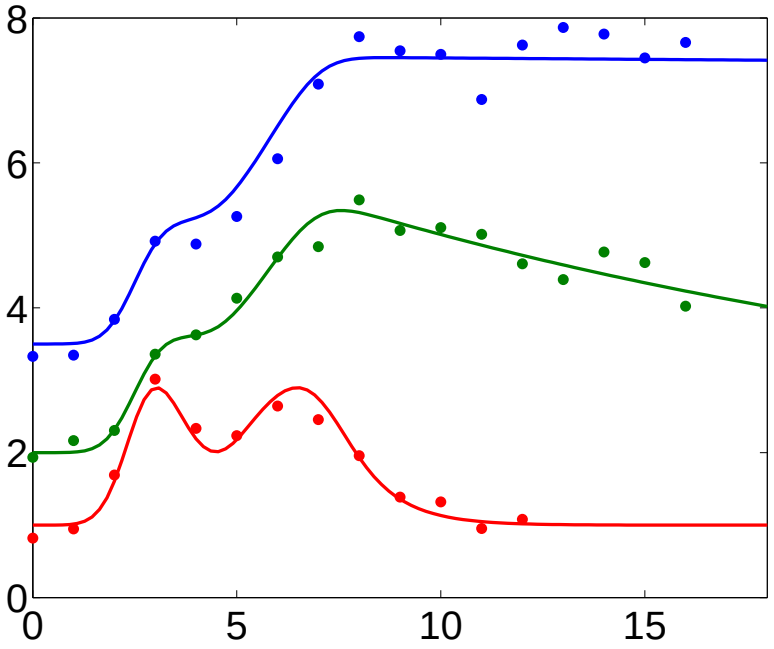
True "gene profiles" and noisy observations.



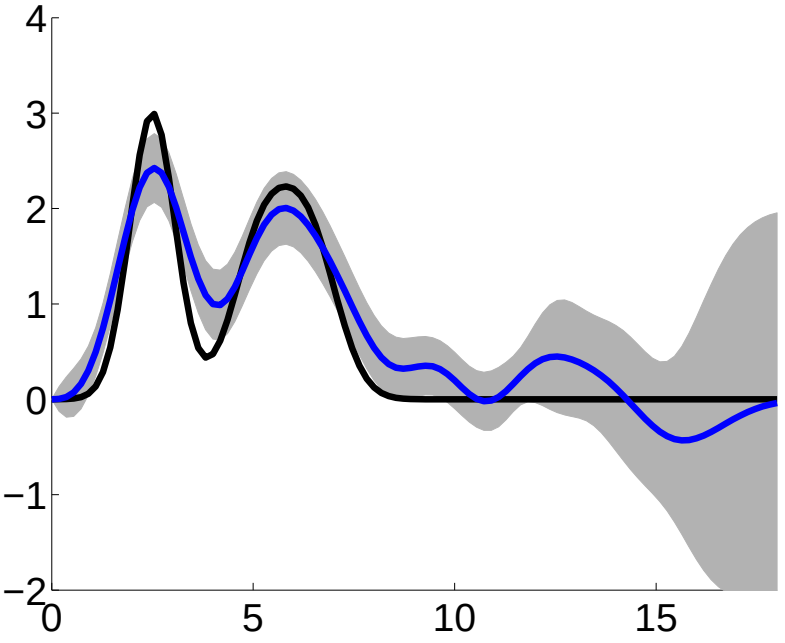
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



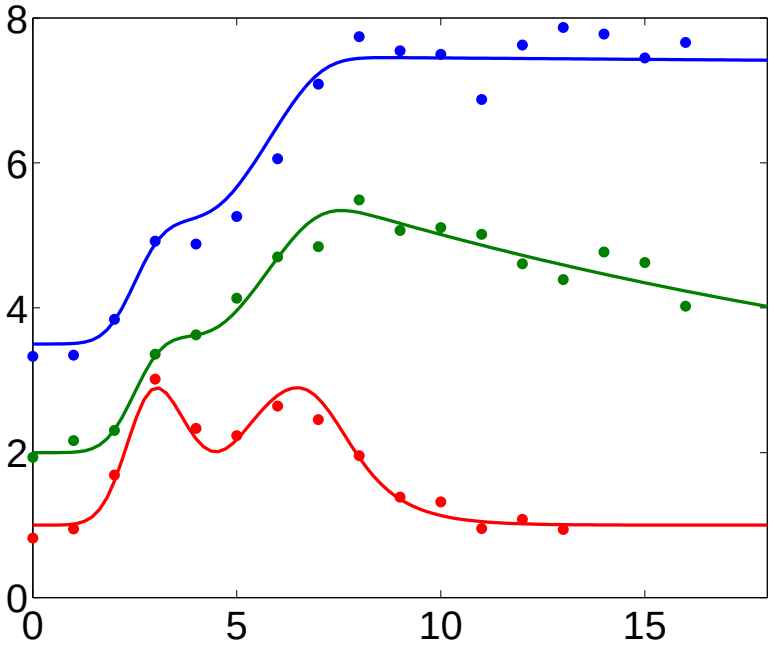
True "gene profiles" and noisy observations.



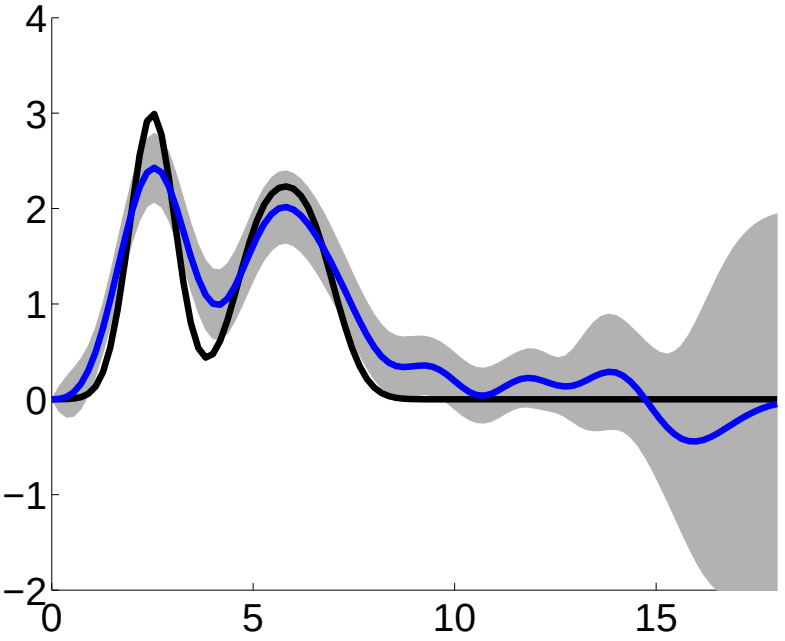
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



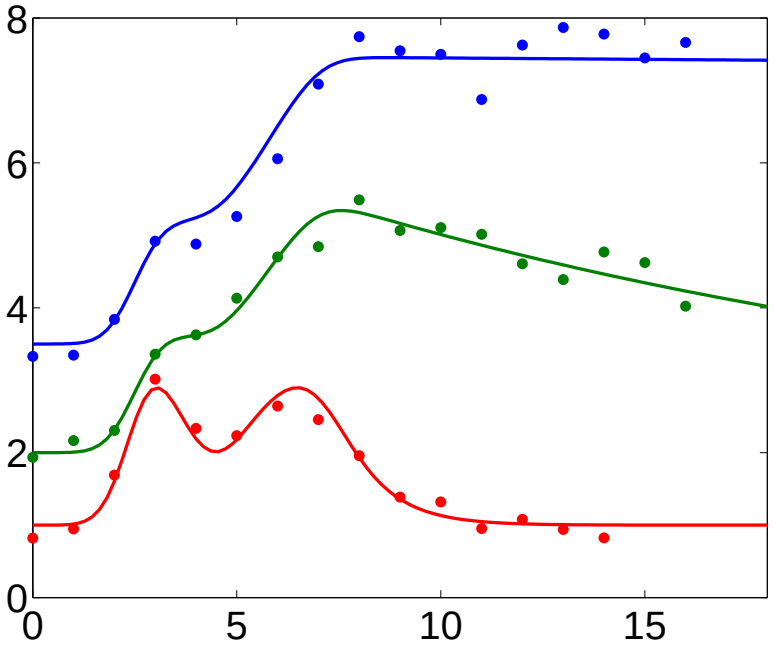
True "gene profiles" and noisy observations.



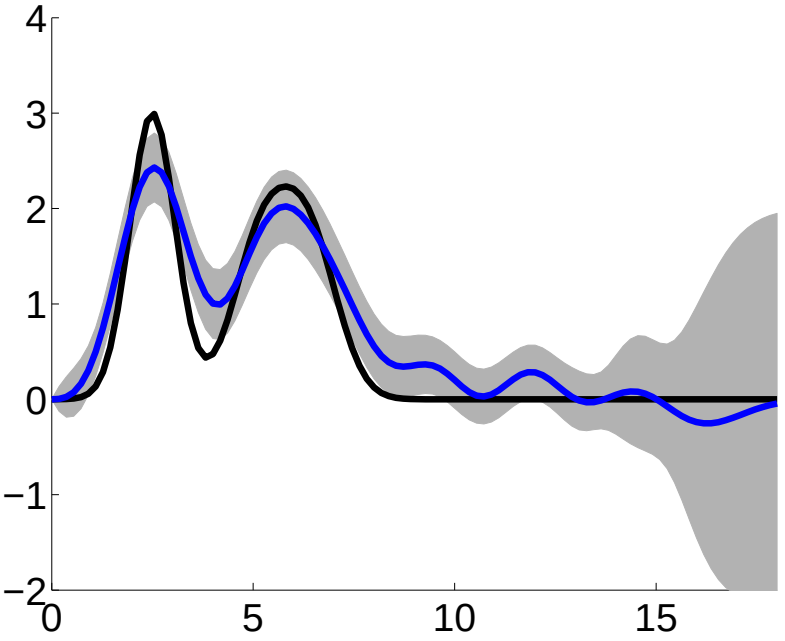
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



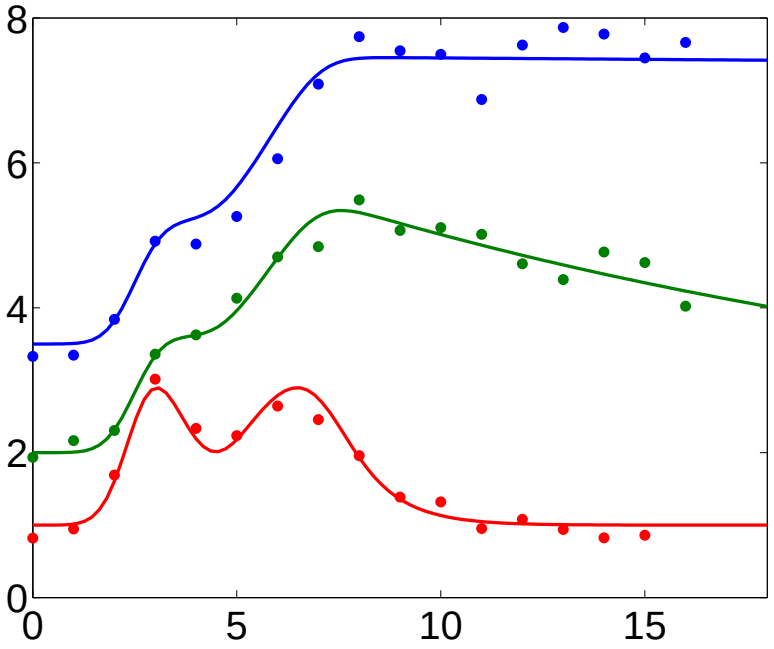
True "gene profiles" and noisy observations.



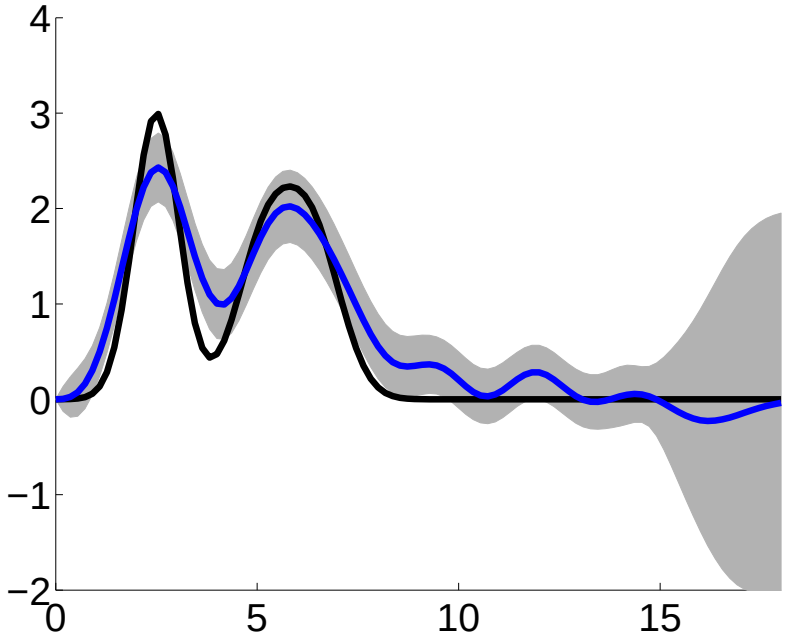
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



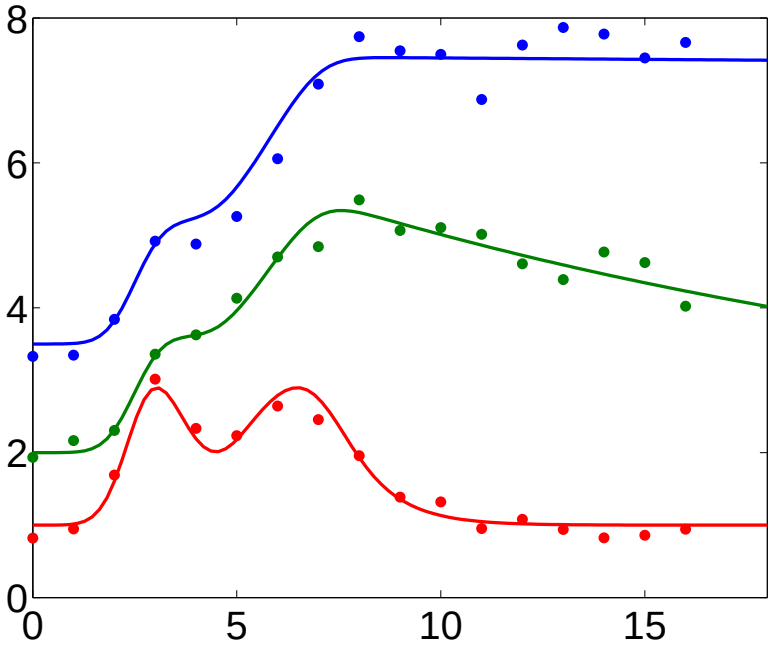
True "gene profiles" and noisy observations.



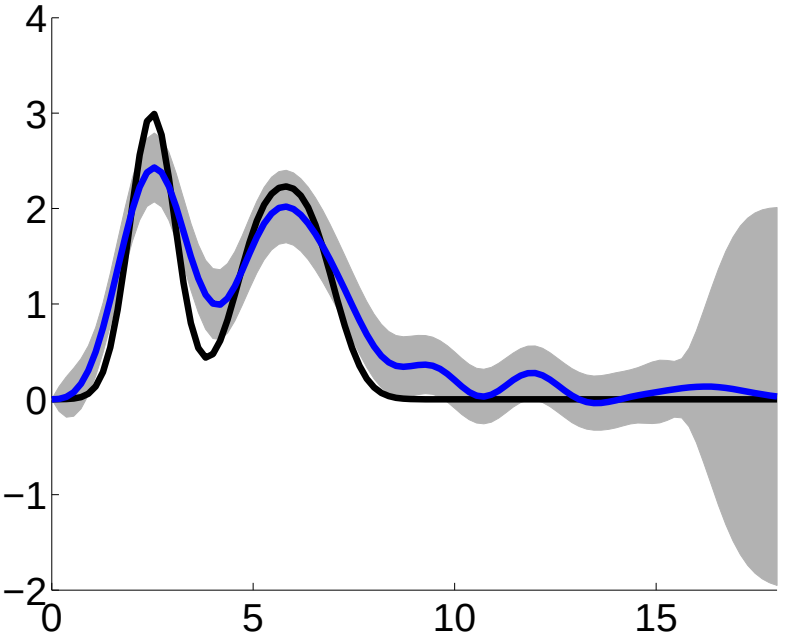
Inferred transcription factor activity.

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

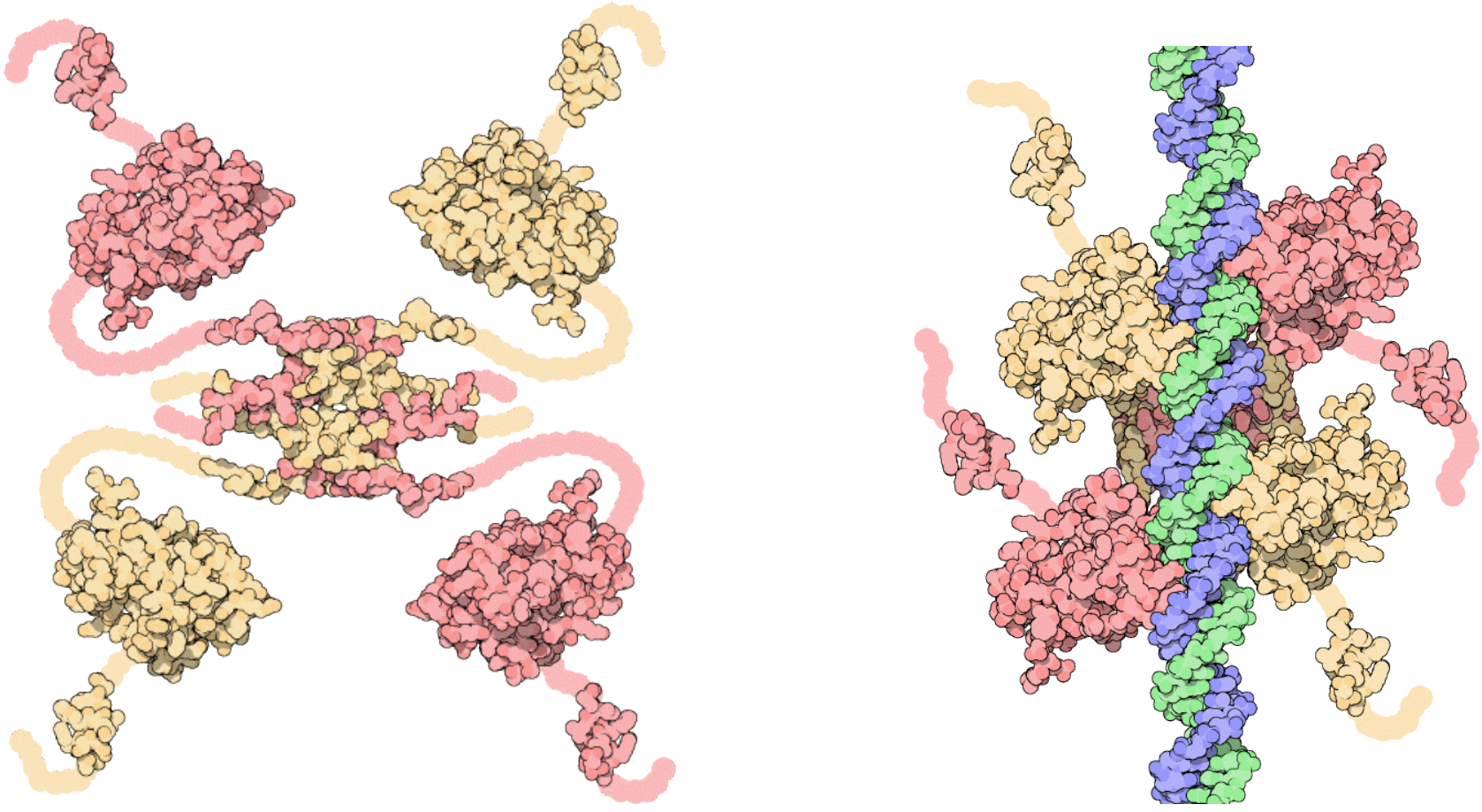


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the"Molecule of the Month" feature).

p53

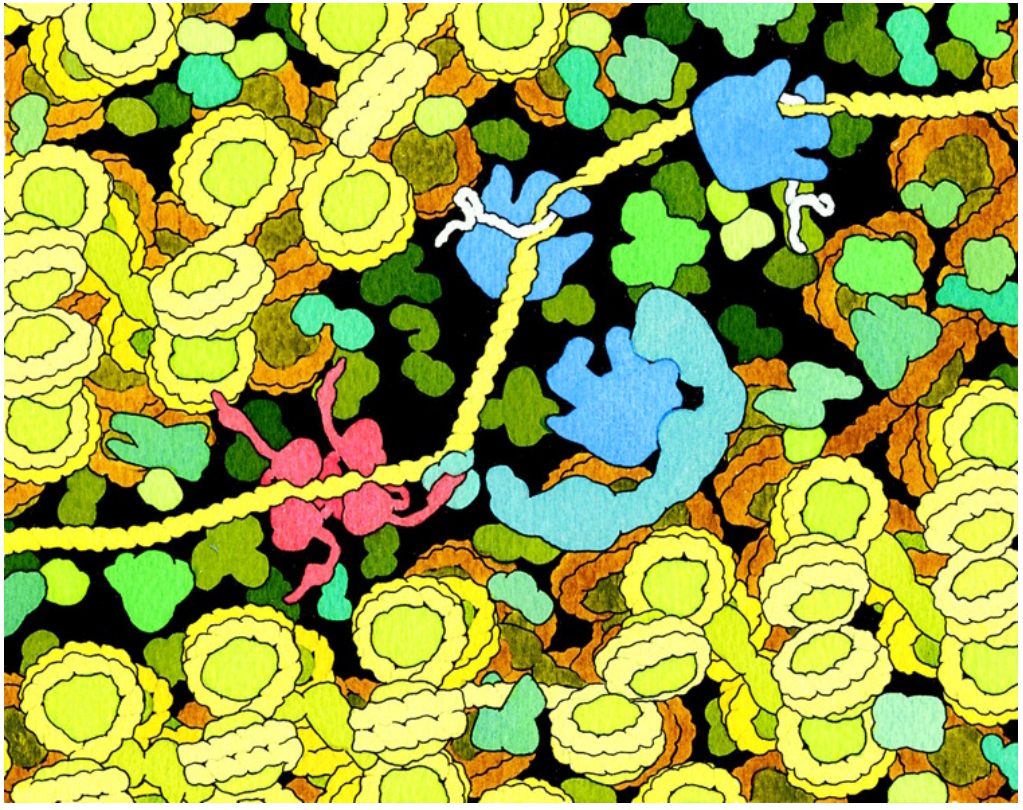


Figure: Repair of DNA damage by p53. Image from Goodsell (1999).

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).

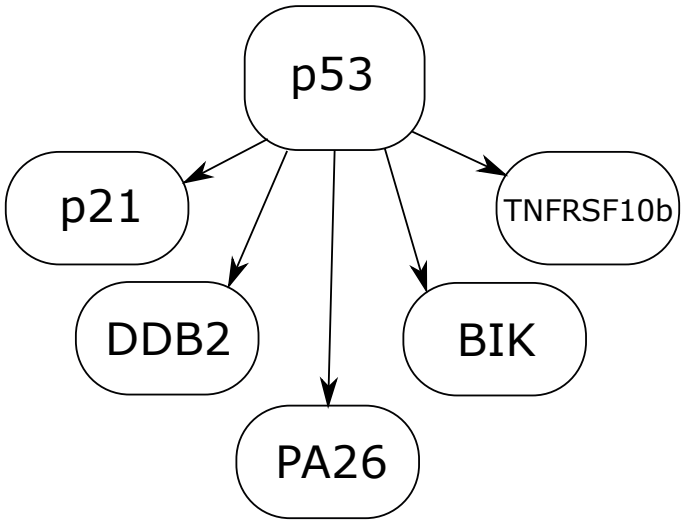


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

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

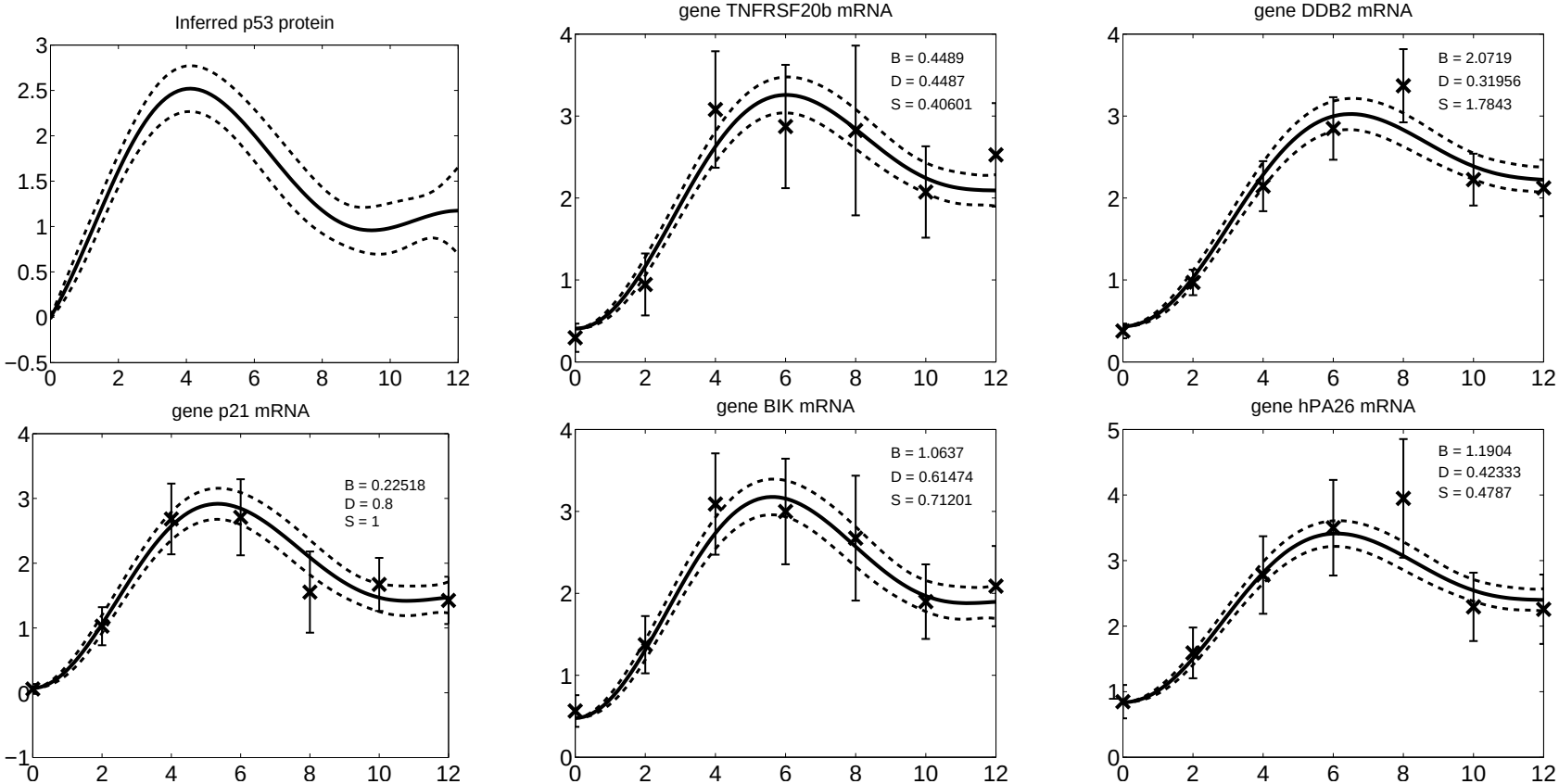
ABSTRACT

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

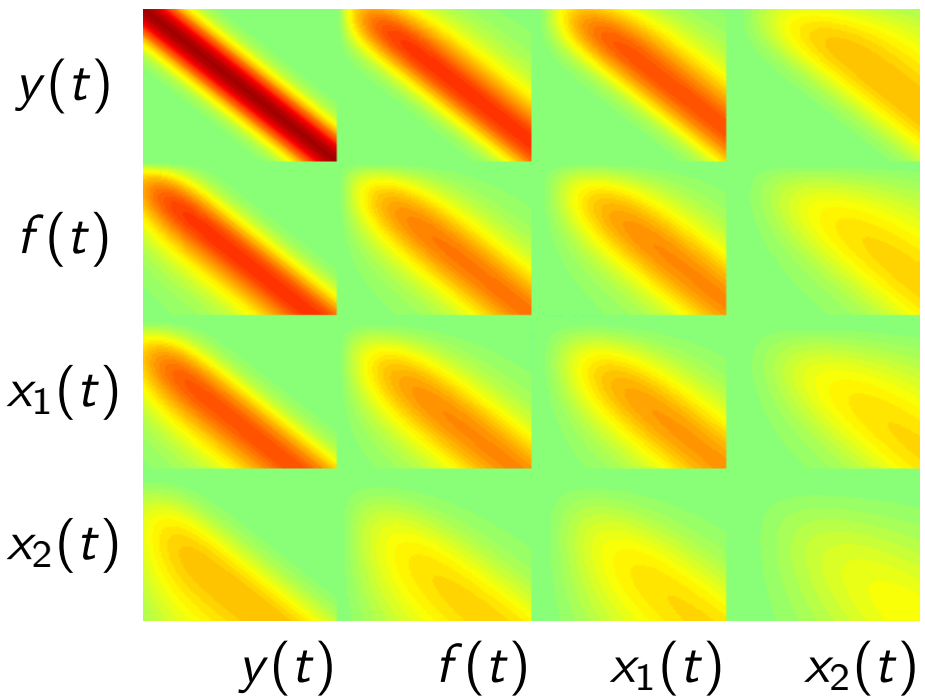
RBF covariance function for $y(t)$

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

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

- 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`

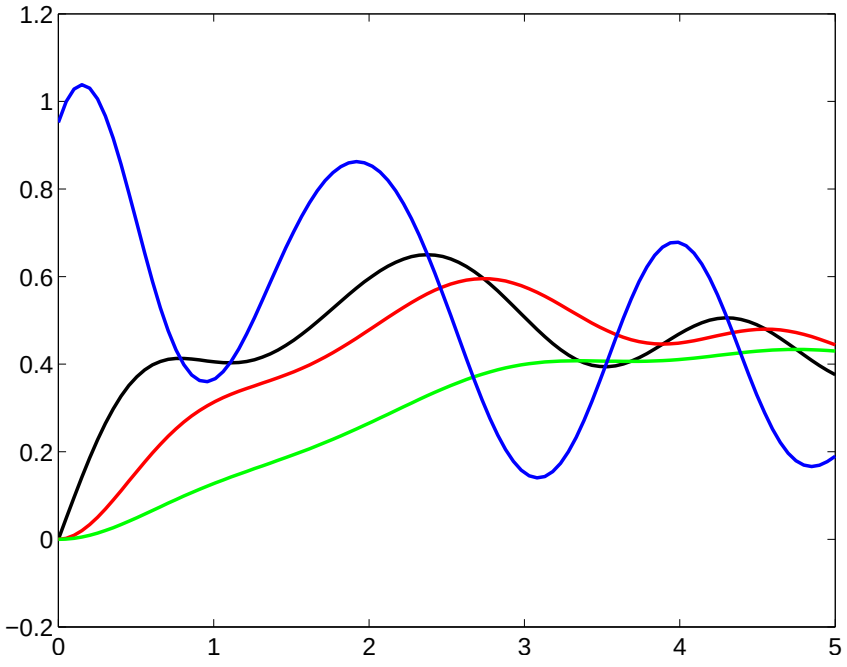


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

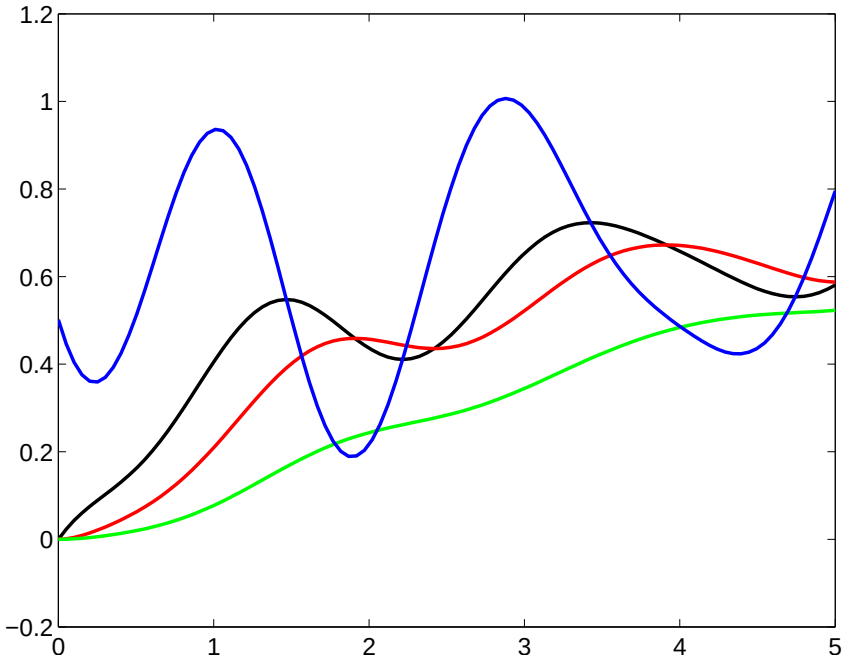


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

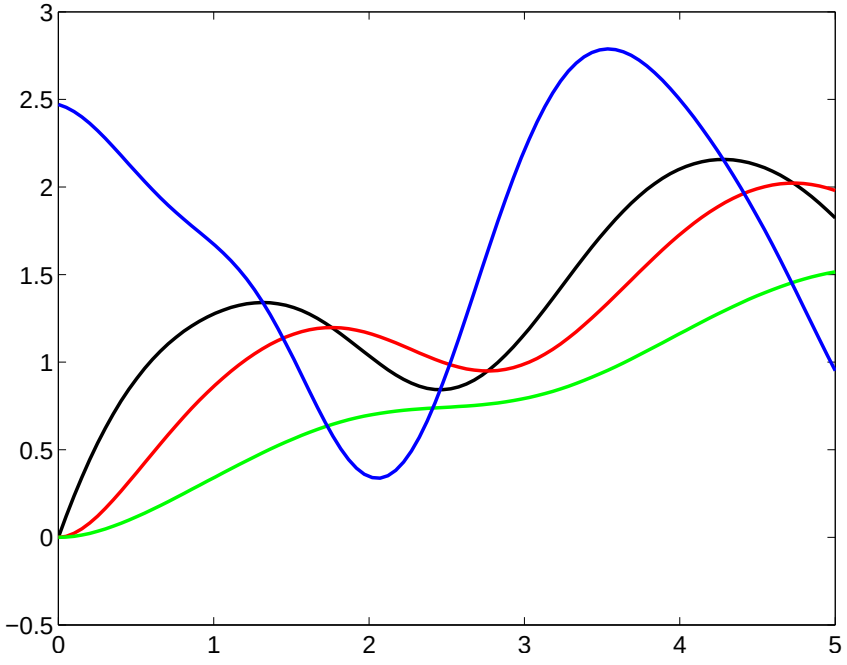


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

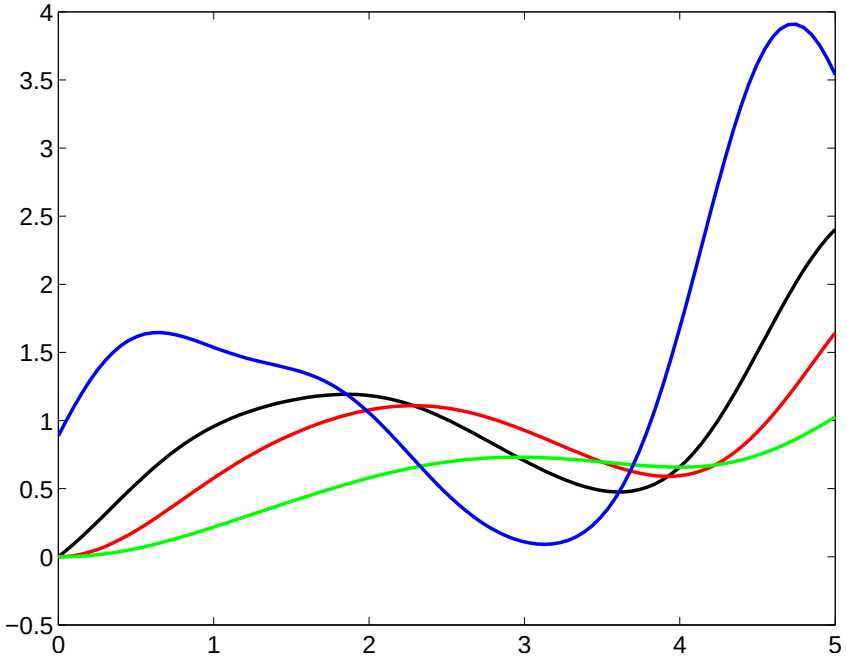


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

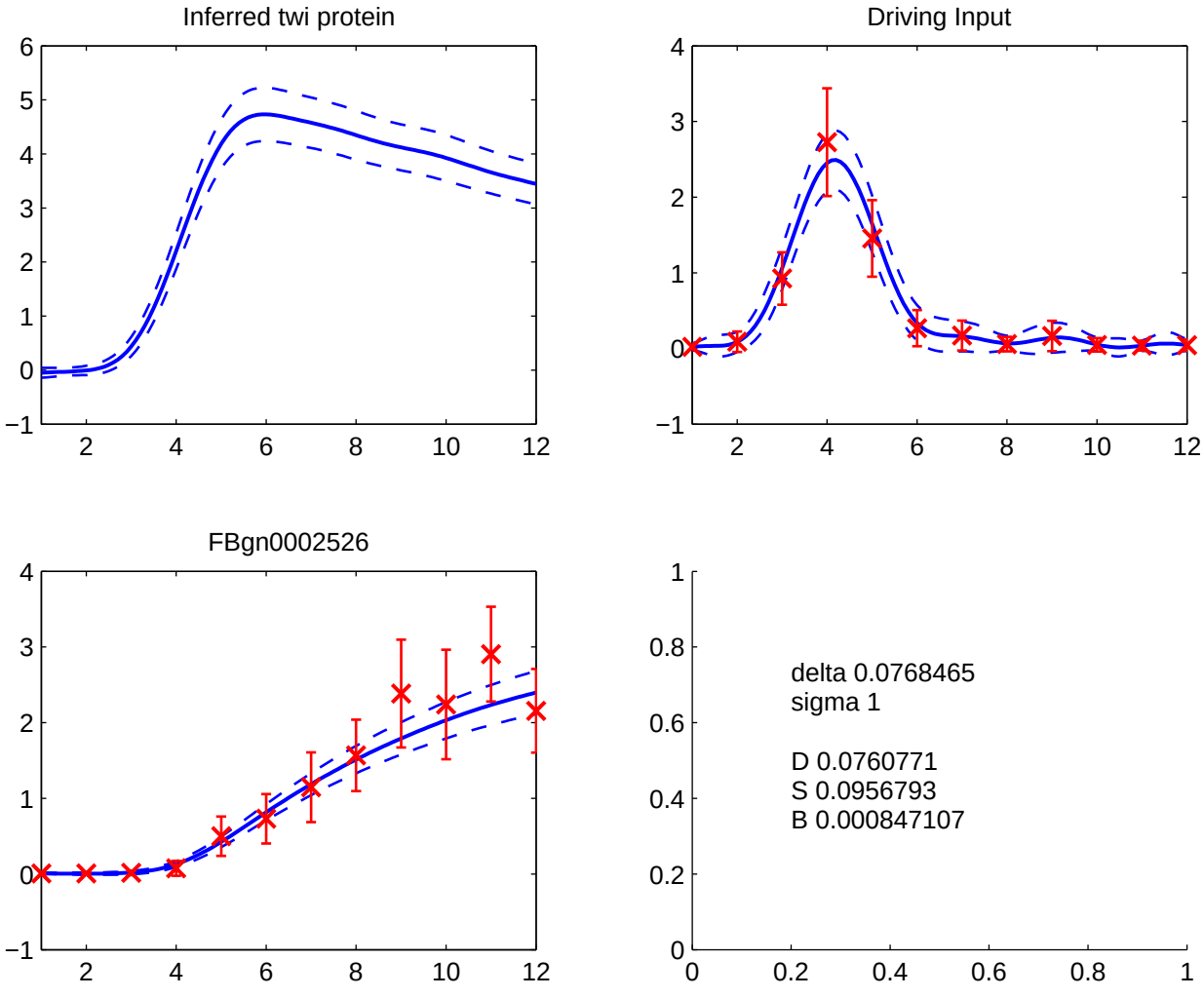


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

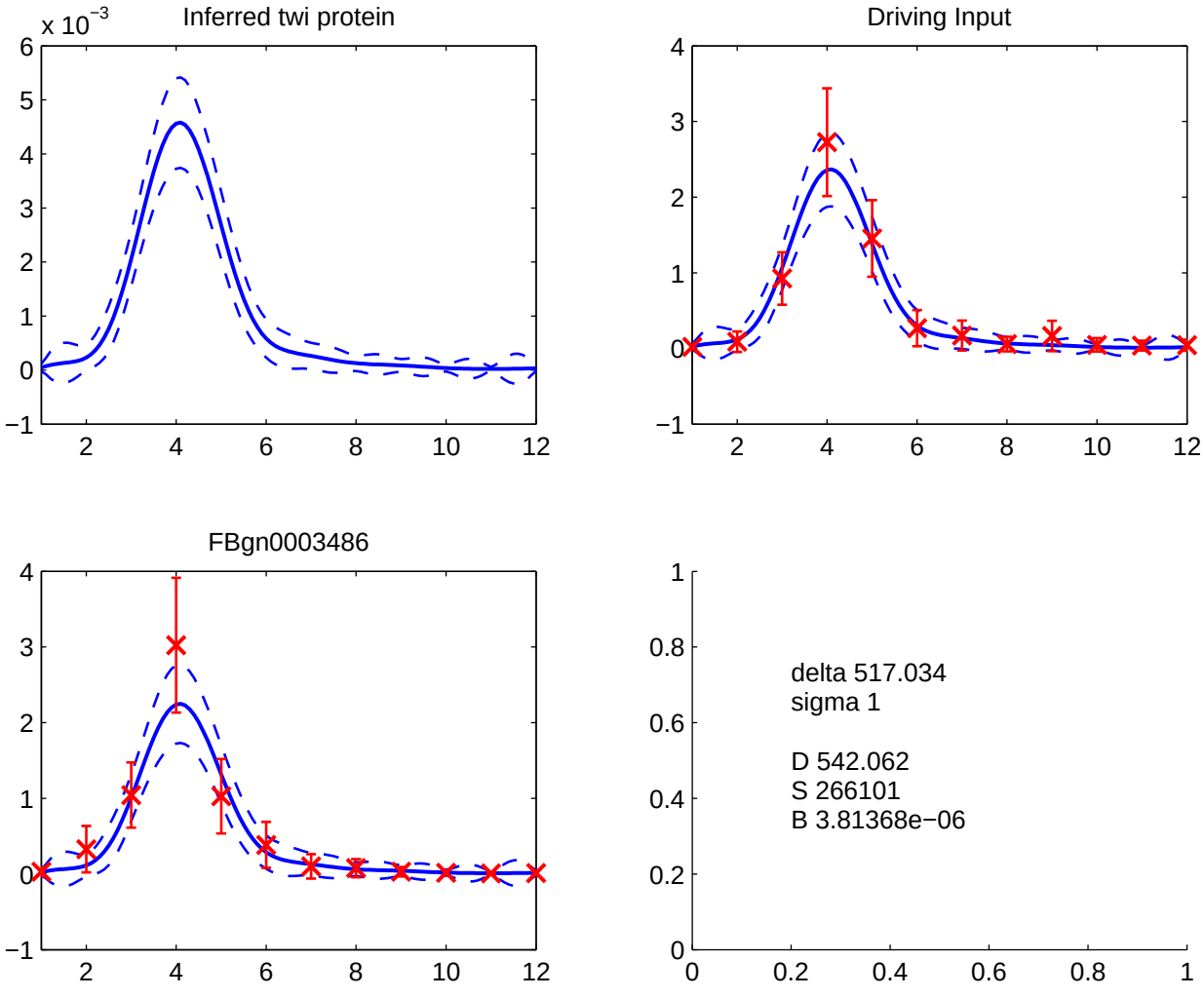


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

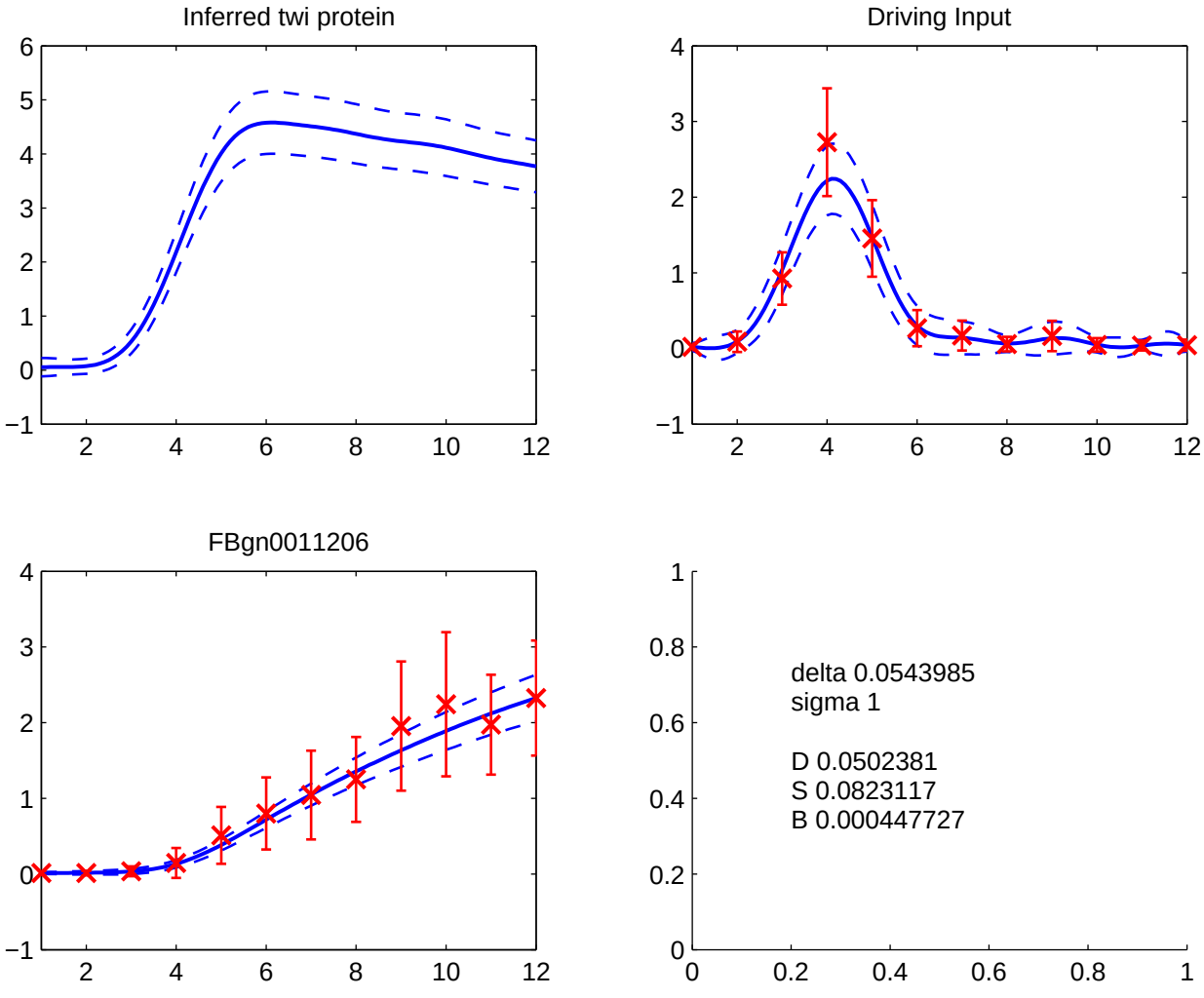


Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

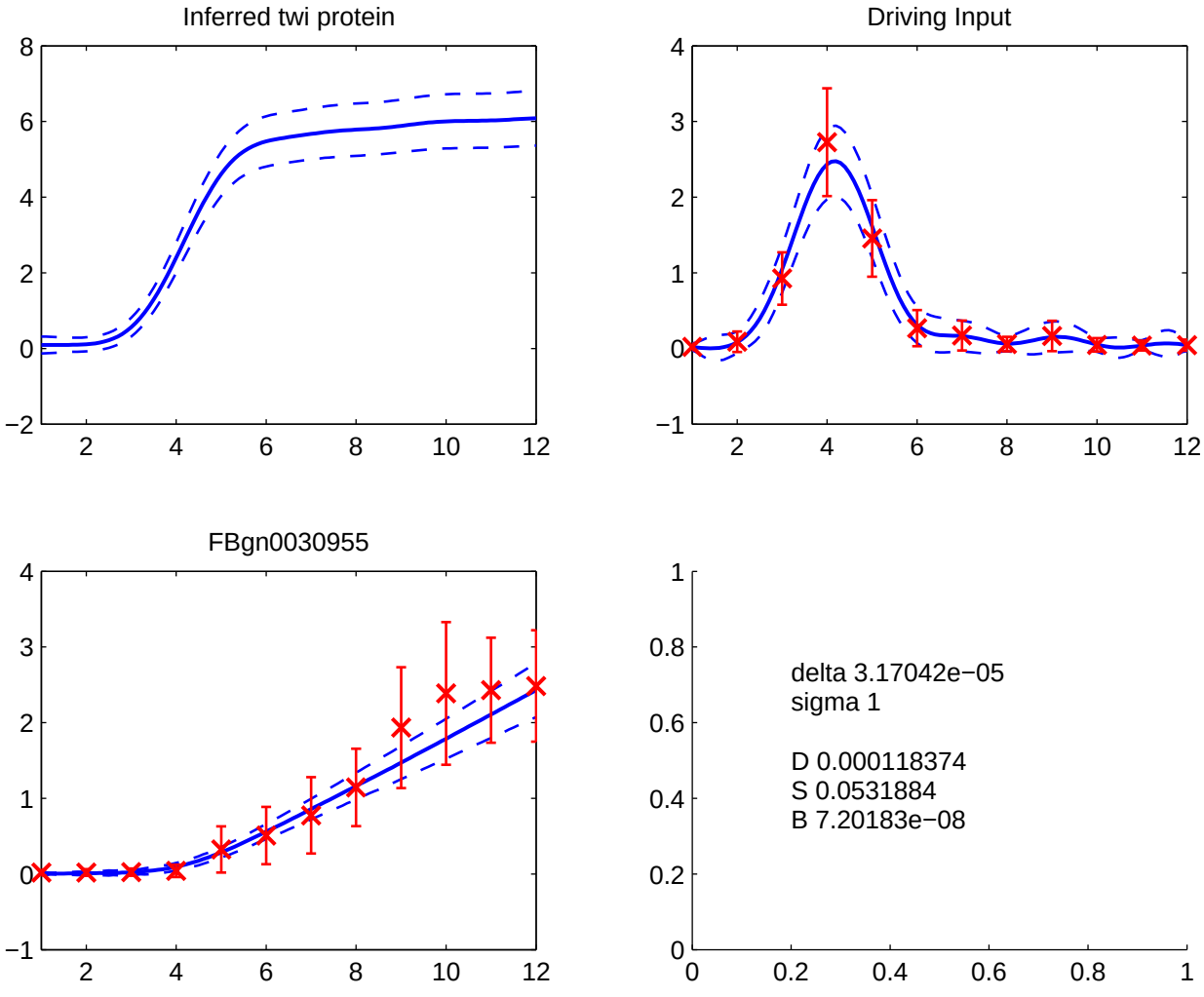


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

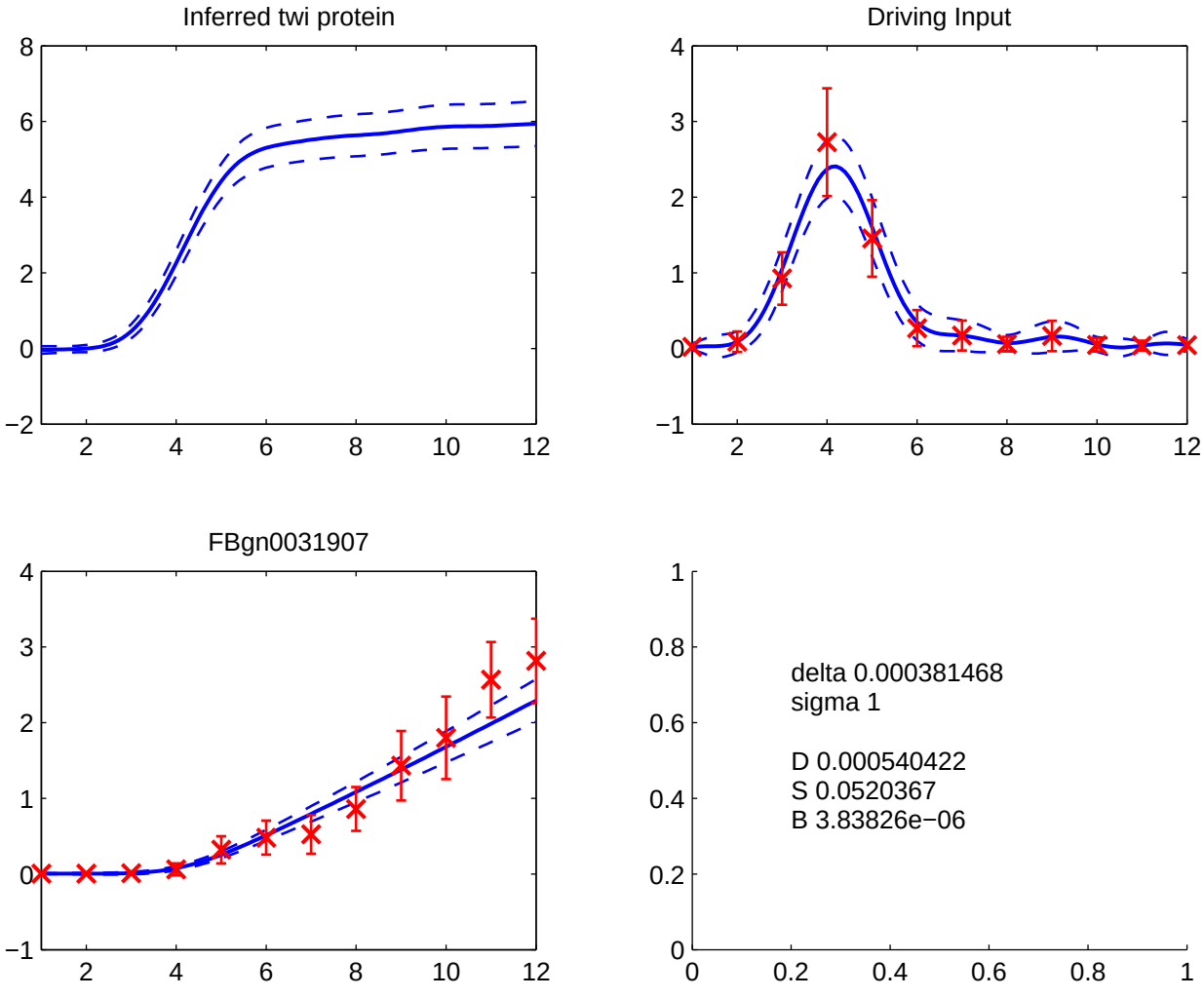


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

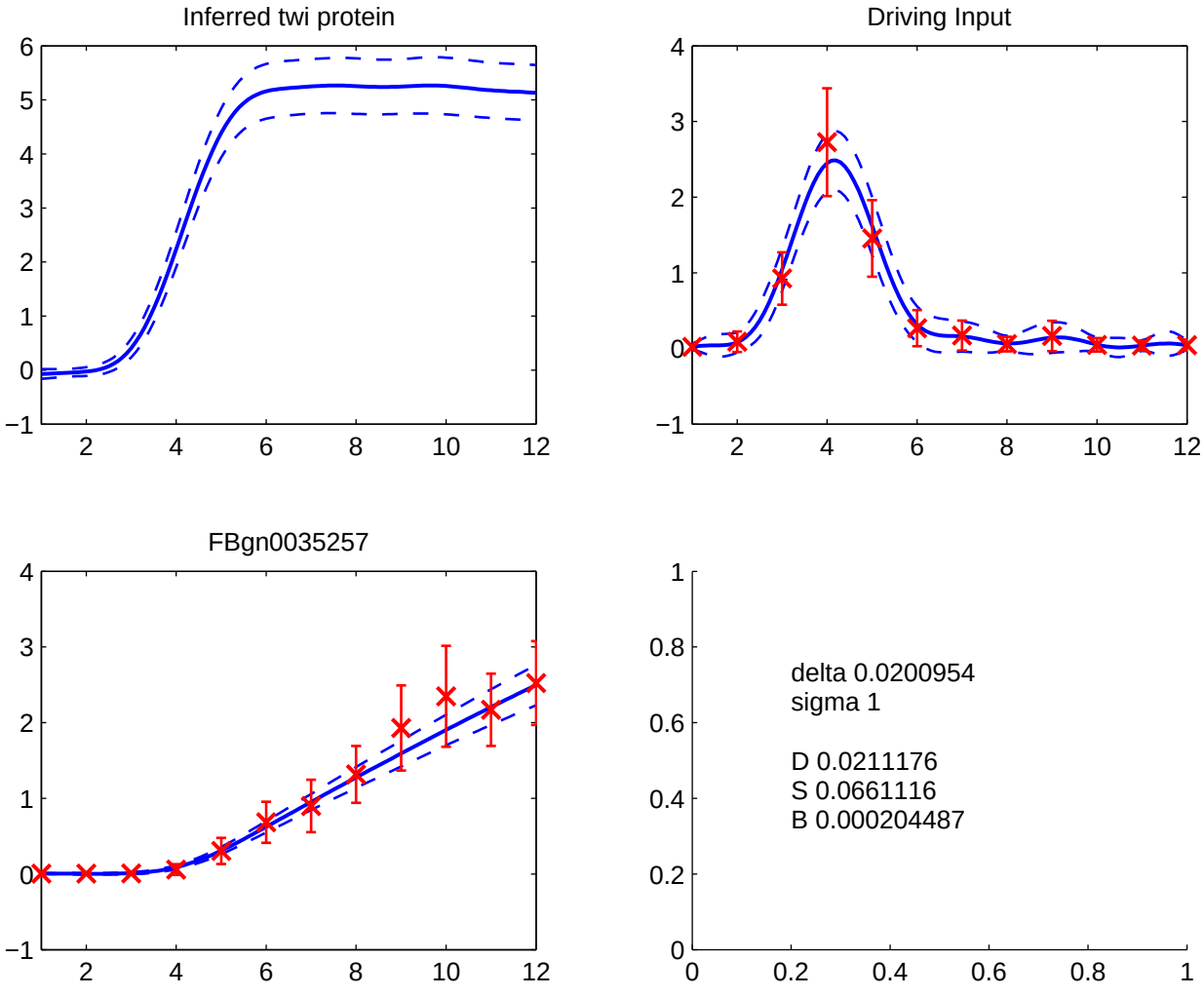


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

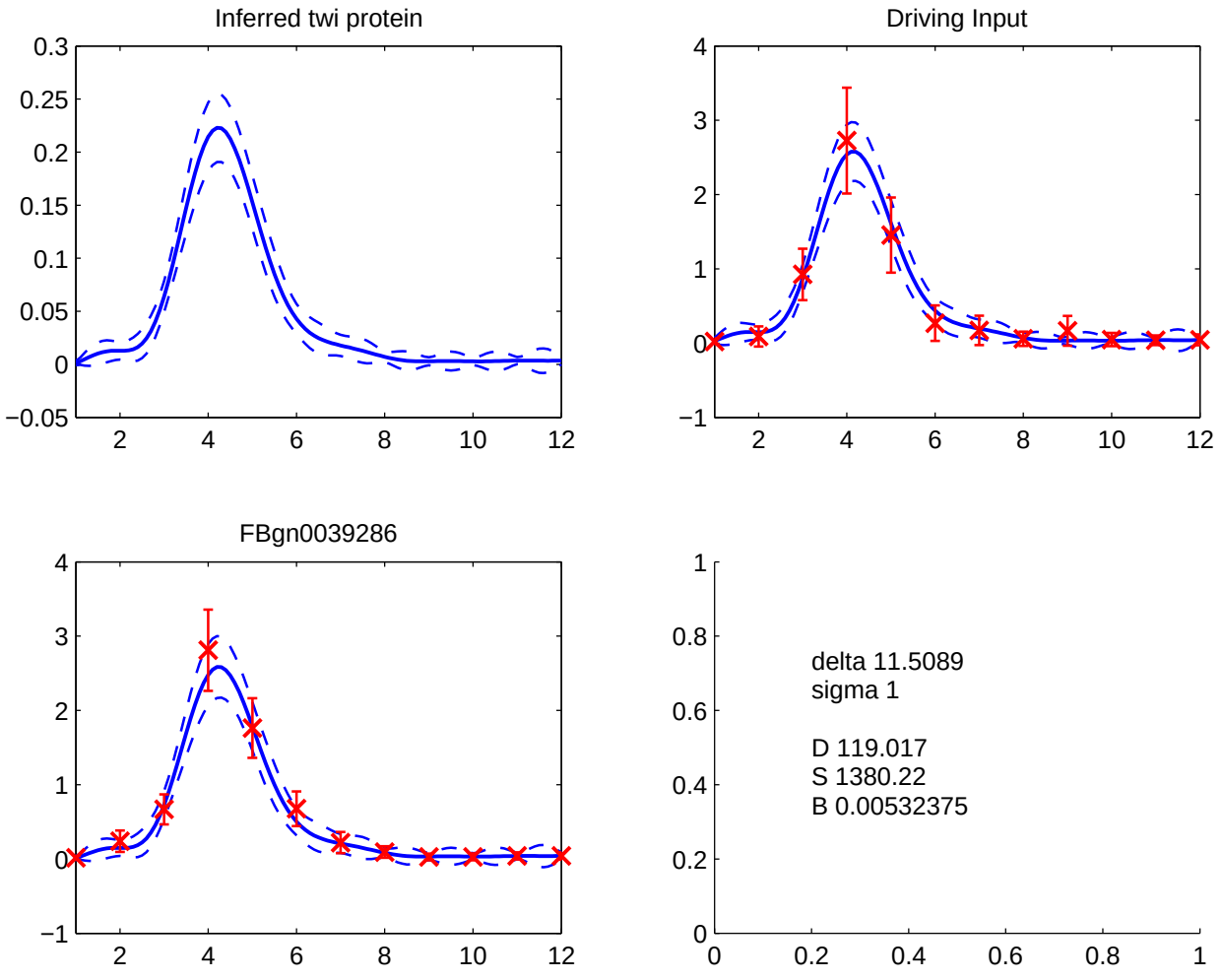
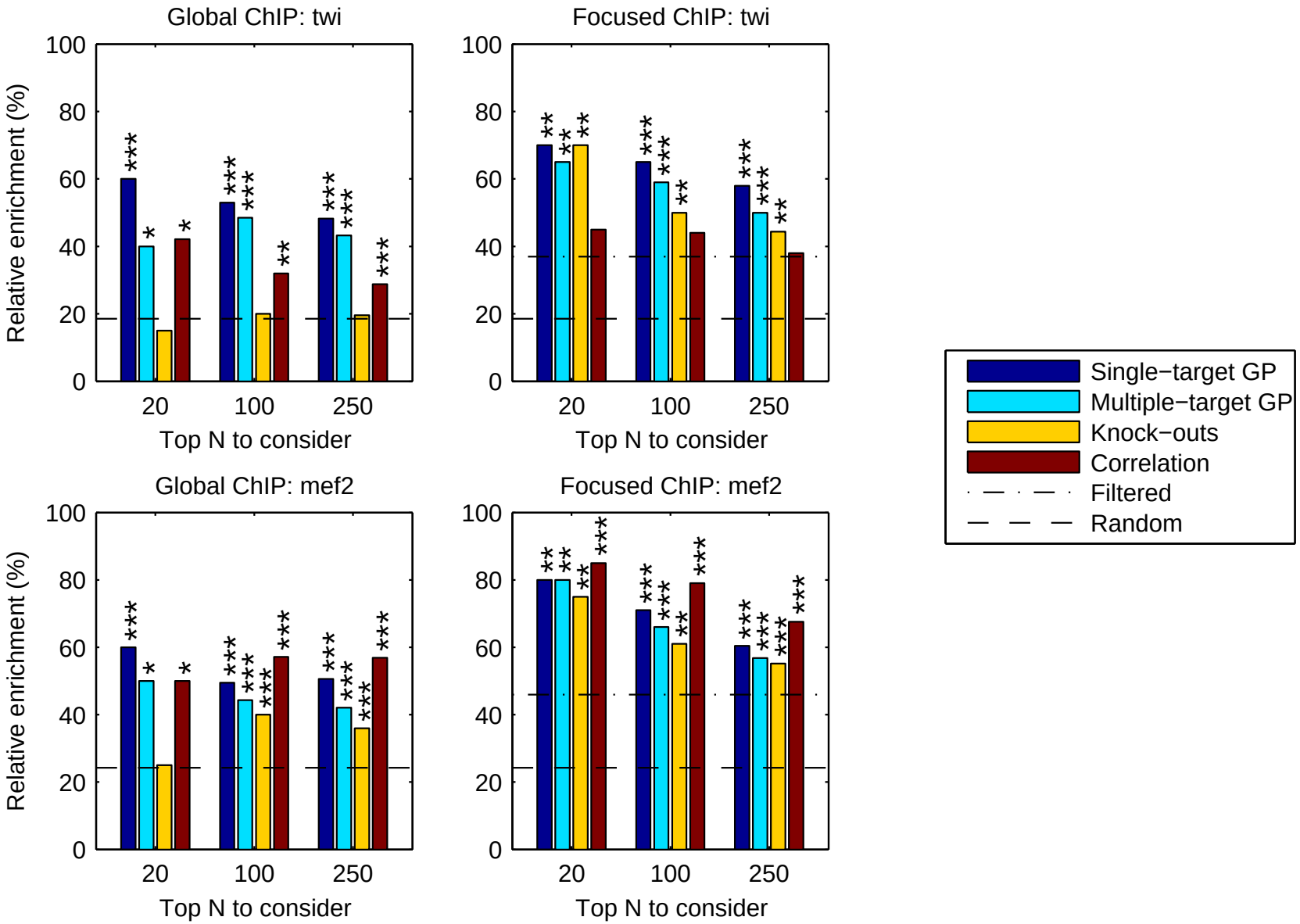


Figure: Model for flybase gene identity FBgn0039286.

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. (2011b)
  - Robotics applications.
  - Applications to other types of system, *e.g.* spatial systems Álvarez et al. (2011a).
  - 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

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 EU Networks of excellence.

References I

M. A. 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. A. 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, and N. D. Lawrence. Linear latent force models using Gaussian processes. Technical report, University of Sheffield, [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. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 55–63, Cambridge, MA, 2011b. MIT Press.

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

G. Della Gatta, M. Bansal, A. Ambesi-Impiombato, D. Antonini, C. Missero, and D. di Bernardo. Direct targets of the trp63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Research*, 18(6):939–948, Jun 2008. [URL]. [DOI].

P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: Applications to inferring transcription factor activities. *Bioinformatics*, 24:i70–i75, 2008. [PDF]. [DOI].

D. S. Goodsell. The molecular perspective: p53 tumor suppressor. *The Oncologist, Vol. 4, No. 2, 138-139, April 1999*, 4(2):138–139, 1999.

A. Honkela, C. Girardot, E. H. Gustafson, Y.-H. Liu, E. E. M. Furlong, N. D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proc. Natl. Acad. Sci. USA*, 107(17):7793–7798, Apr 2010. [DOI].

References II

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.

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].