Model Based Target Identification from Expression Data

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

University of Sheffield, U.K.
Talk at Cancer Research UK, Cambridge

6th February 2012

Outline

Outline

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Coregulation of Gene Expression

The Case for Computational Biology

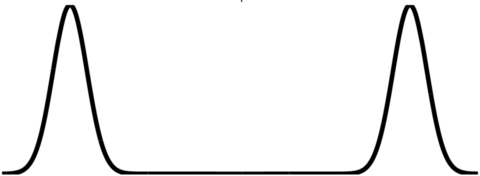
- Gene Expression to Transcriptional Regulation.
- A "data exploration" problem (computational biology/bioinformatics):
    - Use gene expression data to speculate on coregulated genes.
    - Traditionally use clustering of gene expression profiles.
- Contrast with (computational) systems biology approach: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

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling

mechanistic-modeling



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling

let the data “speak”

mechanistic-modeling

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

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

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling	*mechanistic-modeling*
let the data “speak”	impose physical laws
computational models	systems models

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling

let the data “speak”

computational models

adaptive models

mechanistic-modeling

impose physical laws

systems models

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

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

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

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

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

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

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

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

Weakly Mechanistic

Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling

let the data "speak"

computational models

adaptive models

PCA, clustering

mechanistic-modeling

impose physical laws

systems models

differential equations

SDE, ODE models

A Hybrid Approach

Introduce aspects of systems biology to computational models

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

Radiation Damage in the Cell

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

p53 “Guardian of the Cell”

- Responsible for Repairing DNA damage
- Activates DNA Repair proteins
- Pauses the Cell Cycle (prevents replication of damage DNA)
- Initiates *apoptosis* (cell death) in the case where damage can’t be repaired.
- Large scale feeback loop with NF-κB.

p53 DNA Damage Repair

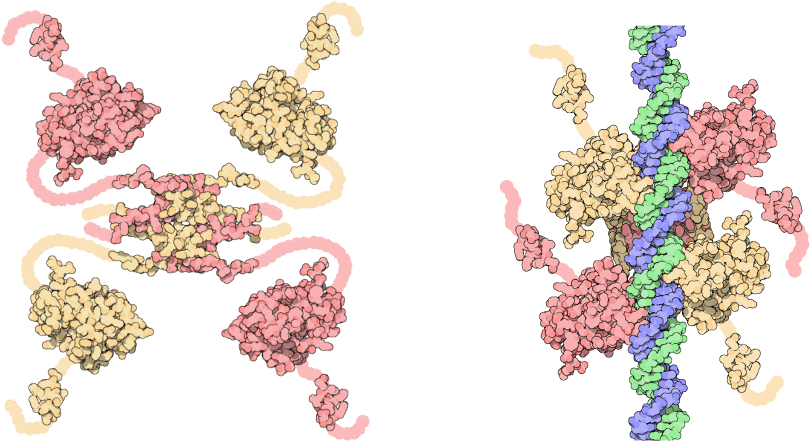


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the“Molecule of the Month” feature).

p53

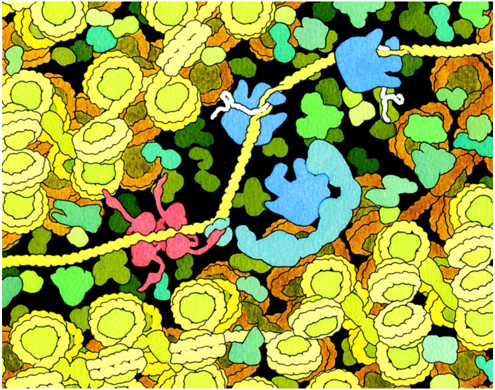


Figure: Repair of DNA damage by p53. Image from Goodsell (1999).

Some p53 Targets

DDB2	DNA Damage Specific DNA Binding Protein 2. (also governed by C/ EBP-beta, E2F1, E2F3,...).
p21	Cycline-dependent kinase inhibitor 1A (CDKN1A). A regulator of cell cycle progression. (also governed by SREBP-1a, Sp1, Sp3,...).
hPA26/SESN1	sestrin 1 Cell Cycle arrest.
BIK	BCL2-interacting killer. Induces cell death (apoptosis)
TNFRSF10b	tumor necrosis factor receptor superfamily, member 10b. A transducer of apoptosis signals.

Modelling Assumption

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

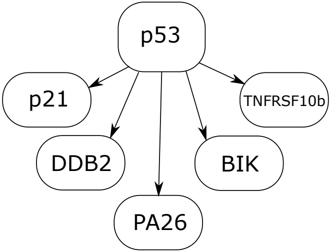


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

Standard Approach

Clustering of Gene Expression Profiles

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

Mathematical Model

Method

Open Access

Ranked prediction of p53 targets using hidden variable dynamic modeling

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

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

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

Published: 31 March 2006

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

Received: 24 November 2005
Revised: 30 January 2006
Accepted: 21 February 2006

Mathematical Model

- Differential equation model of system.

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

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

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

Mathematical Model

- Differential equation model of system.

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

$$d_j m_j(t) + \frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j p(t)$$

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

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

Mathematical Model

- Differential equation model of system.

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

$$d_j m_j(t) + \frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j p(t)$$

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

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

Mathematical Model

- Differential equation model of system.

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

$$d_j m_j(t) + \frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j p(t)$$

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

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

Mathematical Model

- Differential equation model of system.

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

$$d_j m_j(t) + \frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j p(t)$$

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

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

Mathematical Model

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

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

  rate of mRNA transcription, baseline transcription rate, transcription factor activity, mRNA decay
- We have observations of $m_j(t)$ from gene expression.

Mathematical Model

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

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

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

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

Mathematical Model

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

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

$$d_j m_j\left(t\right) \approx b_j + s_j p\left(t\right)$$

  rate of mRNA transcription, baseline transcription rate, transcription factor activity, mRNA decay
- We have observations of $m_j\left(t\right)$ from gene expression.
- Reorder differential equation and ignore gradient term.
- This suggests genes are scaled and offset versions of the TF.

Mathematical Model

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

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

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

Response of p53

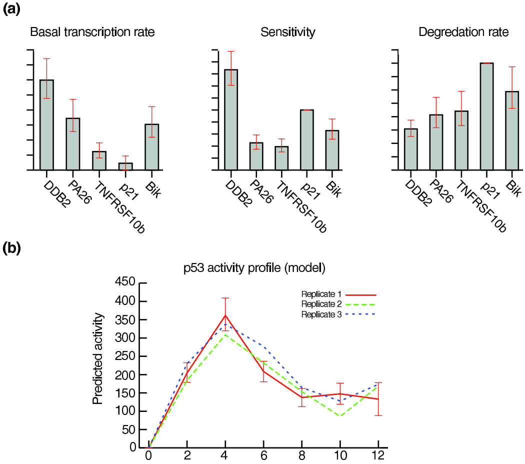
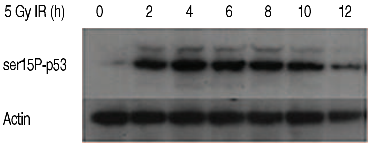


Figure: Results from Barenco et al. (2006). Top is parameter estimates. Bottom is inferred profile.

Response to p53 ...



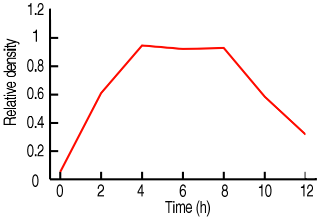


Figure: Results from Barenco et al. (2006). Activity profile of p53 was measured by Western blot to determine the levels of ser-15 phosphorylated p53 (ser15P-p53).

Outline

Motivation

Probabilistic Model for $p(t)$

Cascade Differential Equations

Multiple Transcription Factors

Discussion and Future Work

Probabilistic Model for $p(t)$

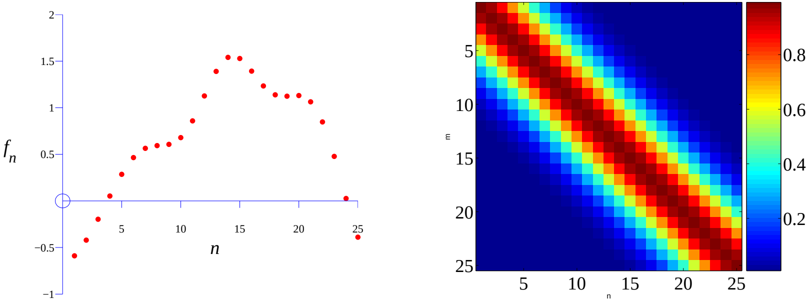
- We introduce a probabilistic model for $p(t)$.
- It is known as a Gaussian process, but we can think of it as a multivariate Gaussian (also known as a multivariate normal) distribution.
- The distribution has a mean vector, $\mathbf{m}$ and a covariance matrix, $\mathbf{K}$.
- We will consider the mean to be zero: $\mathbf{m} = 0$.
- The covariance matrix will be structured to give correlation between samples.
- We will sample 25 points from the Gaussian distribution.
- Samples are governed by a 25×25 correlation matrix.

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 p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_m and p_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- In practice covariance matrix is computed as a function of time—index is equivalent to time.
- Different covariance functions give different characteristics.
- Because the models are *probabilistic* we can sample different characteristics.

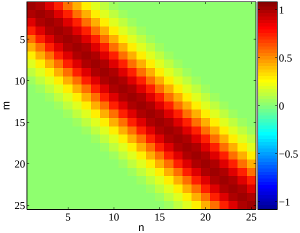
Covariance Functions

Where did this covariance matrix come from?

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

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

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



Covariance Samples

`demCovFuncSample`

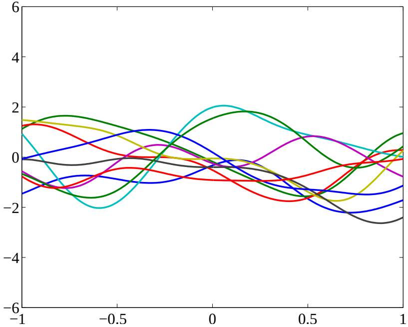


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

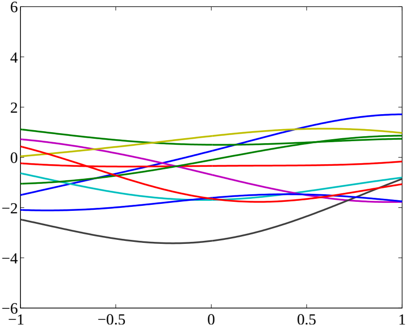


Figure: Exponentiated quadratic kernel with $\ell = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

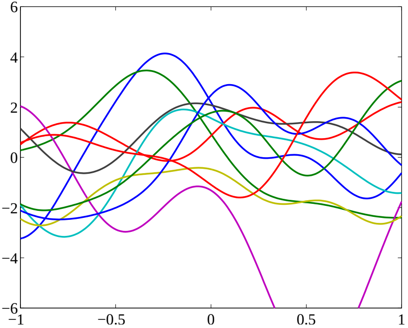


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

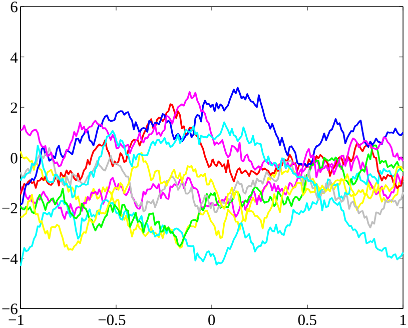


Figure: Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function $\ell = 1$, $\alpha = 4$

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

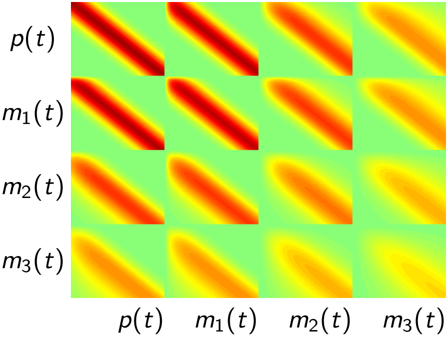
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



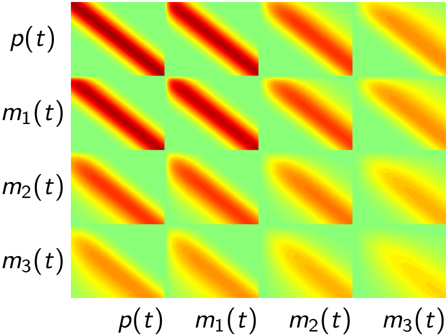
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



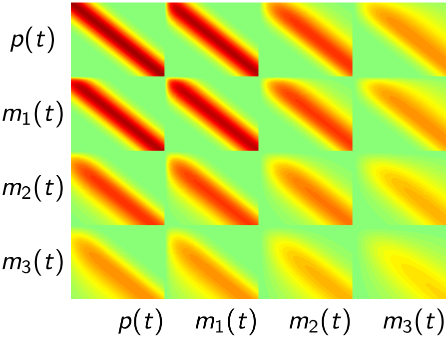
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



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

- `simSample`

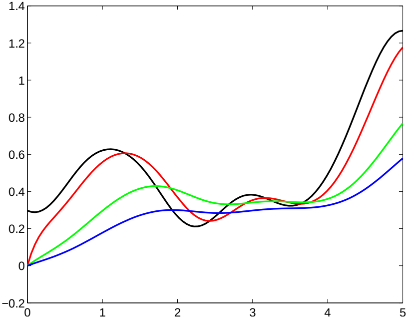


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

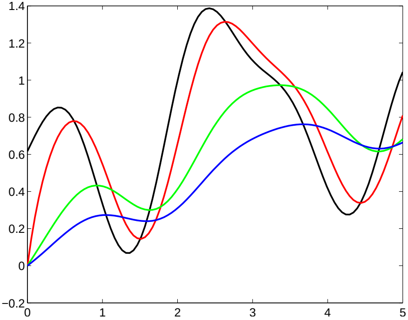


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

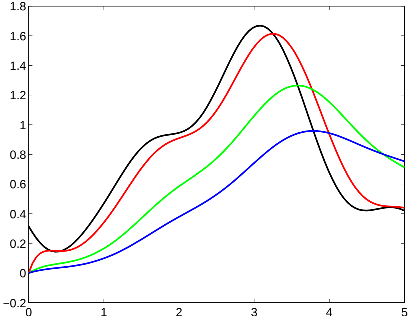


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

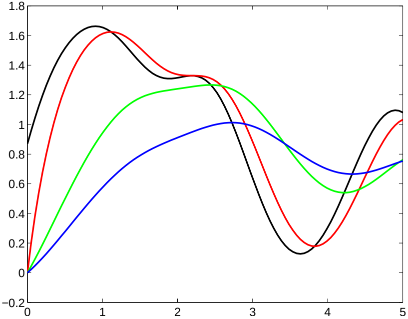
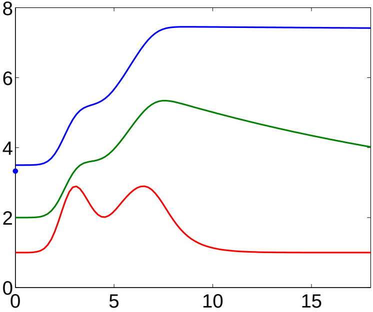


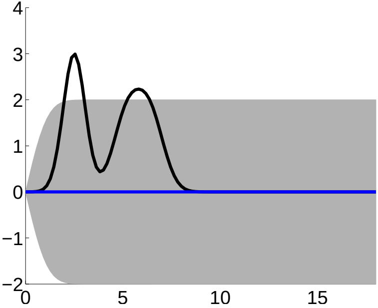
Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



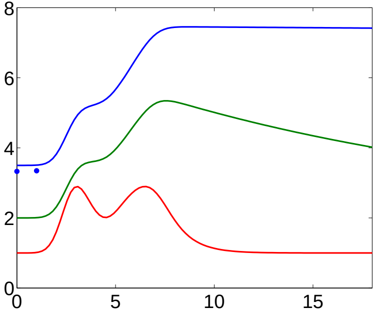
True "gene profiles" and noisy observations.



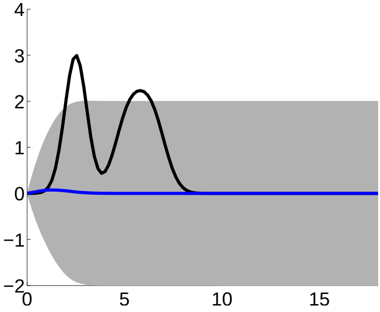
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



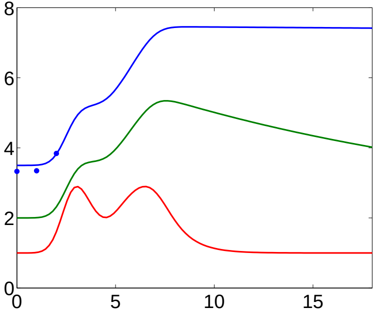
True "gene profiles" and noisy observations.



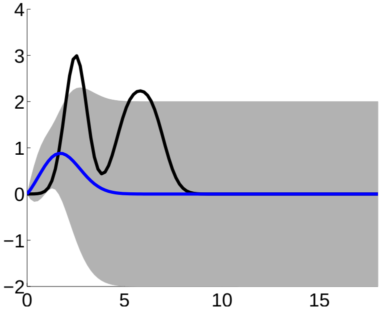
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



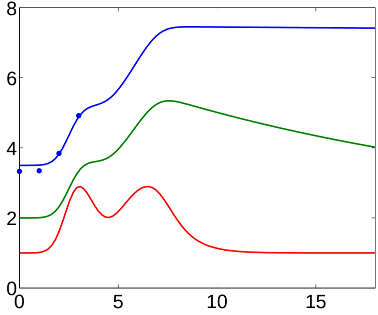
True "gene profiles" and noisy observations.



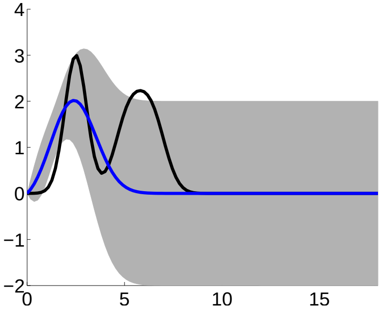
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



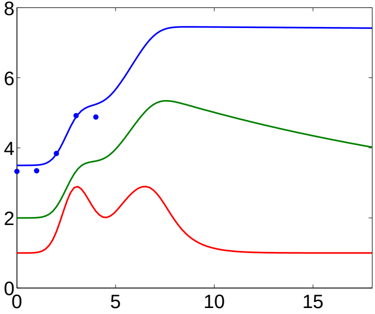
True "gene profiles" and noisy observations.



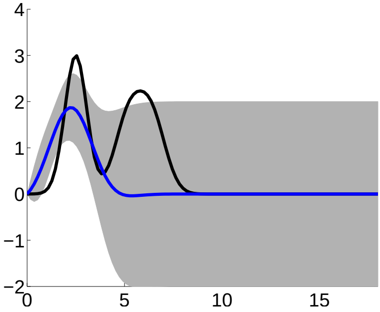
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



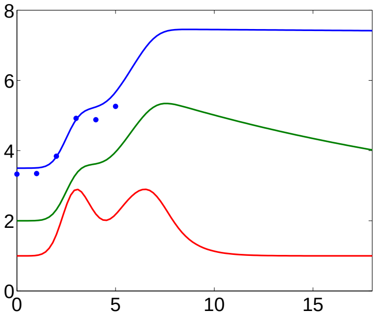
True "gene profiles" and noisy observations.



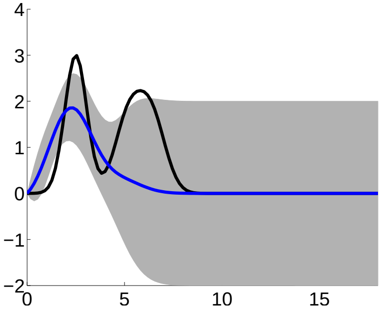
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



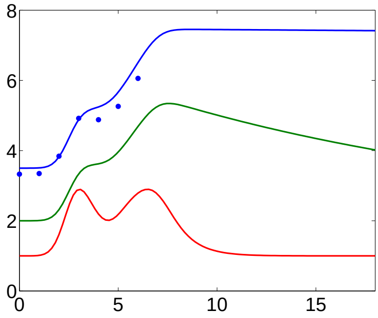
True "gene profiles" and noisy observations.



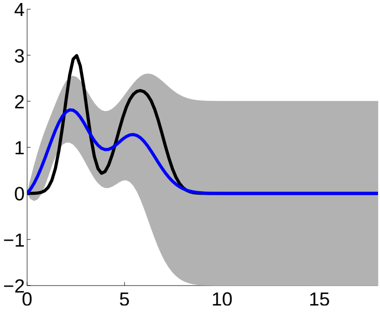
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



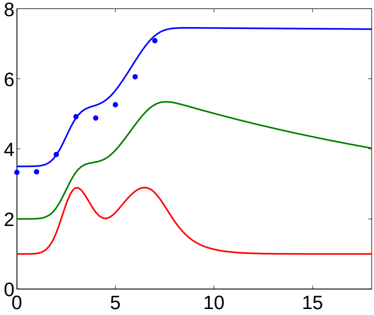
True “gene profiles” and noisy observations.



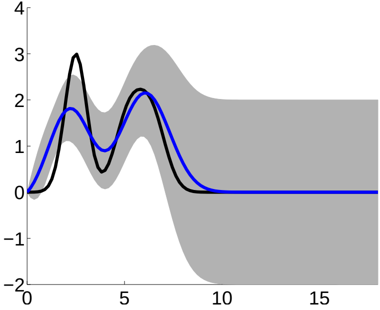
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



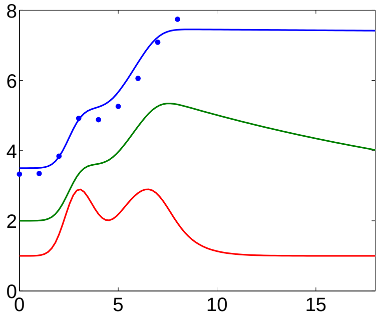
True “gene profiles” and noisy observations.



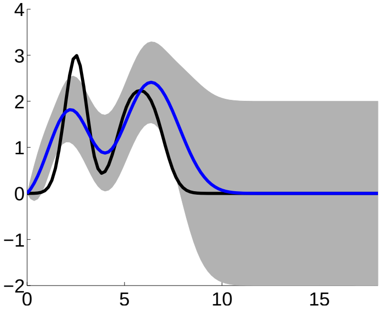
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



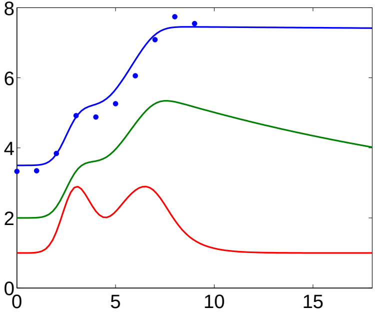
True “gene profiles” and noisy observations.



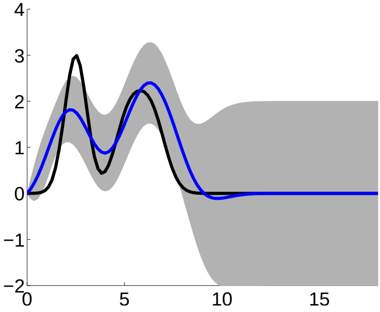
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



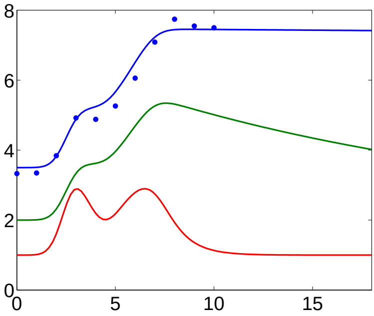
True “gene profiles” and noisy observations.



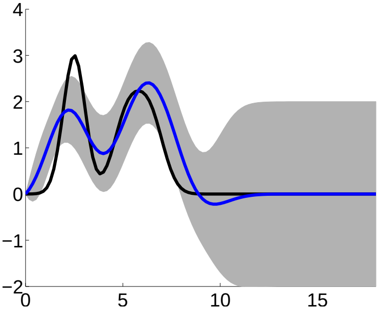
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



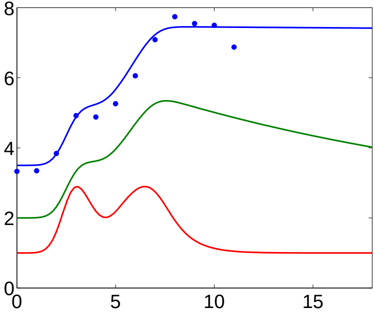
True "gene profiles" and noisy observations.



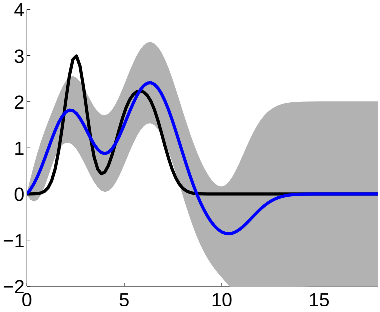
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



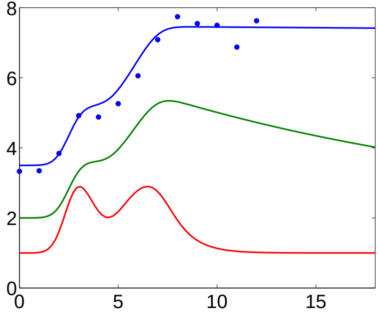
True "gene profiles" and noisy observations.



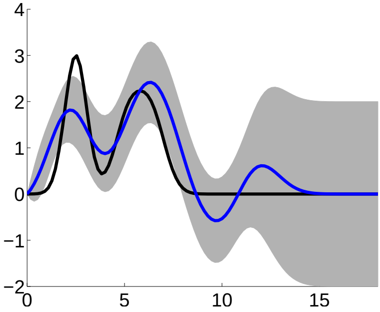
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



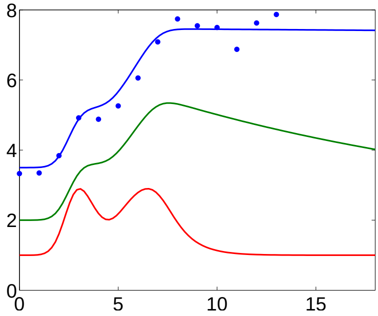
True "gene profiles" and noisy observations.



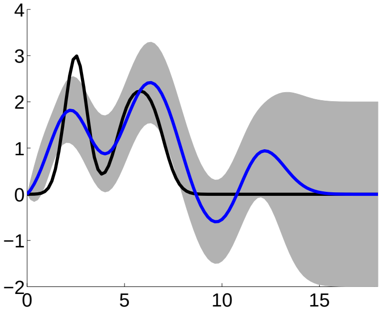
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



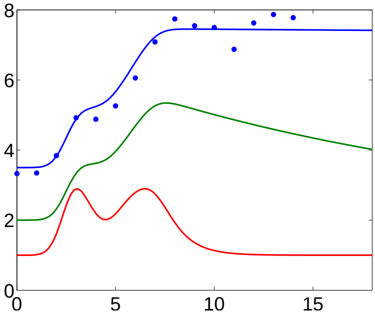
True "gene profiles" and noisy observations.



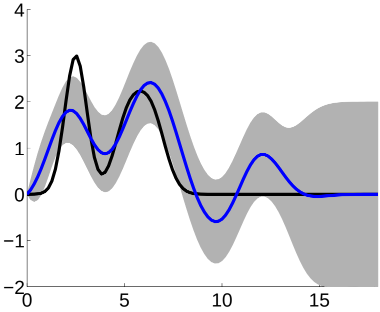
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



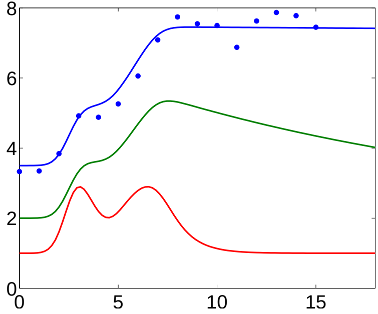
True “gene profiles” and noisy observations.



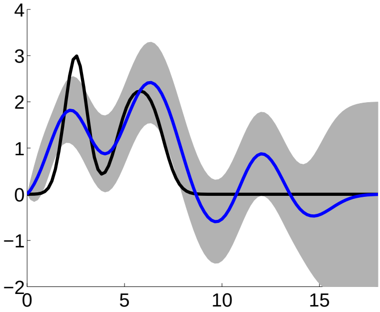
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



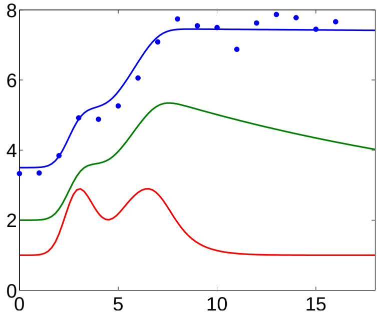
True "gene profiles" and noisy observations.



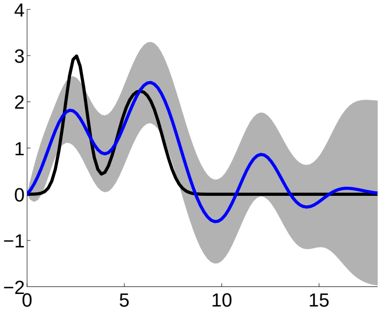
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



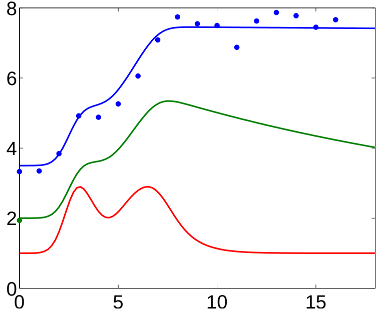
True "gene profiles" and noisy observations.



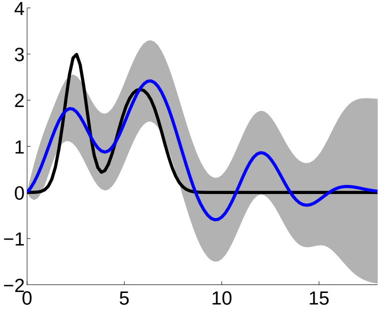
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



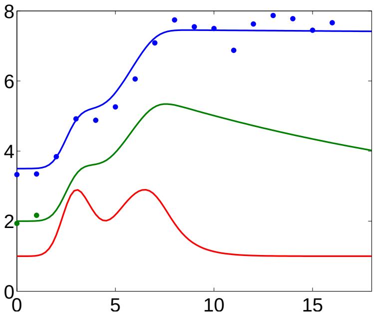
True "gene profiles" and noisy observations.



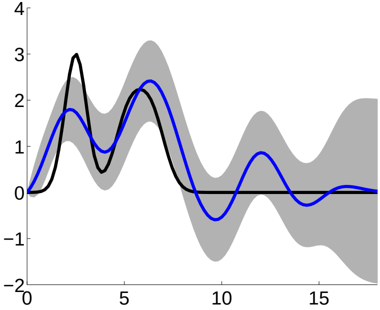
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



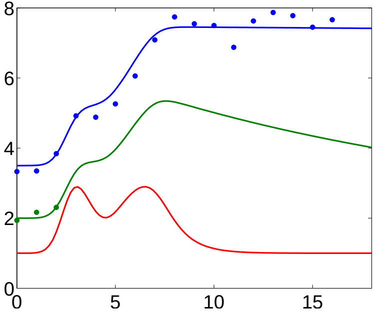
True "gene profiles" and noisy observations.



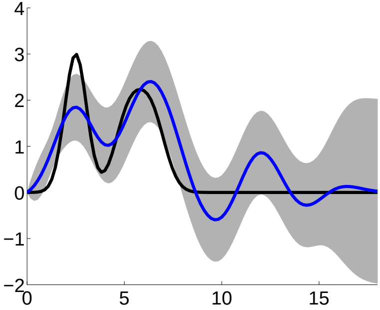
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



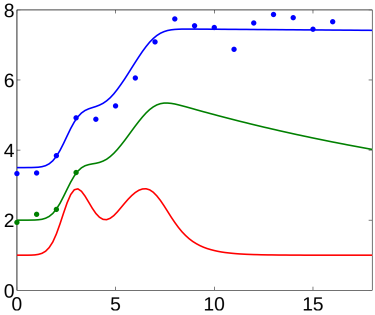
True "gene profiles" and noisy observations.



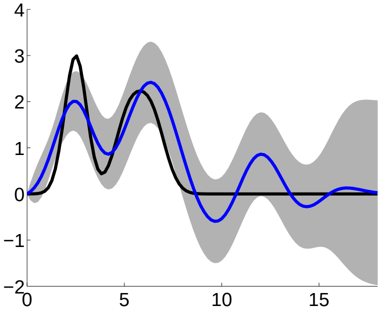
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



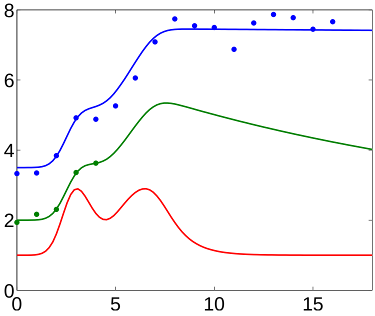
True "gene profiles" and noisy observations.



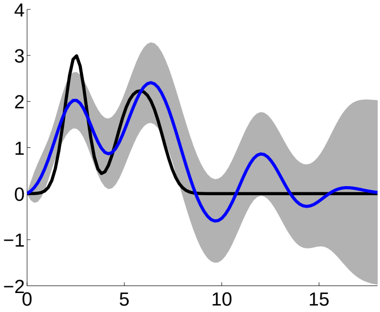
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



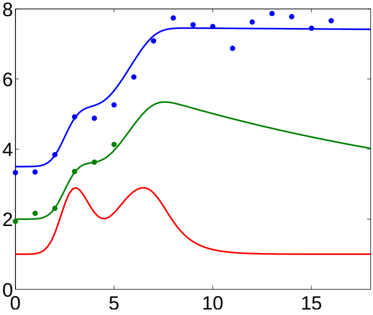
True "gene profiles" and noisy observations.



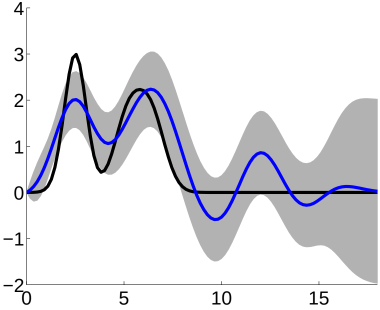
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



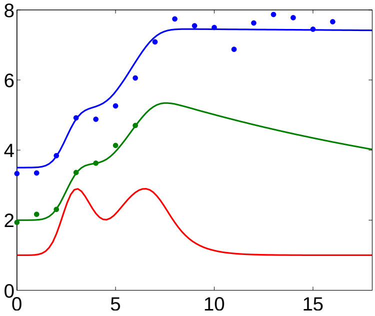
True "gene profiles" and noisy observations.



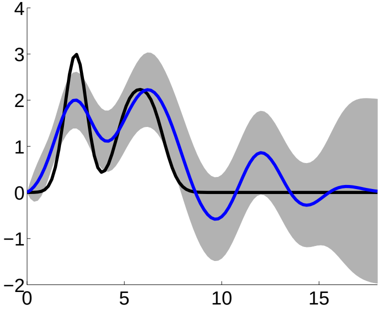
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



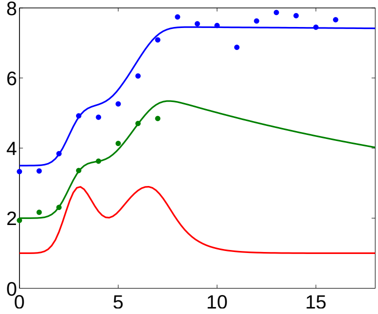
True "gene profiles" and noisy observations.



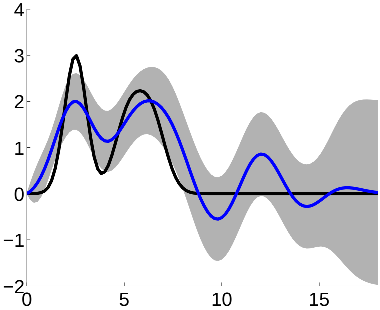
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



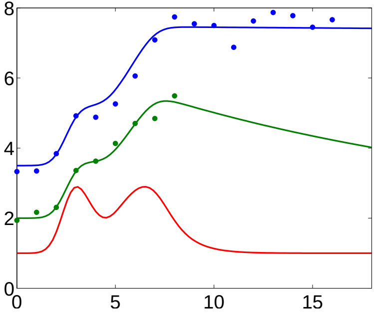
True "gene profiles" and noisy observations.



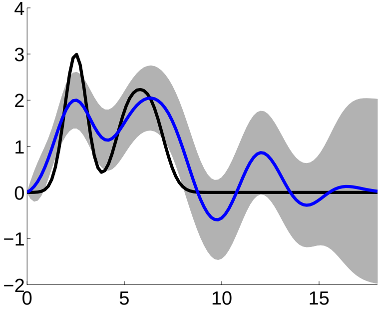
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



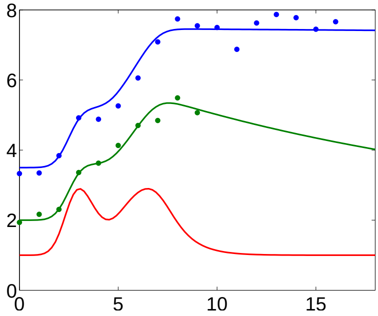
True "gene profiles" and noisy observations.



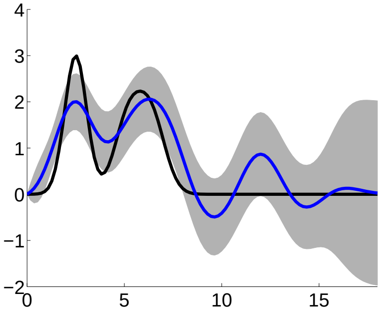
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



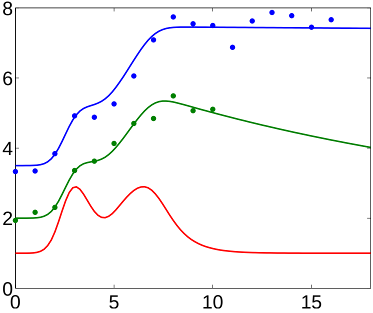
True “gene profiles” and noisy observations.



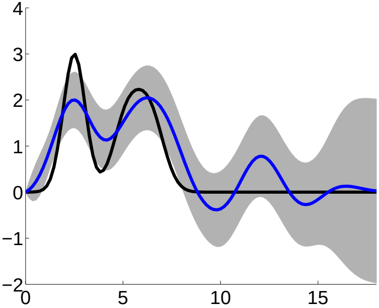
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



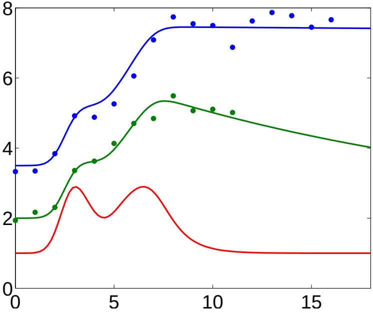
True "gene profiles" and noisy observations.



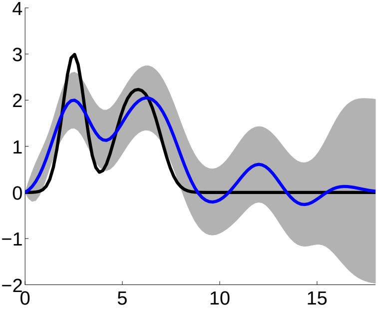
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



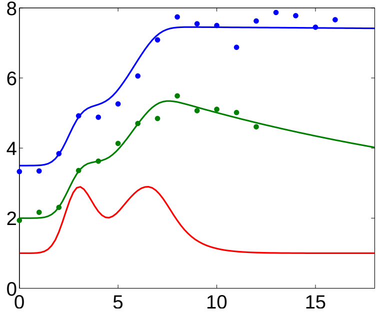
True "gene profiles" and noisy observations.



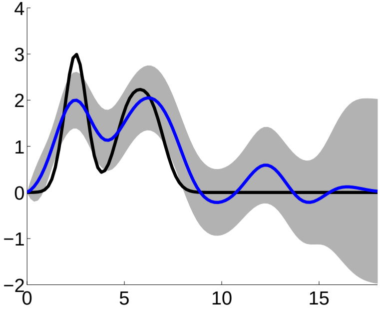
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



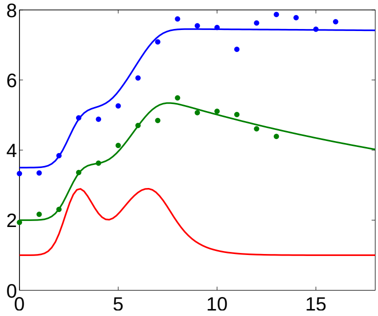
True "gene profiles" and noisy observations.



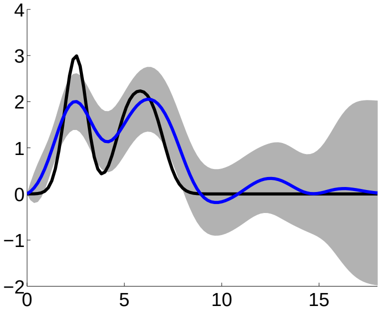
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



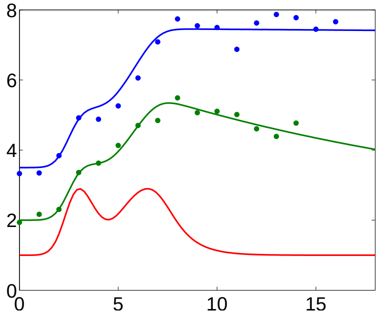
True "gene profiles" and noisy observations.



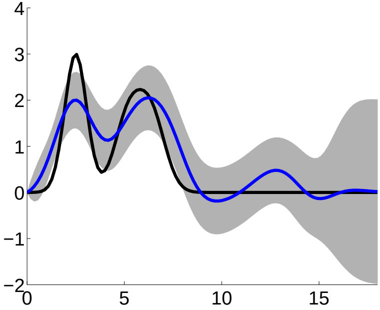
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



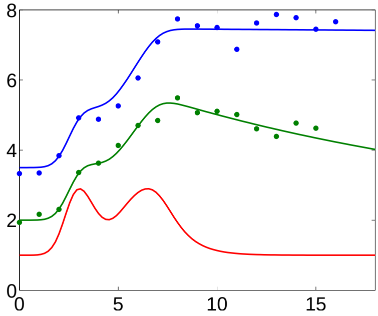
True “gene profiles” and noisy observations.



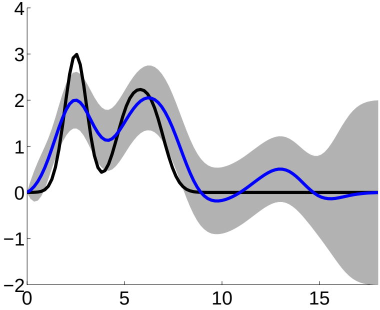
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



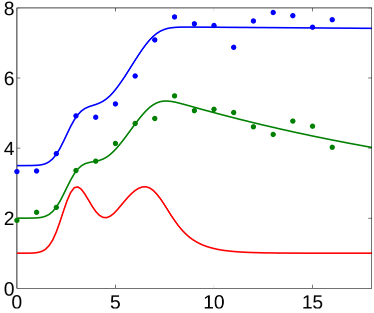
True "gene profiles" and noisy observations.



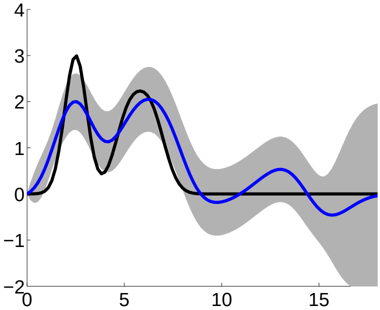
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



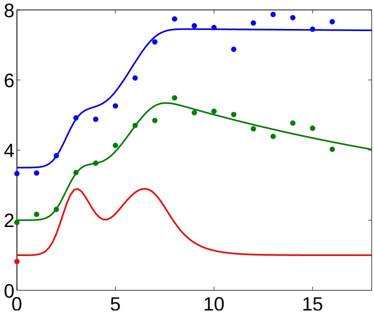
True "gene profiles" and noisy observations.



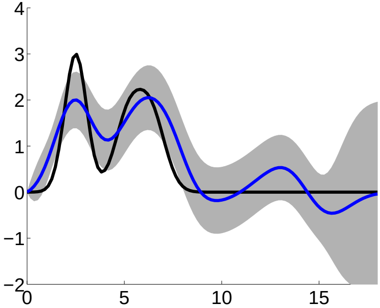
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



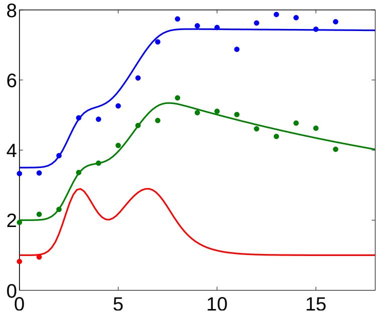
True "gene profiles" and noisy observations.



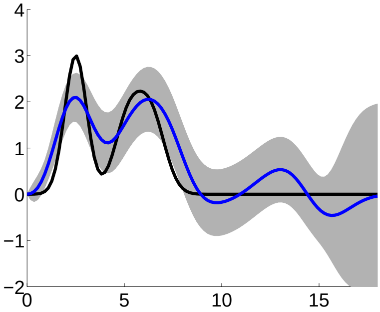
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



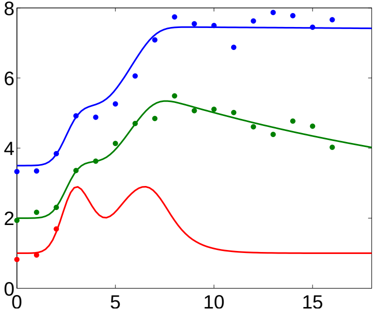
True “gene profiles” and noisy observations.



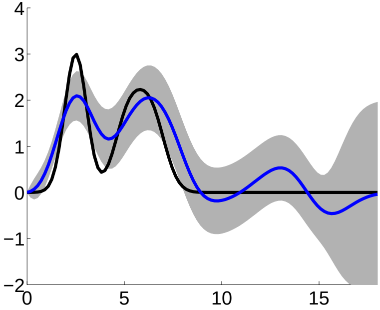
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



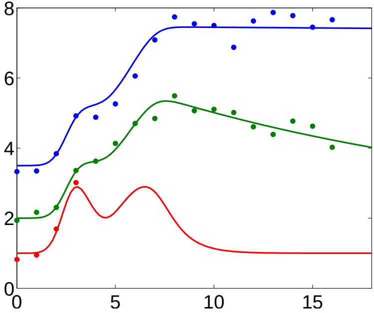
True "gene profiles" and noisy observations.



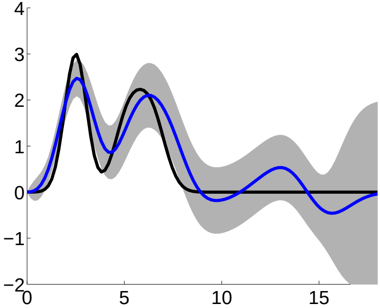
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



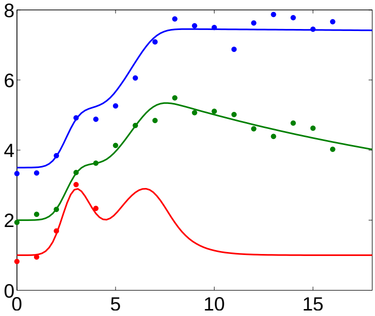
True “gene profiles” and noisy observations.



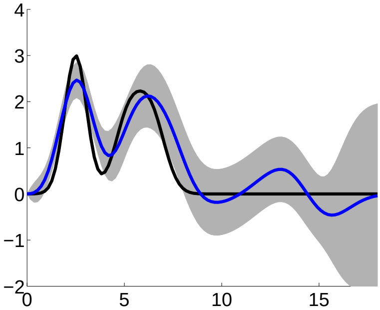
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



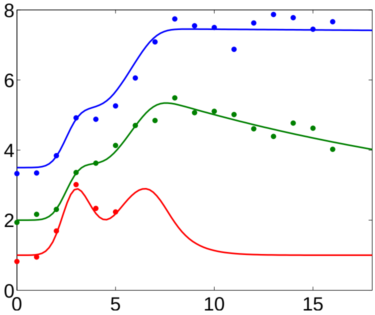
True "gene profiles" and noisy observations.



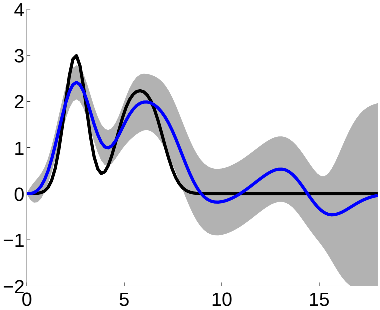
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



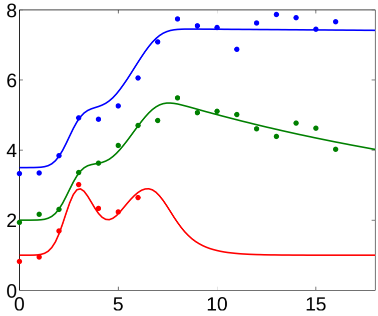
True "gene profiles" and noisy observations.



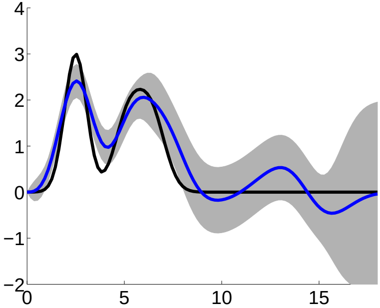
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



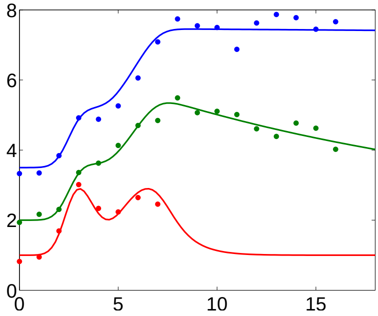
True “gene profiles” and noisy observations.



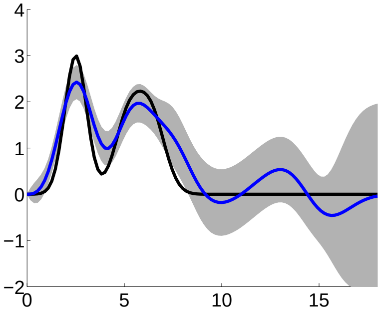
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



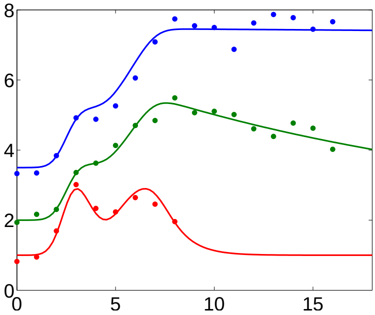
True "gene profiles" and noisy observations.



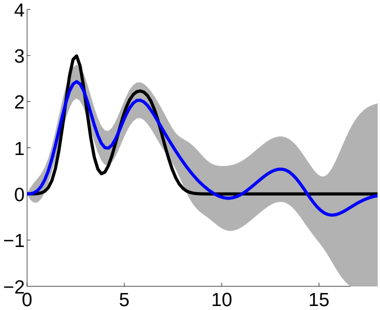
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



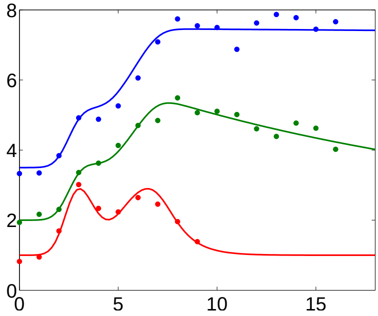
True "gene profiles" and noisy observations.



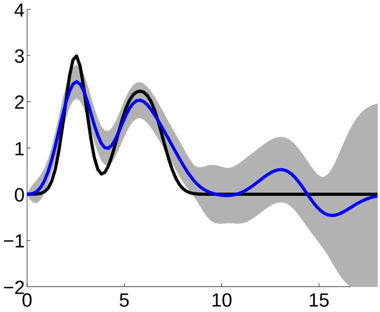
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



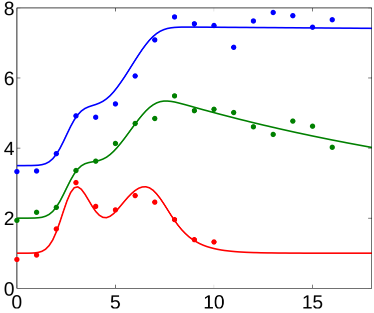
True “gene profiles” and noisy observations.



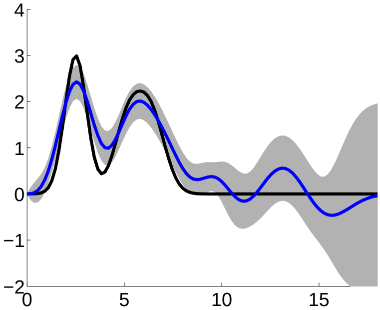
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



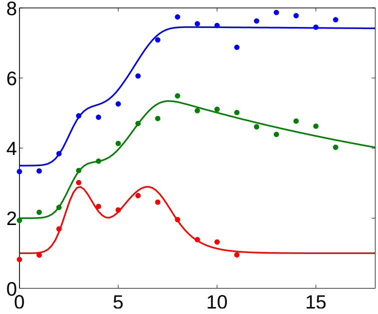
True "gene profiles" and noisy observations.



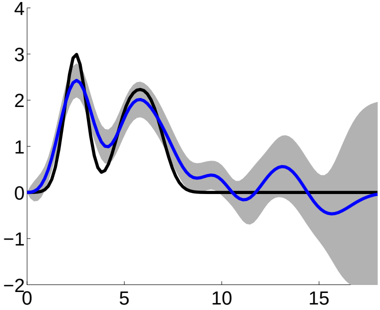
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



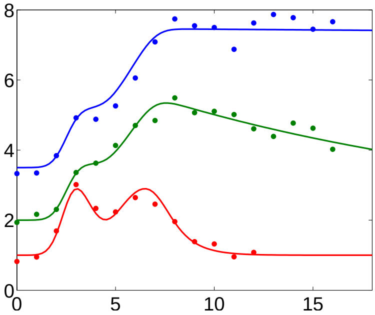
True “gene profiles” and noisy observations.



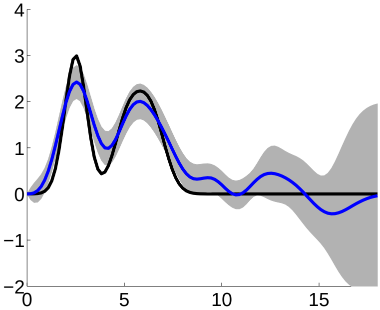
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



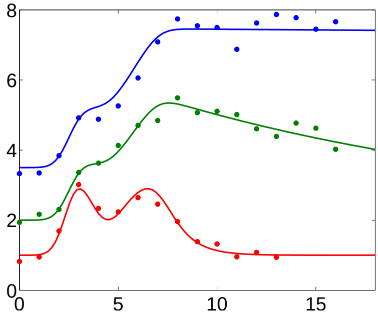
True "gene profiles" and noisy observations.



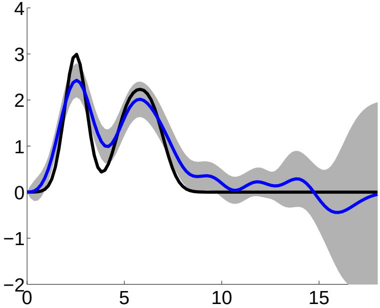
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



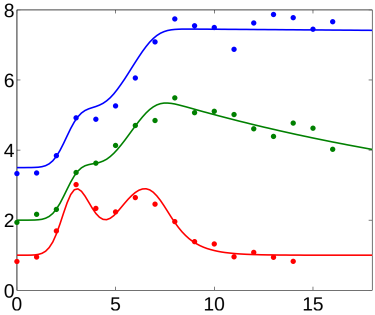
True "gene profiles" and noisy observations.



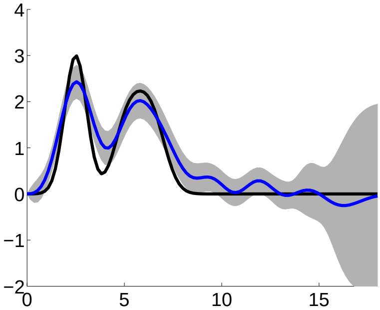
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



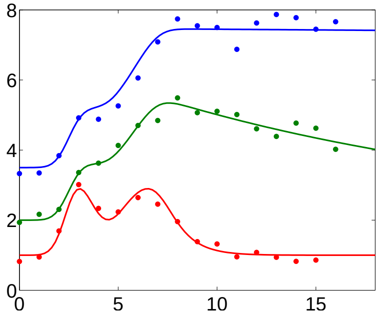
True "gene profiles" and noisy observations.



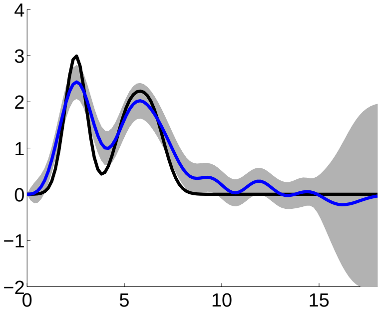
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



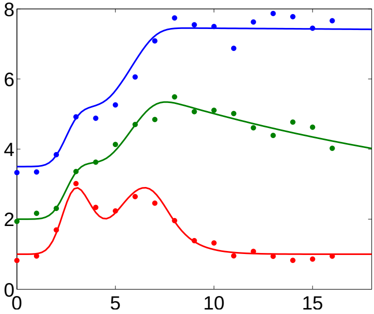
True “gene profiles” and noisy observations.



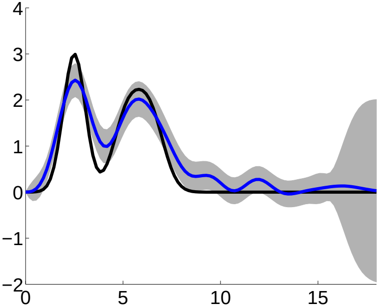
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.

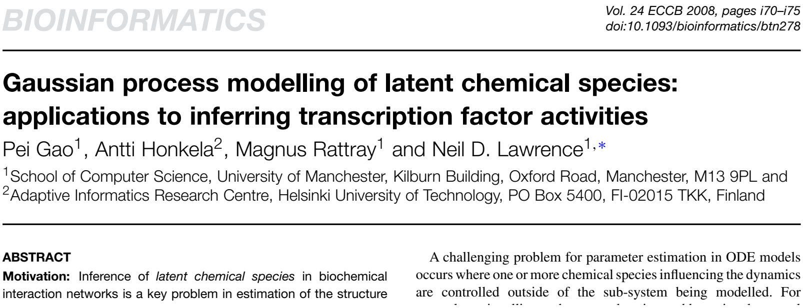


True "gene profiles" and noisy observations.



Inferred transcription factor activity.

p53 Results with GP



BIOINFORMATICS

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

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

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

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

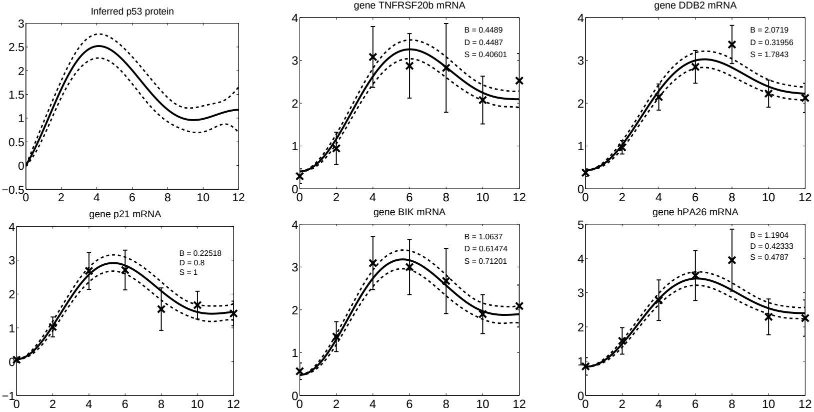
ABSTRACT

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

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

p53 Results with GP

(Gao et al., 2008)

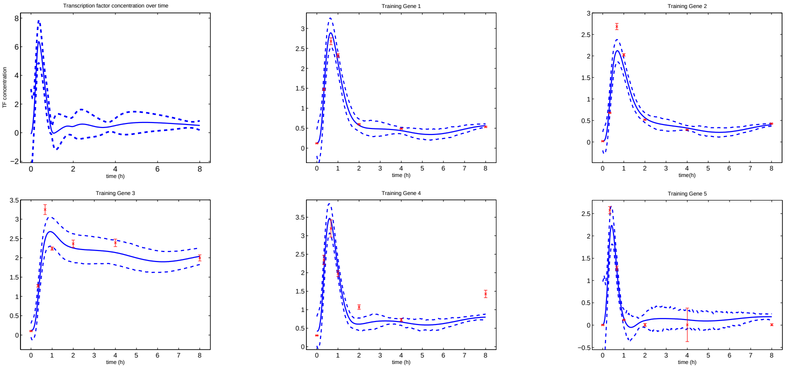


Ranking with ERK Signalling

- Target Ranking for Elk-1.
- Elk-1 is phosphorylated by ERK from the EGF signalling pathway.
- Predict concentration of Elk-1 from known targets.
- Rank other targets of Elk-1.

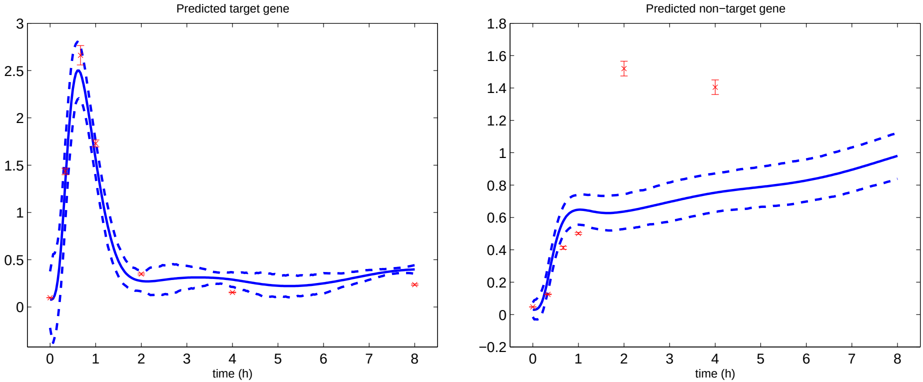
Elk-1 (MLP covariance)

Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



Outline

Motivation

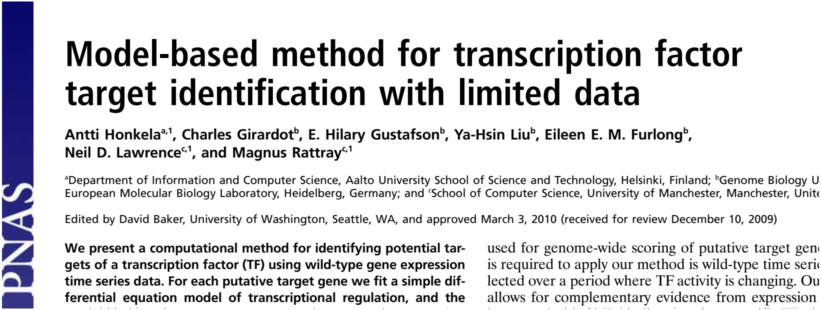
Probabilistic Model for $p(t)$

Cascade Differential Equations

Multiple Transcription Factors

Discussion and Future Work

Cascaded Differential Equations



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

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

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

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

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

used for genome-wide scoring of putative target gen
is required to apply our method is wild-type time seri
lected over a period where TF activity is changing. Ou
allows for complementary evidence from expression

Cascaded Differential Equations

(Honkela et al., 2010)

- Transcription factor protein also has governing mRNA.
- This mRNA can be measured.
- In signalling systems this measurement can be misleading because it is activated (phosphorylated) transcription factor that counts.
- In development phosphorylation plays less of a role.

Drosophila *Mesoderm* Development

Collaboration with Furlong Lab in EMBL Heidelberg.

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

Cascaded Differential Equations

(Honkela et al., 2010)

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

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

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

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

Covariance for Translation/Transcription Model

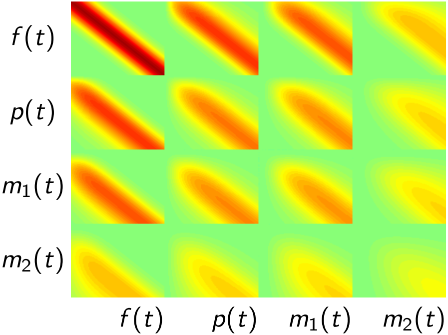
RBF covariance function for $f(t)$

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

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

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

δ	d_1	s_1	d_2	s_2
1	5	5	0.5	0.5



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

- `disimSample`

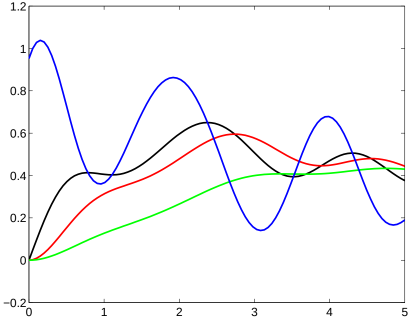


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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

- `disimSample`

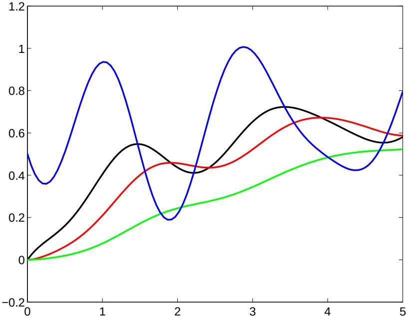


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

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

- `disimSample`

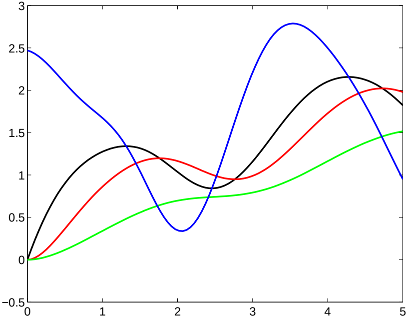


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

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

- `disimSample`

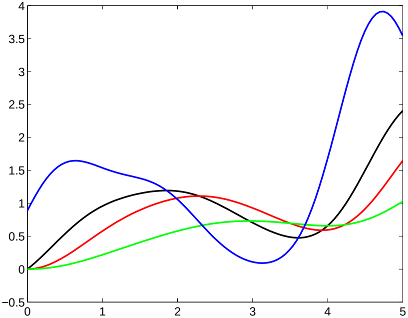


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

Twist Results

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

Results for Twi using the Cascade model

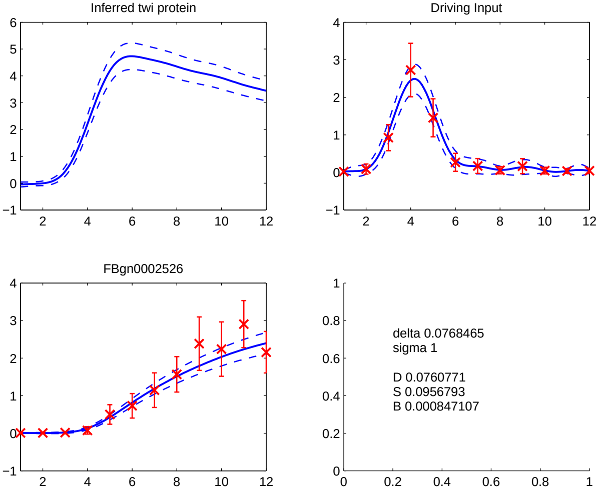


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

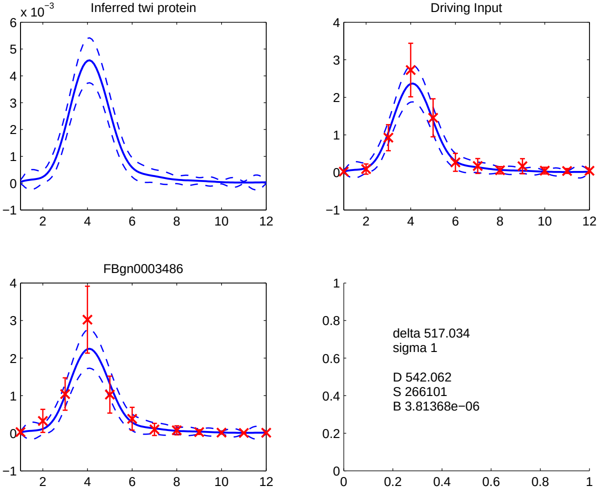
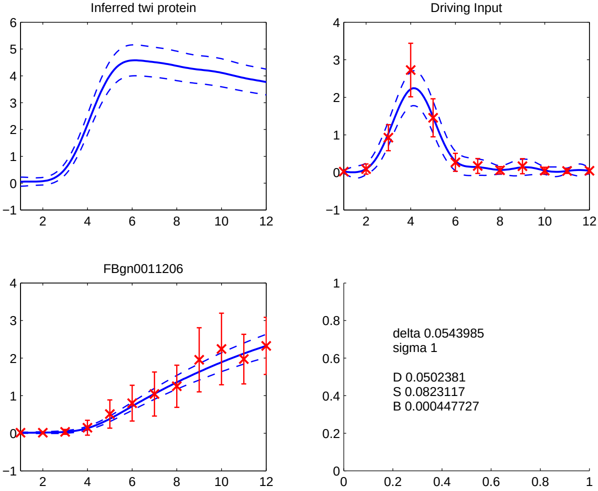


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model



delta 0.0543985
sigma 1

D 0.0502381
S 0.0823117
B 0.000447727

Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

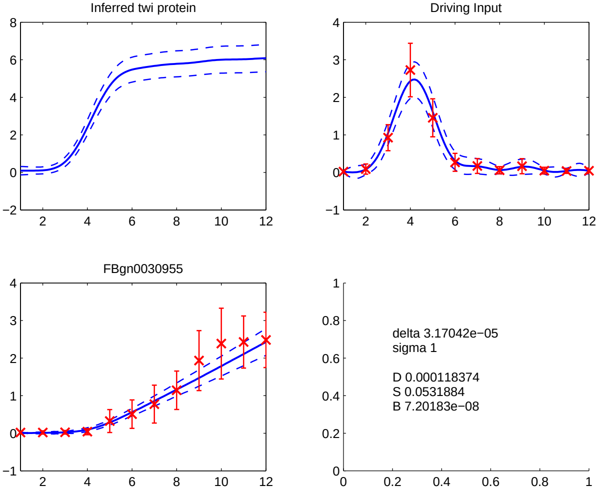


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

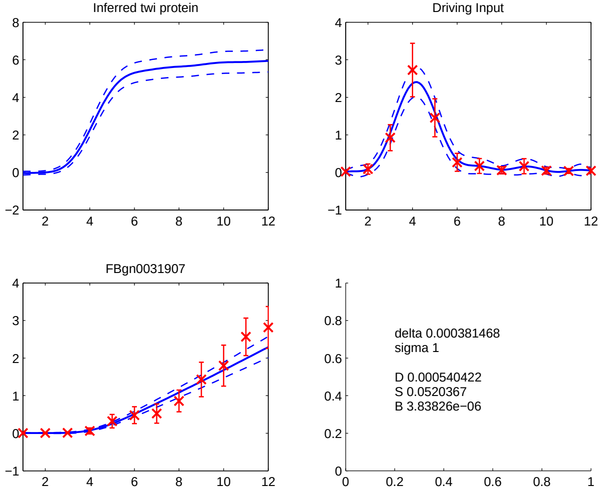


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

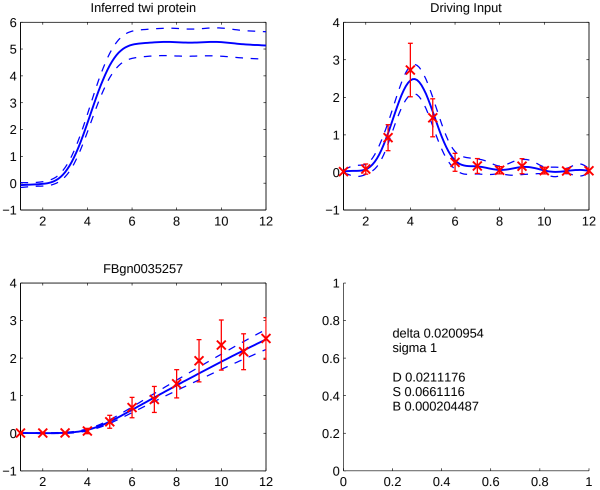


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

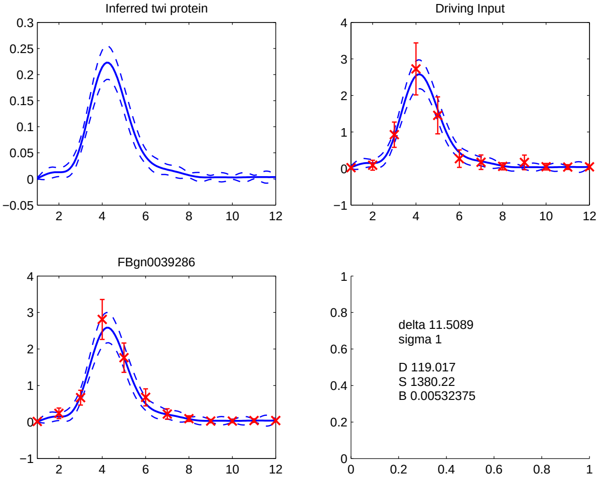
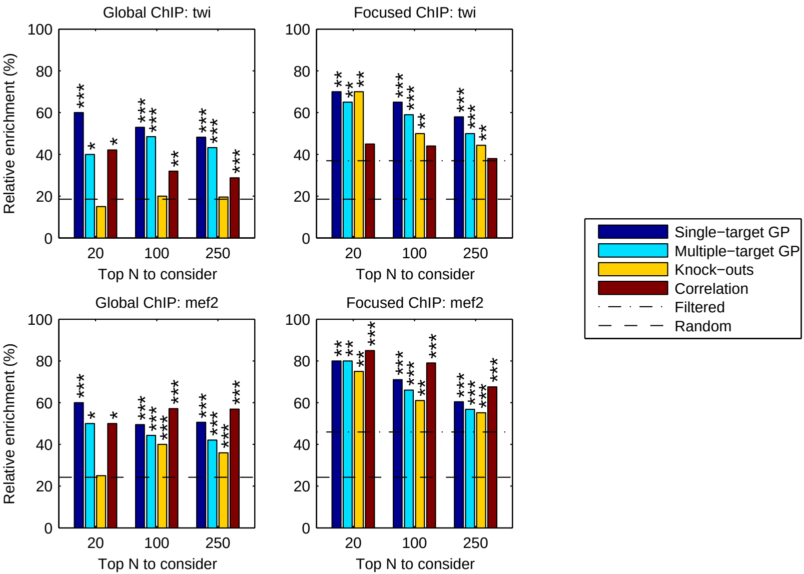


Figure: Model for flybase gene identity FBgn0039286.

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

Motivation

Probabilistic Model for $p(t)$

Cascade Differential Equations

Multiple Transcription Factors

Discussion and Future Work

A “middle-out” approach for inferring regulatory networks

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

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

A “middle-out” approach for inferring regulatory networks

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

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

A “middle-out” approach for inferring regulatory networks

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

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

A “middle-out” approach for inferring regulatory networks

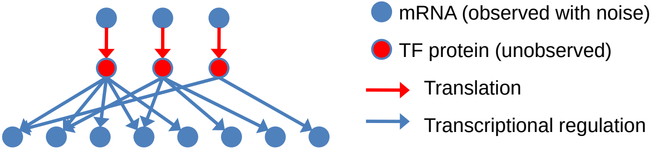
Task: find targets of a small number of co-regulating transcription factors (TFs) from time-series expression data:

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

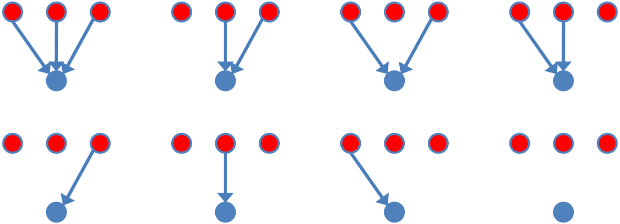
A “middle-out” approach for inferring regulatory networks

- Training stage: Parameter estimation on known network

(a): Training phase



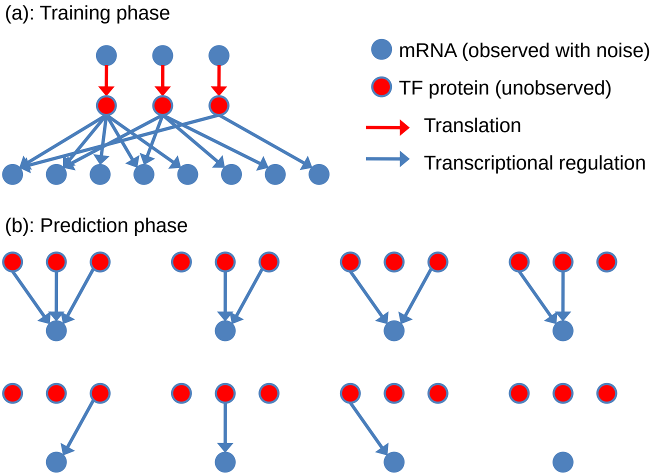
(b): Prediction phase



- Scanning stage: Bayesian evidence model scoring for target inference

A “middle-out” approach for inferring regulatory networks

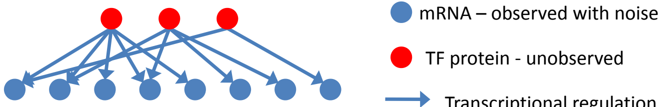
- Training stage: Parameter estimation on known network



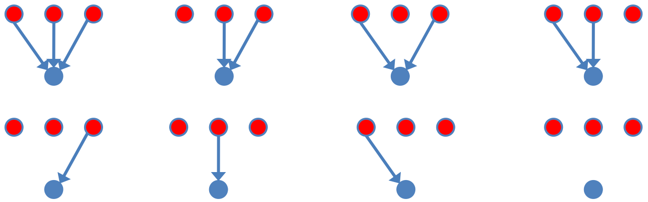
- Scanning stage: Bayesian evidence model scoring for target inference

A “middle-out” approach for inferring regulatory networks

- Training stage with post-translational modification



- Scanning stage: Bayesian evidence model scoring for target inference



Model of transcriptional regulation

- Transcription

$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = F\left(p_1(t), \ldots, p_K(t); \boldsymbol{\theta}_j\right) - d_j m_j(t)$$

$m_j(t)$ – target gene j mRNA concentration function
$p_i(t)$ – transcription factor i protein concentration function
$F(\mathbf{p}; \boldsymbol{\theta}_j)$ – regulation model, d_j – mRNA decay rate

- Translation (optional)

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = f_i(t) - \delta_i p_i(t)$$

$f_i(t)$ – transcription factor i mRNA concentration function
δ_i – protein decay rate

Model of transcriptional regulation

- Transcription

$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = F\left(p_1(t), \dots, p_K(t); \boldsymbol{\theta}_j\right) - d_j m_j(t)$$

$m_j(t)$ – target gene j mRNA concentration function
$p_i(t)$ – transcription factor i protein concentration function
$F(\mathbf{p}; \boldsymbol{\theta}_j)$ – regulation model, d_j – mRNA decay rate

- Translation (optional)

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = f_i(t) - \delta_i p_i(t)$$

$f_i(t)$ – transcription factor i mRNA concentration function
δ_i – protein decay rate

Gaussian process inference over latent functions

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

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

Gaussian process inference over latent functions

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

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

Gaussian process inference over latent functions

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

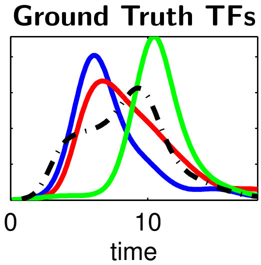
Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010)

Gaussian process inference over latent functions

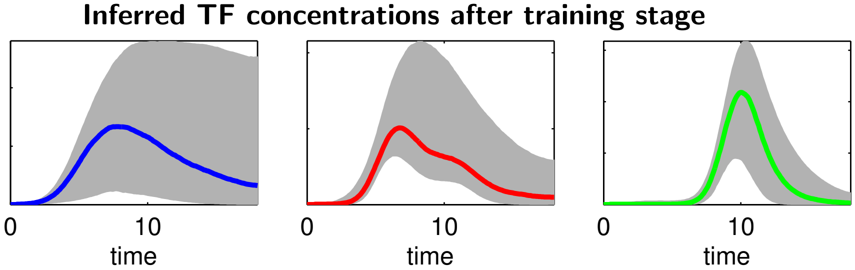
- Transcription factors considered **inputs** to the system
- Modelled as samples from a Gaussian process prior distribution
- Equations linear in $\mathbf{m}(t)$ can be solved as a function of $\mathbf{p}(t)$ so no need for numerical ODE solver to compute likelihood
- Useful way to close an open system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

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

Artificial data: one experimental condition

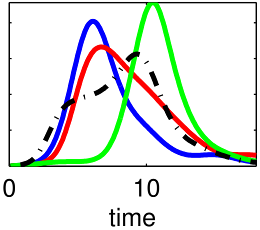


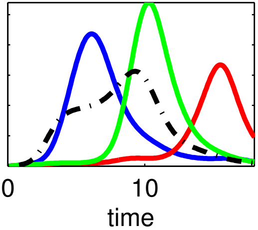
Ground Truth TFs



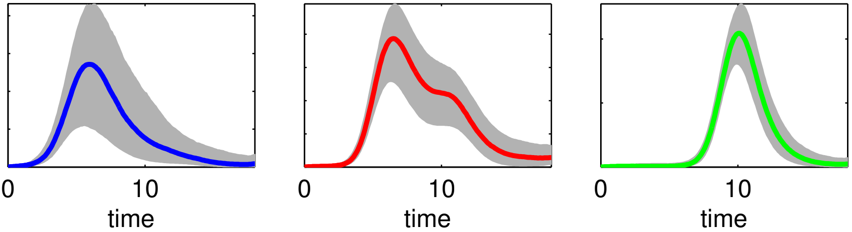
Inferred TF concentrations after training stage

Artificial data: two experimental conditions

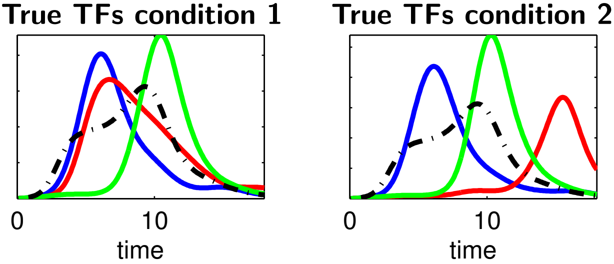

True TFs condition 1


True TFs condition 2

Inferred TF concentrations for condition 1

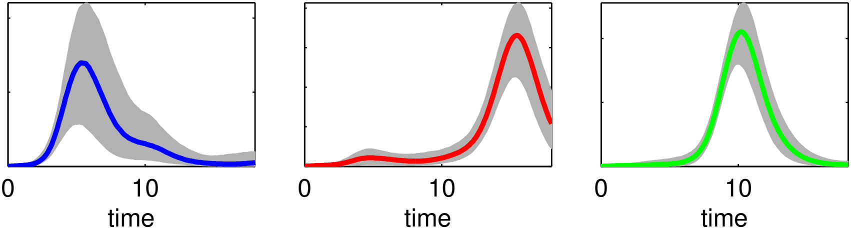


Artificial data: two experimental conditions



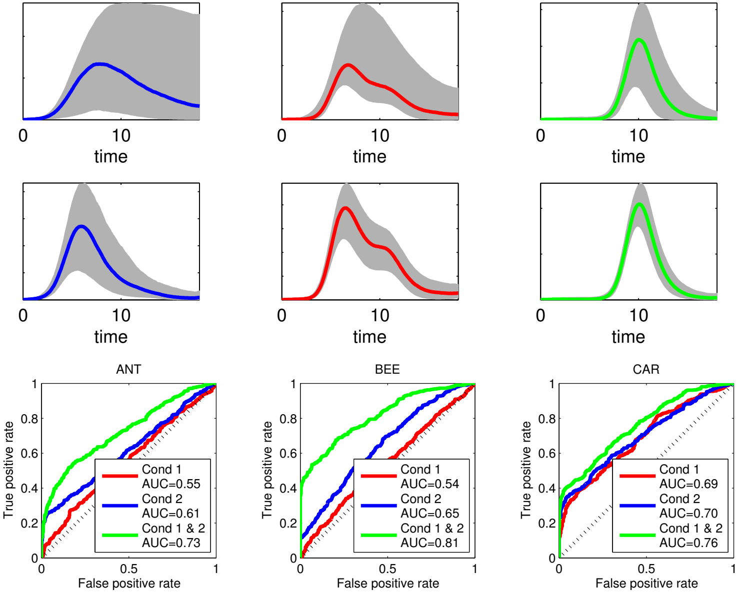
True TFs condition 1

True TFs condition 2

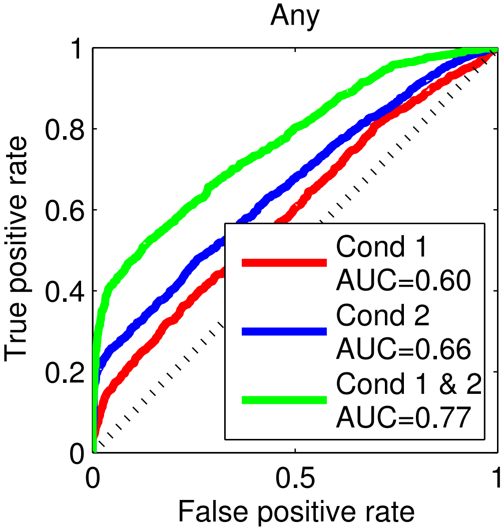


Inferred TF concentrations for condition 2

Artificial data: scanning performance for each TF

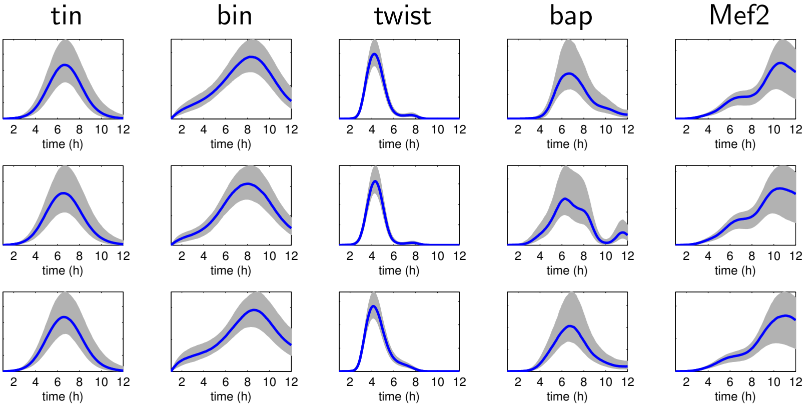


Artificial data: scanning performance for all TFs



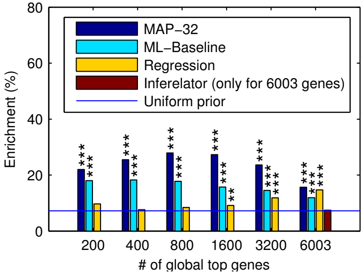
Drosophila training

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

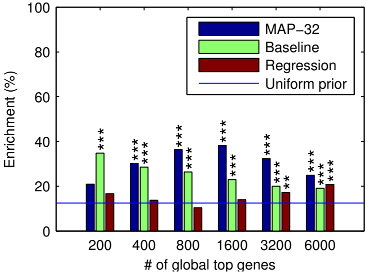


Drosophila scanning: model ranking

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



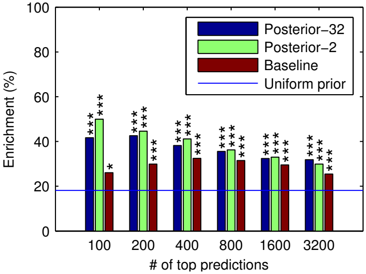
All “non-quiet” genes



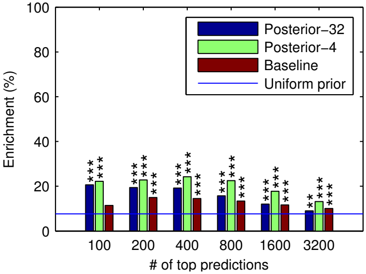
All targets with in situ evidence

Drosophila scanning: link ranking

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



TF regulation



TF pair regulation

Summary and Conclusion

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

M. K. Titsias, A. Honkela, M. Rattray and N. D. Lawrence "Regulatory network inference using probabilistic gene regulation models" manuscript in preparation (2010).

Summary and Conclusion

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

M. K. Titsias, A. Honkela, M. Rattray and N. D. Lawrence "Regulatory network inference using probabilistic gene regulation models" manuscript in preparation (2010).

Summary and Conclusion

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

M. K. Titsias, A. Honkela, M. Rattray and N. D. Lawrence "Regulatory network inference using probabilistic gene regulation models" manuscript in preparation (2010).

Outline

Motivation

Probabilistic Model for $p(t)$

Cascade Differential Equations

Multiple Transcription Factors

Discussion and Future Work

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- Software available through bioconductor (TIGRE Package) `http://bioconductor.org/packages/2.6/bioc/html/tigre.html`.
- Applications in modeling gene expression.
- Cascade model introduces model of translation.
- Ongoing/other work:
    - Non linear response and non linear differential equations.
    - Improving computational complexity.
    - Stochastic differential equations.

Acknowledgements

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

Funded by the BBSRC award "Improved Processing of microarray data using probabilistic models" and EPSRC award "Gaussian Processes for Systems Identification with applications in Systems Biology" Academy of Finland and FP7 PASCALII NoE

References I

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

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

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

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

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

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

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

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

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

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

M. K. Titsias, N. D. Lawrence, and M. Rattray. Efficient sampling for Gaussian process inference using control variables. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 1681–1688, Cambridge, MA, 2009. MIT Press. [PDF].

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

Nonlinear Response

Nonlinear Response Models

Consider the model of transcription,

$$\frac{\mathrm{d}m_j\left(t\right)}{\mathrm{d}t} = b_j + s_j F\left(p\left(t\right)\right) - d_j m_j\left(t\right),$$

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

$$m_j\left(t\right) = \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j\left(t-u\right)} F_j\left(p\left(u\right)\right) \mathrm{d}u$$

MAP-Laplace Approximation

Laplace's method: approximate posterior mode as Gaussian

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

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

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

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

MAP-Laplace Approximation

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

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

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

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

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

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

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

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

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

Model Parameter Estimation

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

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

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

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

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

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

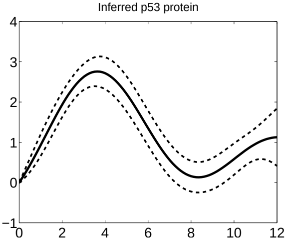
Michaelis-Menten Kinetics

Pei Gao

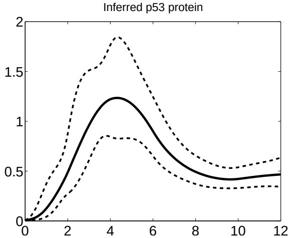
- The Michaelis-Menten activation model uses the following non-linearity

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

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



(a) Linear Response



(b) Laplace Approximation Nonlinear

Valdiation of Laplace Approximation

Michalis Titsias

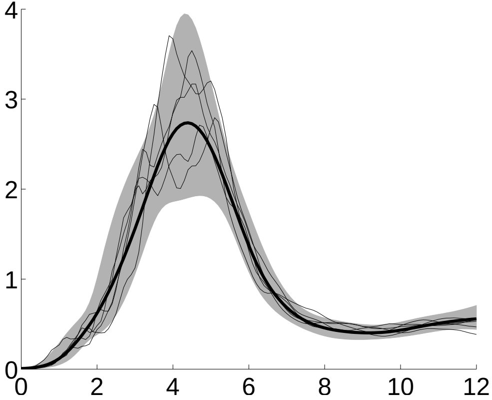


Figure: Laplace approximation error bars along with samples from the true posterior distribution.

SOS Response

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

LexA Experimental Description

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

Khanin et al. Model

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

$$m_i(t) = m_i^0 e^{-d_i t} + \frac{b_i}{d_i} + s_i e^{-d_i t} \int_0^t F(p(u)) e^{d_i u} \mathrm{d}u.$$

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

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

Khanin et al. Results

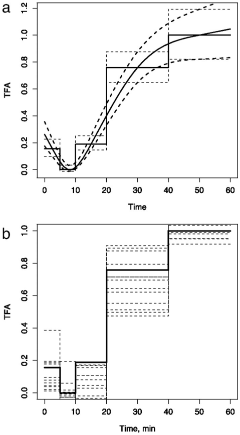


Figure: Fig. 2 from Khanin et al. (2006): Reconstructed activity level of master repressor LexA, following a UV dose of 40 J/m2.

Khanin et al. Results

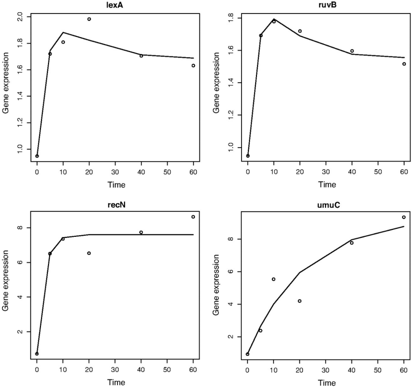


Figure: Fig. 3 from Khanin et al. (2006): Reconstructed profiles for four genes in the LexA SIM.

Repression Model

Pei Gao

- We can use the same model of repression,

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

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

  $$m_j\left(t\right) = \alpha_j e^{-d_j t} + \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j(t-u)} F_j(p\left(u\right)) \mathrm{d}u$$

Results for the repressor LexA

Pei Gao

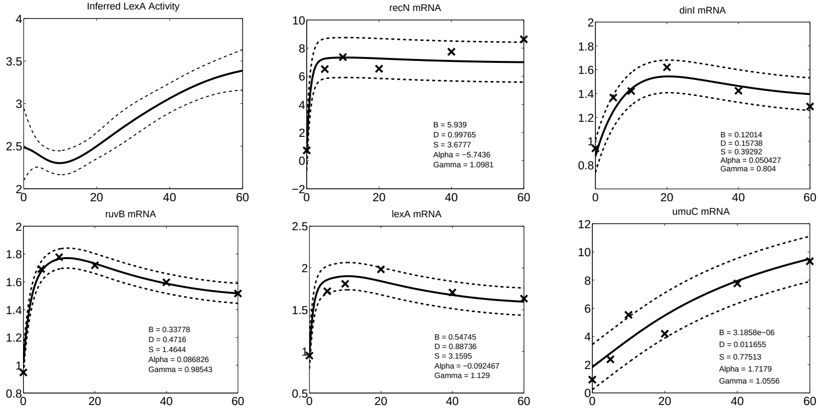


Figure: Our results using an MLP kernel. From Gao et al. (2008).

Use Samples to Represent Posterior

Michalis Titsias

- Sample in Gaussian processes

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

- Likelihood relates GP to data through

$$m_j\left(t\right) = \alpha_j e^{-d_j t} + \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j(t-u)} F_j(p\left(u\right))\mathrm{d}u$$

- We use *control points* for fast sampling.

MCMC for Non Linear Response

The Metropolis-Hastings algorithm

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

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

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

Sampling using control points

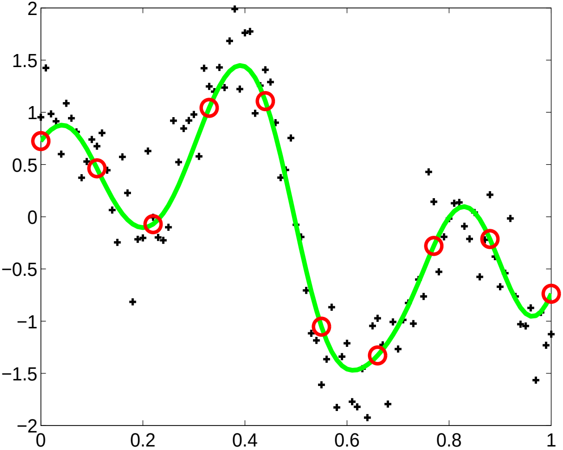
- Separate the points in $\mathbf{p}$ into two groups:
  - few control points $\mathbf{p}_c$
  - and the large majority of the remaining points $\mathbf{p}_\rho = \mathbf{p} \setminus \mathbf{p}_c$
- Sample the control points $\mathbf{p}_c$ using a proposal $q\left(\mathbf{p}_c^{(t+1)}|\mathbf{p}_c^{(t)}\right)$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

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

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

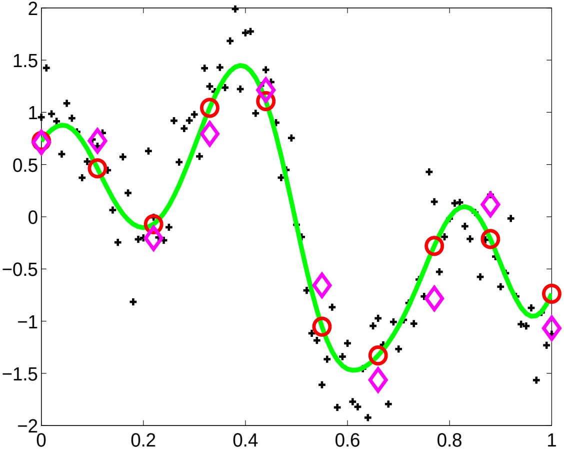
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



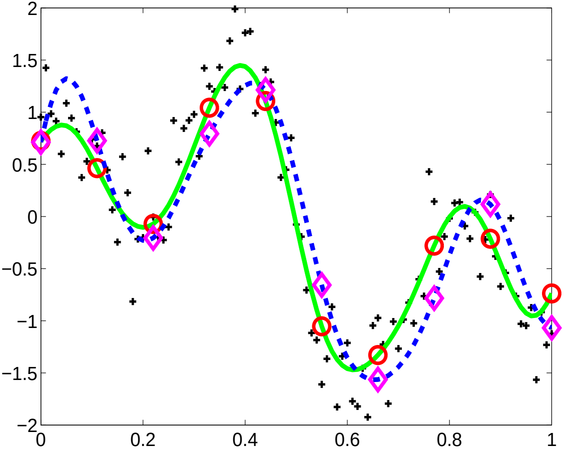
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



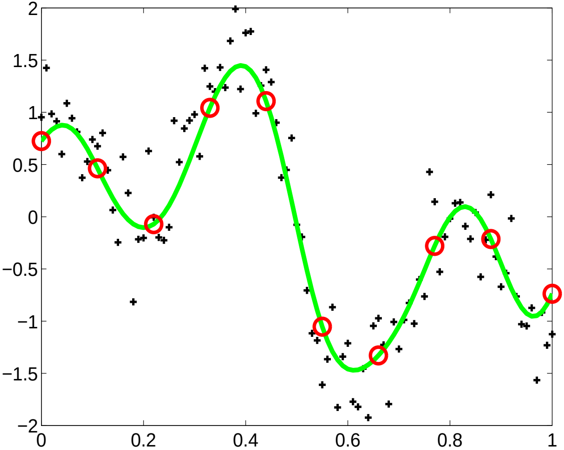
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



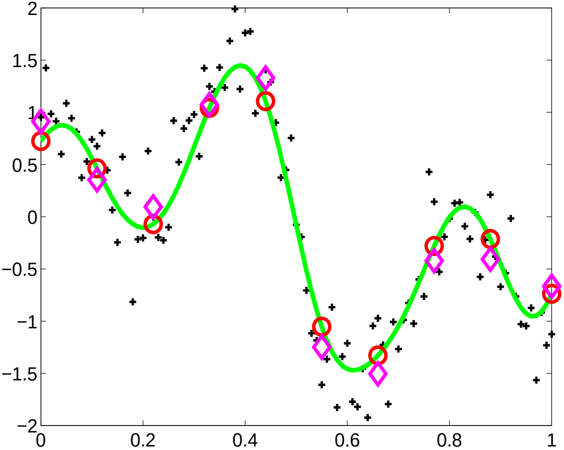
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



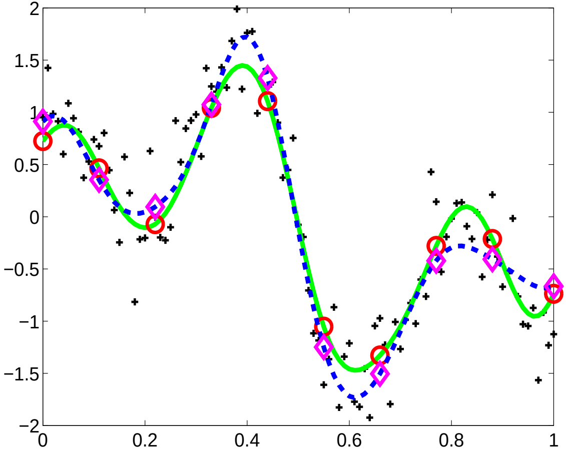
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



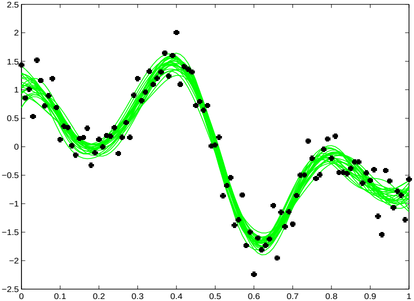
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



Sampling using control points

Few samples drawn during MCMC



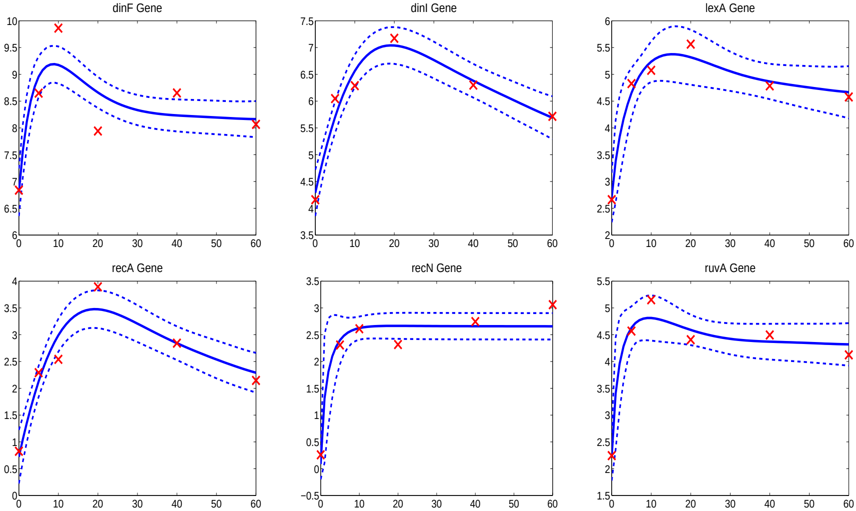
Results on SOS System

- Again consider the Michaelis-Menten kinetic equation

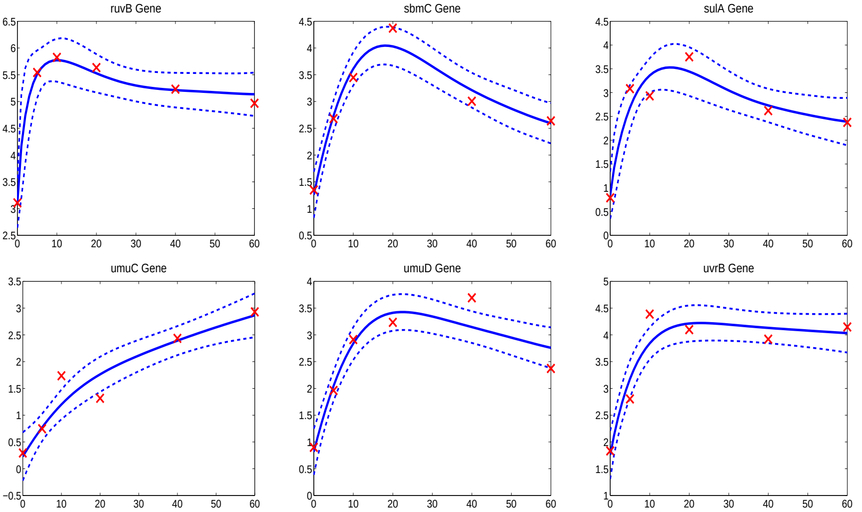
$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j \frac{1}{\exp(p(t)) + \gamma_j} - d_j m_j(t)$$

- We have 14 genes (5 kinetic parameters each)
- Gene expressions are available for $T = 6$ time slots
- TF (**p**) 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 **p** after burn in was between $15\% - 25\%$

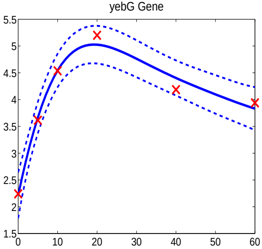
Results in E.coli data: Predicted gene expressions

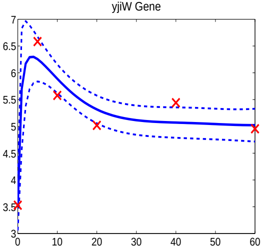


Results in E.coli data: Predicted gene expressions

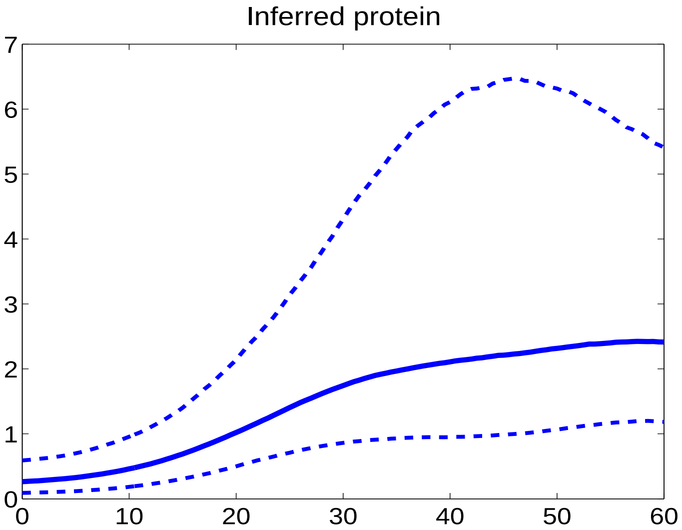


Results in E.coli data: Predicted gene expressions

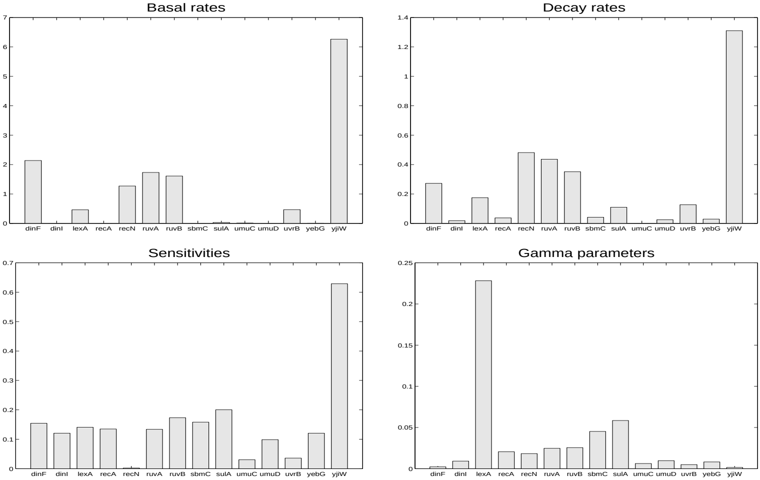




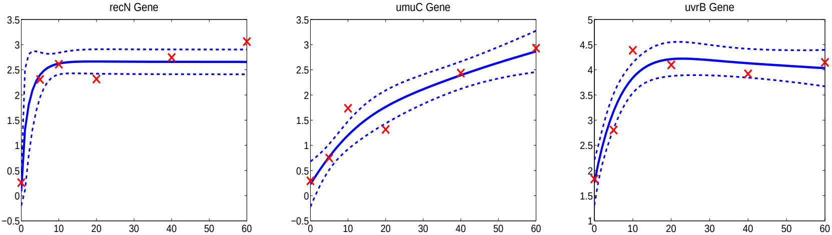
Results in E.coli data: Protein concentration

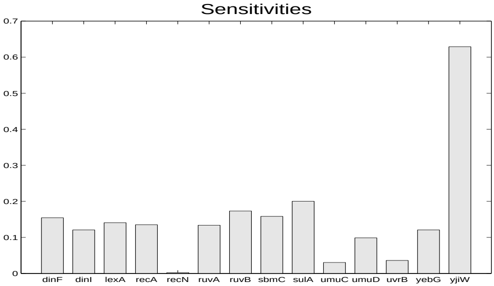


Results in E.coli data: Kinetic parameters

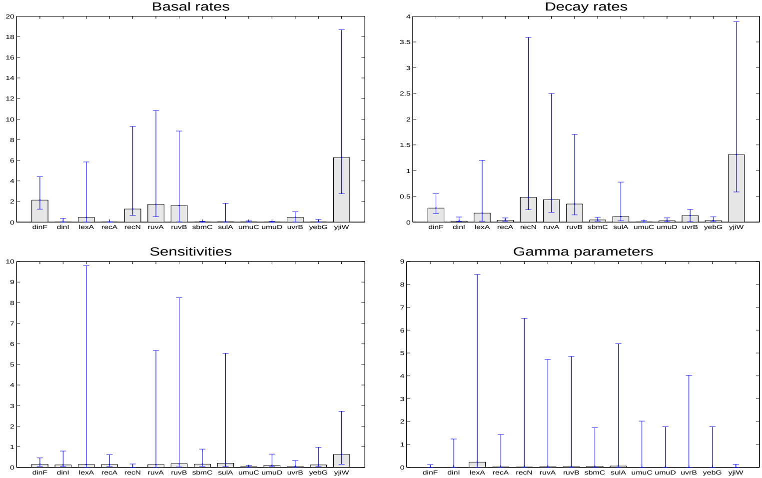


Results in E.coli data: Genes with low sensitivity value





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



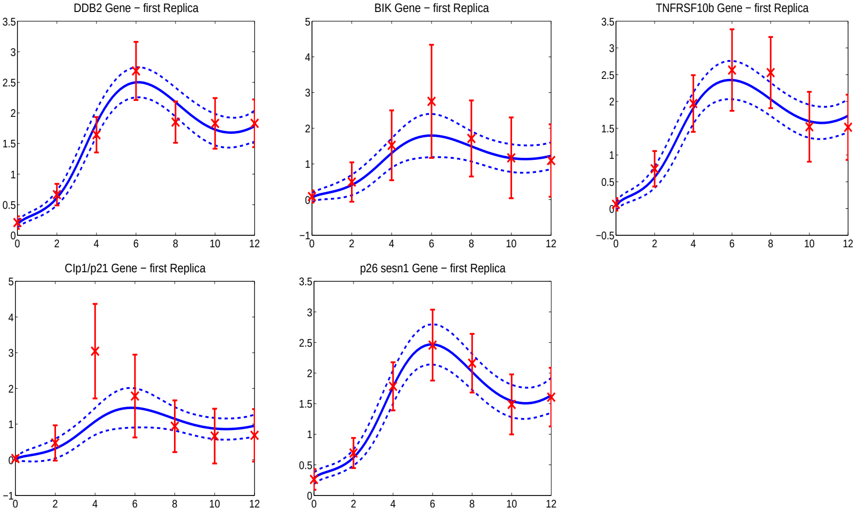
p53 System Again

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

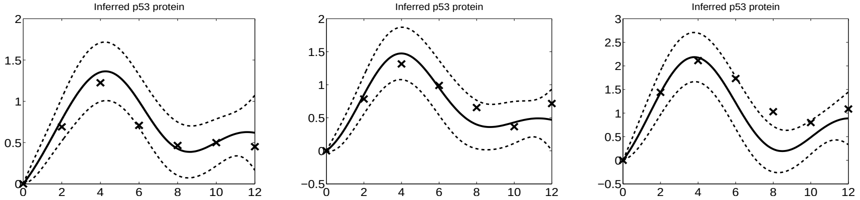
$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j \frac{\exp(p(t))}{\exp(p(t)) + \gamma_j} - d_j m_j(t)$$

- We have 5 genes
- Gene expressions are available for $T = 7$ times and there are 3 replicas of the time series data
- TF ($\mathbf{p}$) 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 $\mathbf{p}$ after burn in was between $15\% - 25\%$

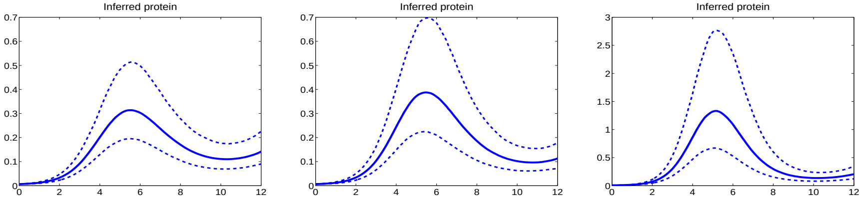
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica



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

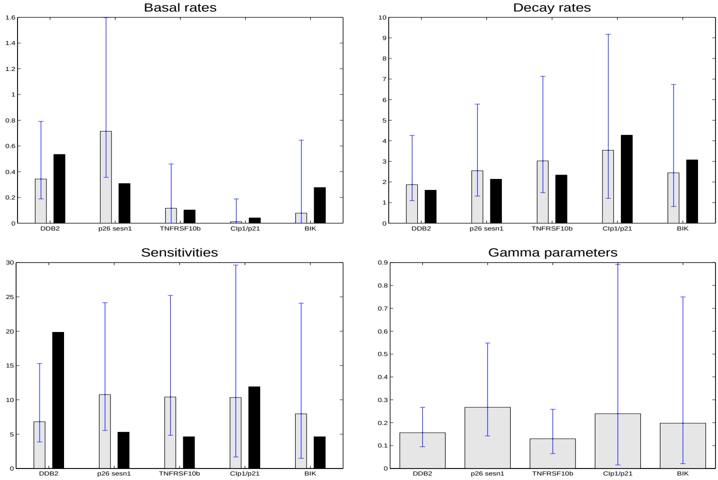


Linear model (Barenco et al. predictions are shown as crosses)



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