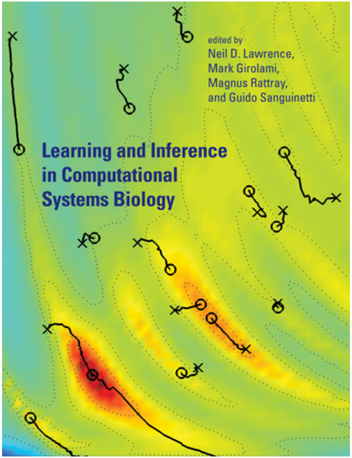
Multioutput Gaussian Processes

Neil D. Lawrence

BioPreDyn Workshop
Barcelona, 12th June 2012

Outline

Book


edited by
Neil D. Lawrence,
Mark Girolami,
Magnus Rattray,
and Guido Sanguinetti
Learning and Inference
in Computational
Systems Biology

?

Outline

Simple Markov Chain

- Assume 1-d latent state, a vector over time, $\mathbf{x} = [x_1 \dots x_T]$.
- Markov property,

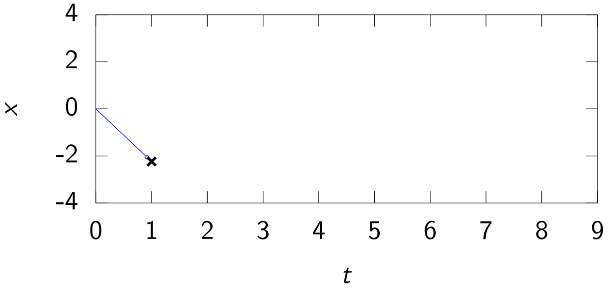
$$
\begin{aligned}
x_i =& x_{i-1} + \epsilon_i, \\
\epsilon_i \sim& \mathcal{N}(0, \alpha) \\
\implies x_i \sim& \mathcal{N}(x_{i-1}, \alpha)
\end{aligned}
$$

- Initial state,

$$x_0 \sim \mathcal{N}(0, \alpha_0)$$

- If $x_0 \sim \mathcal{N}(0, \alpha)$ we have a Markov chain for the latent states.
- Markov chain it is specified by an initial distribution (Gaussian) and a transition distribution (Gaussian).

Gauss Markov Chain

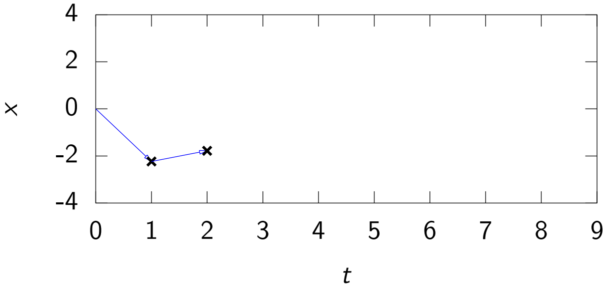


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_0 = 0.000, \quad \epsilon_1 = -2.24$$

$$x_1 = 0.000 - 2.24 = -2.24$$

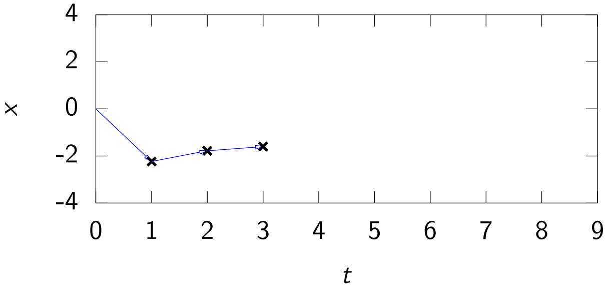
Gauss Markov Chain



$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_1 = -2.24, \quad \epsilon_2 = 0.457$$
$$x_2 = -2.24 + 0.457 = -1.78$$

Gauss Markov Chain

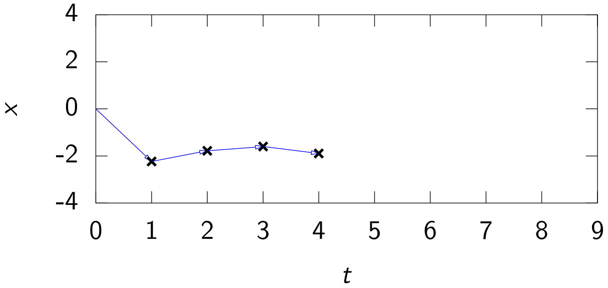


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_2 = -1.78, \qquad \epsilon_3 = 0.178$$

$$x_3 = -1.78 + 0.178 = -1.6$$

Gauss Markov Chain

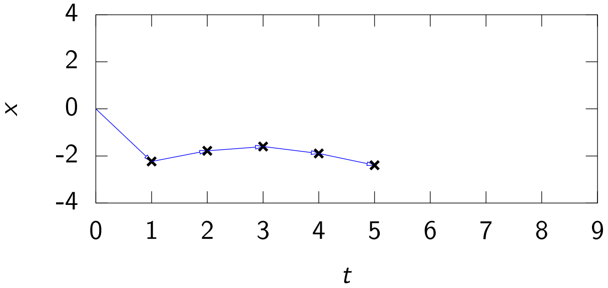


$$x_0 = 0, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, 1)$$

$$x_3 = -1.6, \quad \epsilon_4 = -0.292$$

$$x_4 = -1.6 - 0.292 = -1.89$$

Gauss Markov Chain

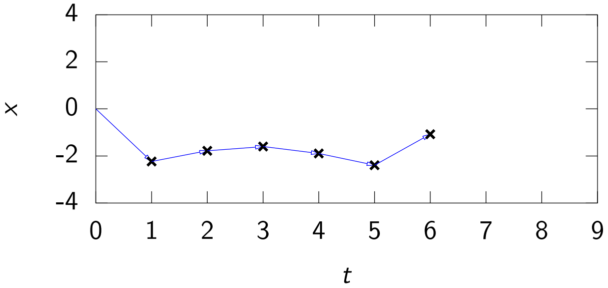


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_4 = -1.89, \quad \epsilon_5 = -0.501$$

$$x_5 = -1.89 - 0.501 = -2.39$$

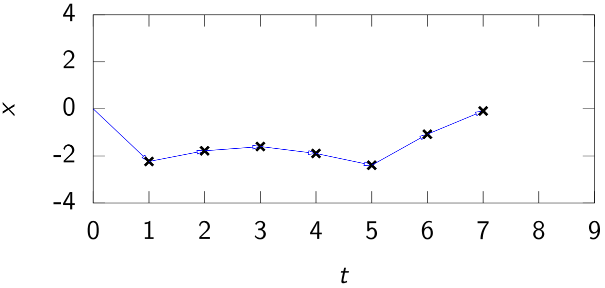
Gauss Markov Chain



$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_5 = -2.39, \quad \epsilon_6 = 1.32$$
$$x_6 = -2.39 + 1.32 = -1.08$$

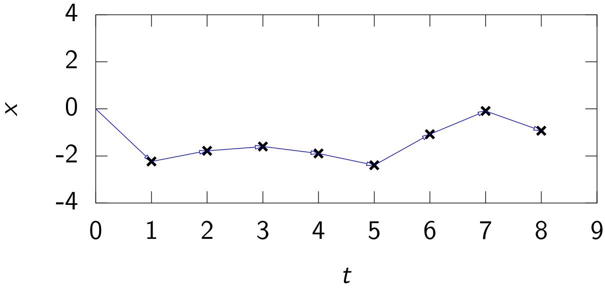
Gauss Markov Chain



$$x_0 = 0, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, 1)$$

$$x_6 = -1.08, \quad \epsilon_7 = 0.989$$
$$x_7 = -1.08 + 0.989 = -0.0881$$

Gauss Markov Chain

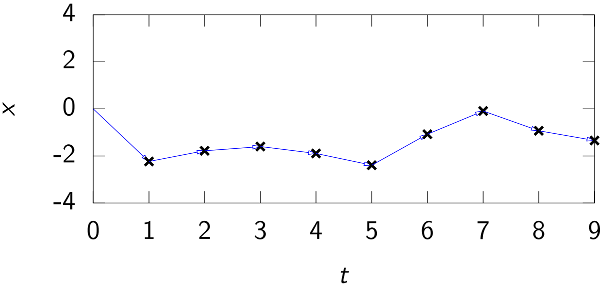


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_7 = -0.0881, \quad \epsilon_8 = -0.842$$

$$x_8 = -0.0881 - 0.842 = -0.93$$

Gauss Markov Chain



$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_8 = -0.93, \quad \epsilon_9 = -0.41$$

$$x_9 = -0.93 - 0.410 = -1.34$$

Multivariate Gaussian Properties: Reminder

If

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$$

and

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{W}\boldsymbol{\mu} + \mathbf{b}, \mathbf{W}\mathbf{C}\mathbf{W}^\top\right)$$

Multivariate Gaussian Properties: Reminder

Simplified: If

$$\mathbf{z} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}\right)$$

and

$$\mathbf{x} = \mathbf{W}\mathbf{z}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{W}\mathbf{W}^\top\right)$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_1 = \epsilon_1$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_2 = \epsilon_1 + \epsilon_2$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_3 = \epsilon_1 + \epsilon_2 + \epsilon_3$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_4 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_5 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5$$

Matrix Representation of Latent Variables

$$\mathbf{x} = \mathbf{L}_1 \times \boldsymbol{\epsilon}$$

Multivariate Process

- Since $\mathbf{x}$ is linearly related to ϵ we know $\mathbf{x}$ is a Gaussian process.
- Trick: we only need to compute the mean and covariance of $\mathbf{x}$ to determine that Gaussian.

Latent Process Mean

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \langle \mathbf{L}_1 \epsilon \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \epsilon \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \boldsymbol{\epsilon} \rangle$$

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

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \mathbf{0}$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{0}$$

Latent Process Covariance

$$\mathbf{x}\mathbf{x}^\top = \mathbf{L_1}\epsilon\epsilon^\top\mathbf{L_1}^\top$$
$$\mathbf{x}^\top = \epsilon^\top\mathbf{L}^\top$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \left\langle \mathbf{L}_1 \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \mathbf{L}_1^\top \right\rangle$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^{\top} \right\rangle = \mathbf{L}_1 \left\langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\top} \right\rangle \mathbf{L}_1^{\top}$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \mathbf{L}_1 \left\langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \right\rangle \mathbf{L}_1^\top$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right)$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^{\top} \right\rangle = \alpha \mathbf{L_1} \mathbf{L_1}^{\top}$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \epsilon$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right)$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

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

$$\Longrightarrow$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

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

$$\Longrightarrow$$

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{L}_1 \mathbf{L}_1^\top\right)$$

Covariance for Latent Process II

- Given
$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha\mathbf{I}\right) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha\mathbf{L_1}\mathbf{L_1}^\top\right).$$
Then
$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \Delta t\alpha\mathbf{I}\right) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \Delta t\alpha\mathbf{L_1}\mathbf{L_1}^\top\right).$$
where Δt is the time interval between observations.

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \alpha\Delta t\mathbf{I}), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t\mathbf{l}_{:,i}^\top\mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T - k$ are zero.

$$k_{i,j} = \alpha\Delta t \min(i, j)$$

define $\Delta ti = t_i$ so

$$k_{i,j} = \alpha \min(t_i, t_j) = k(t_i, t_j)$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \alpha\Delta t \mathbf{I}), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t \mathbf{l}_{:,i}^\top \mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T-k$ are zero.

$$k_{i,j} = \alpha\Delta t \min(i,j)$$

define $\Delta t i = t_i$ so

$$k_{i,j} = \alpha \min(t_i, t_j) = k(t_i, t_j)$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \alpha\Delta t\mathbf{I}), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t\mathbf{l}_{:,i}^\top\mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T-k$ are zero.

$$k_{i,j} = \alpha\Delta t \min(i,j)$$

define $\Delta ti = t_i$ so

$$k_{i,j} = \alpha \min(t_i, t_j) = k(t_i, t_j)$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t\mathbf{l}_{:,i}^\top\mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T-k$ are zero.

$$k_{i,j} = \alpha\Delta t\min(i,j)$$

define $\Delta ti = t_i$ so

$$k_{i,j} = \alpha\min(t_i,t_j) = k(t_i,t_j)$$

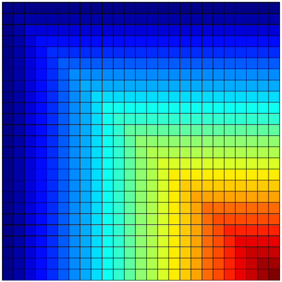
Covariance Functions

Where did this covariance matrix come from?

Markov Process

$$k\left(t, t'\right) = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



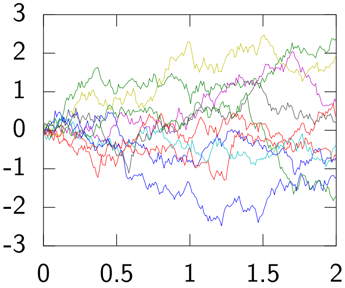
Covariance Functions

Where did this covariance matrix come from?

Markov Process

$$k\left(t, t'\right) = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



Can a Biologist Fix a Radio? ?

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.
- ? 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? **?**

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.
- **?** 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? **?**

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.
- **?** 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? ?

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.
- **?** 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? **?**

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.
- **?** 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? **?**

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.
- **?** 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? **?**

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.
- **?** 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? **?**

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.
- **?** 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.

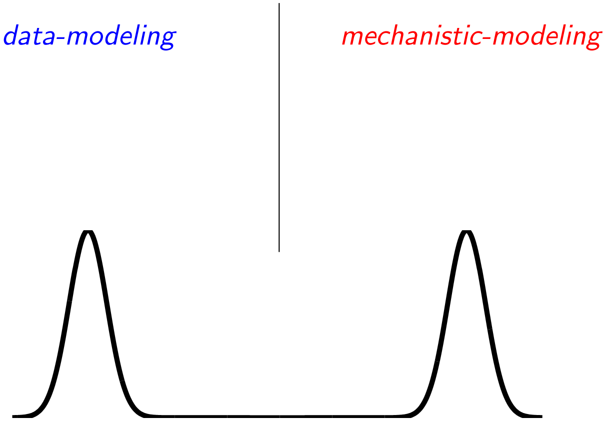
Coregulation of Gene Expression

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



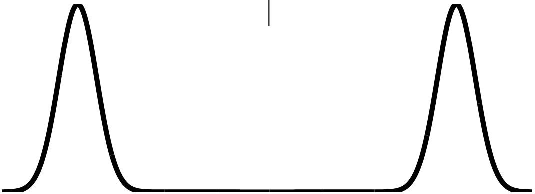
Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

data-modeling

let the data "speak"

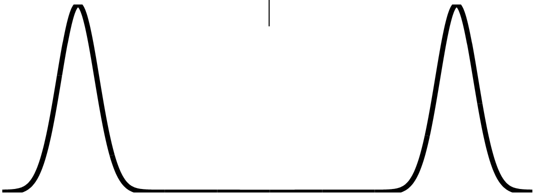
mechanistic-modeling



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

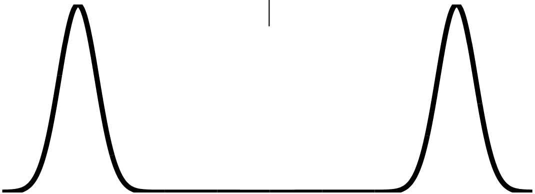
data-modeling

let the data "speak"

computational models

mechanistic-modeling

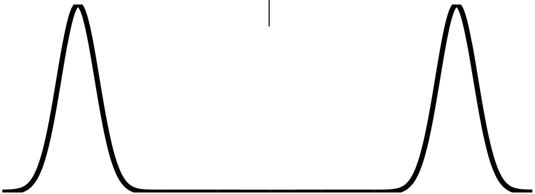
impose physical laws



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

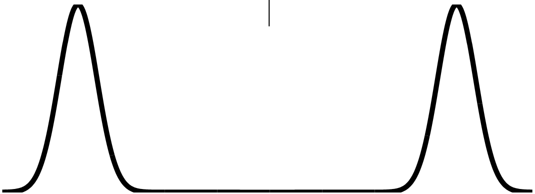
data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws
computational models	systems models



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

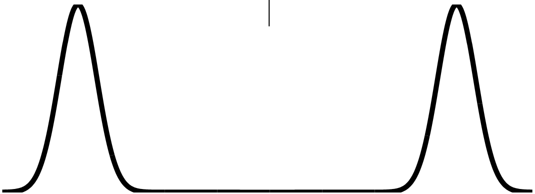
data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws
computational models	systems models
adaptive models	



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

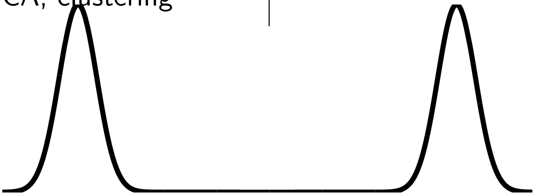
data-modeling	*mechanistic-modeling*
let the data “speak”	impose physical laws
computational models	systems models
adaptive models	differential equations



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

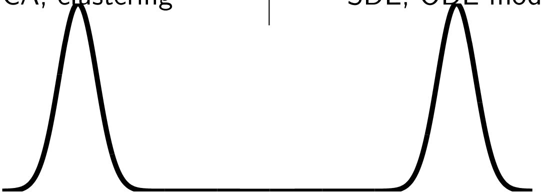
data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws
computational models	systems models
adaptive models	differential equations
PCA, clustering	



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

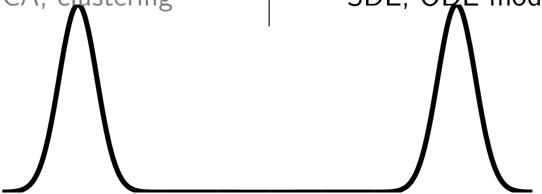
data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws
computational models	systems models
adaptive models	differential equations
PCA, clustering	SDE, ODE models



Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

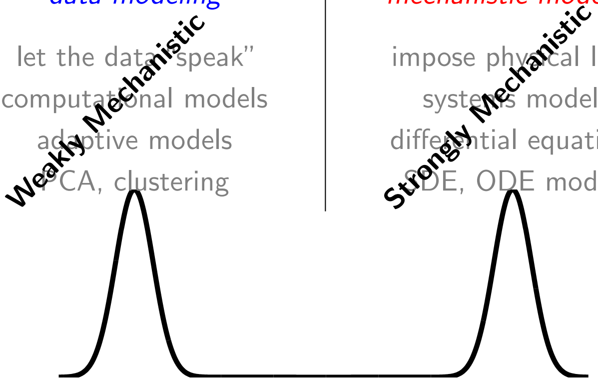
data-modeling	*mechanistic-modeling*
let the data "speak"	impose physical laws
computational models	systems models
adaptive models	differential equations
PCA, clustering	SDE, ODE models



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



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

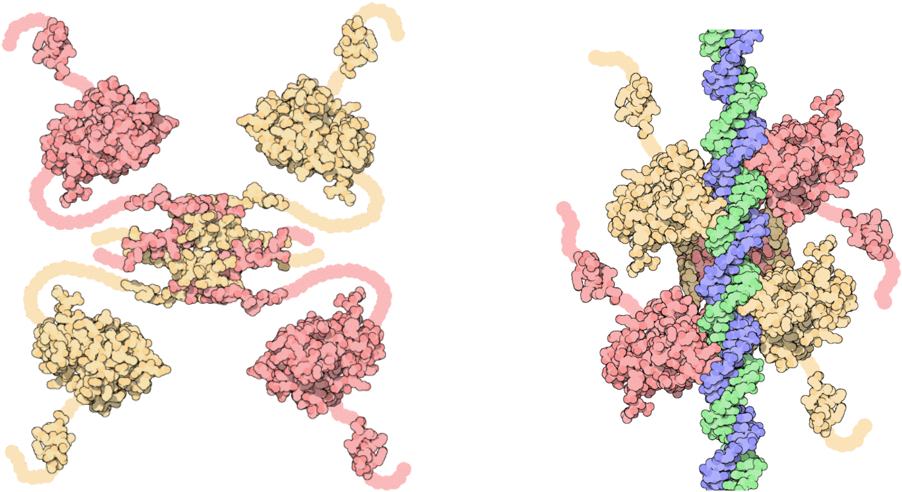


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the"Molecule of the Month" feature).

p53

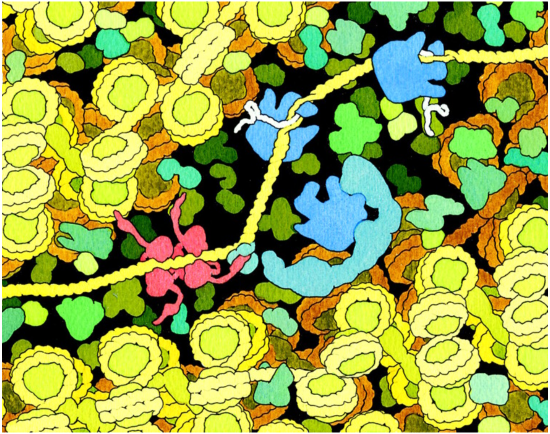


Figure: Repair of DNA damage by p53. Image from **?**.

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).

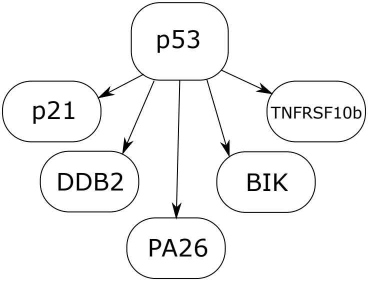


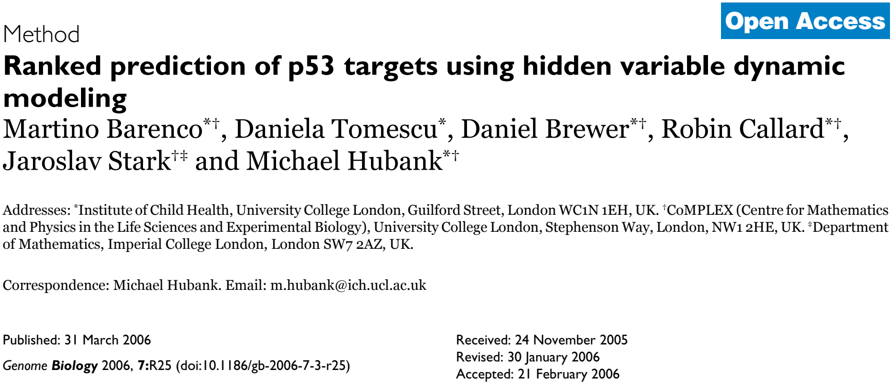
Figure: p53 SIM network motif as modelled by **?**.

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.

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

  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.

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

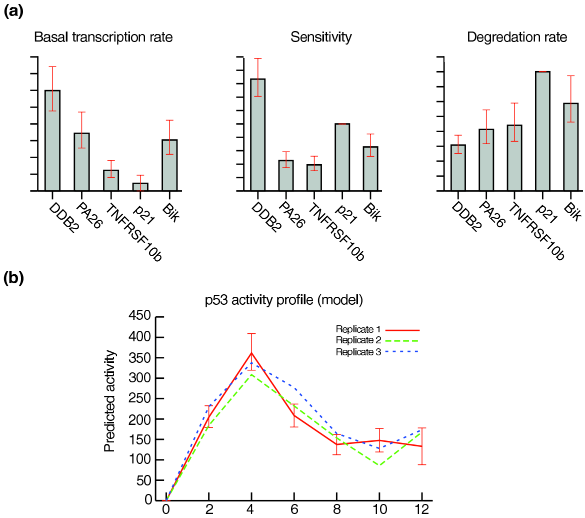


Figure: Results from ?. Top is parameter estimates. Bottom is inferred profile.

Response to p53 ...

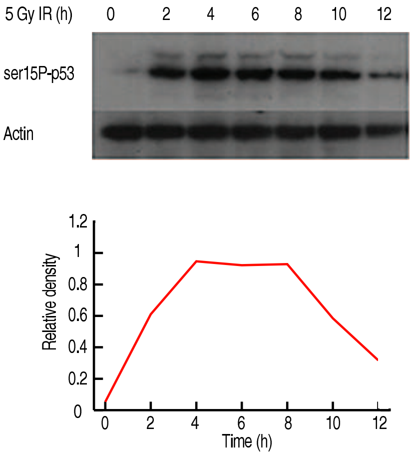


Figure: Results from **?**. Activity profile of p53 was measured by Western blot to determine the levels of ser-15 phosphorylated p53 (ser15P-p53).

Example: Transcriptional Regulation

- First Order Differential Equation

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

- 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

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

- 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.

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.

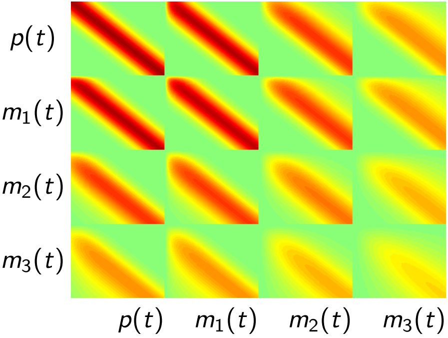
Covariance for Transcription Model

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



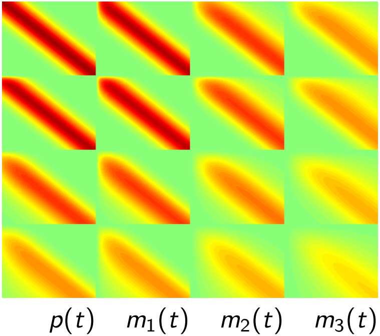
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



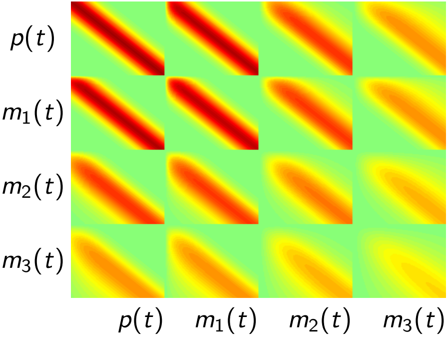
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

- 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`

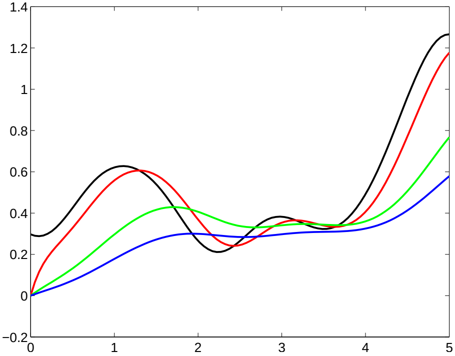


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

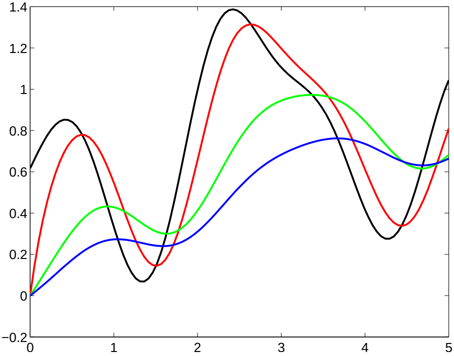


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

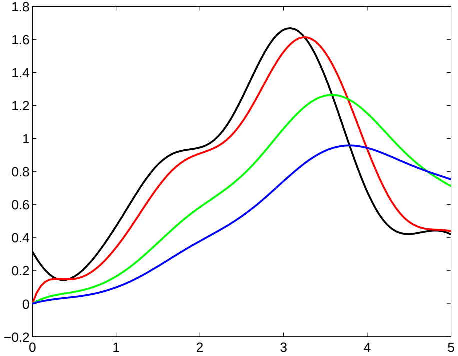


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

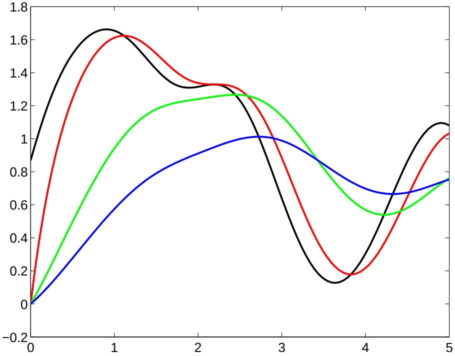
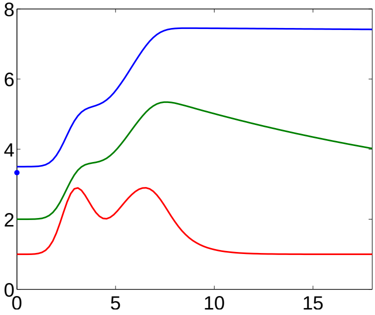


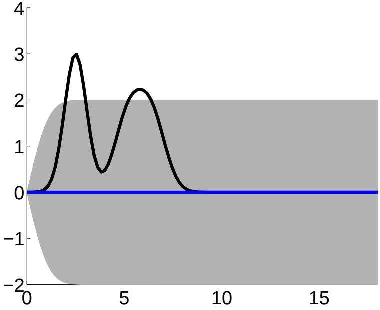
Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



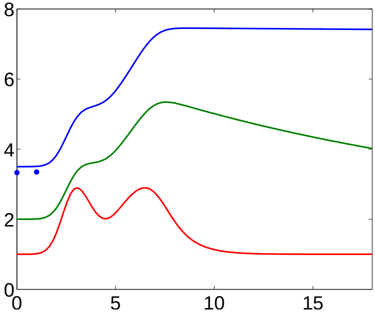
True "gene profiles" and noisy observations.



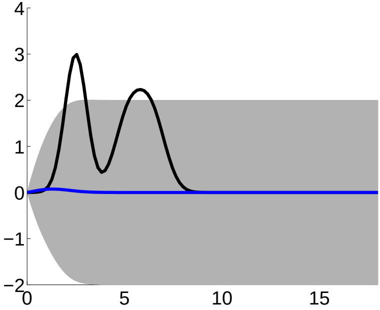
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



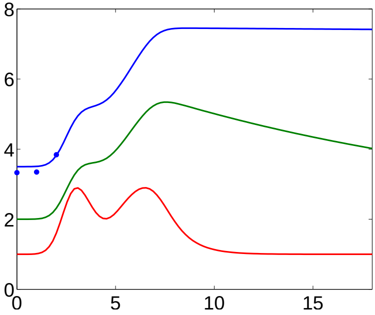
True "gene profiles" and noisy observations.



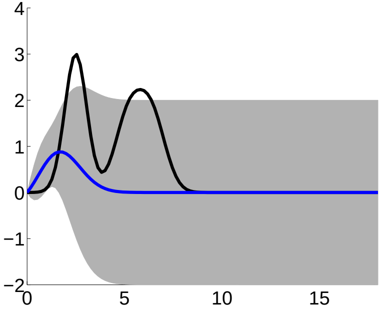
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



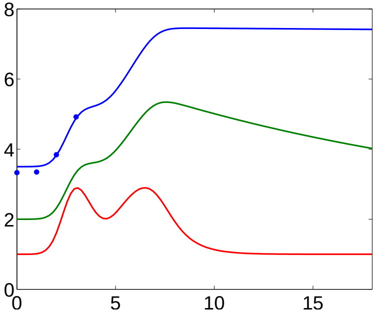
True "gene profiles" and noisy observations.



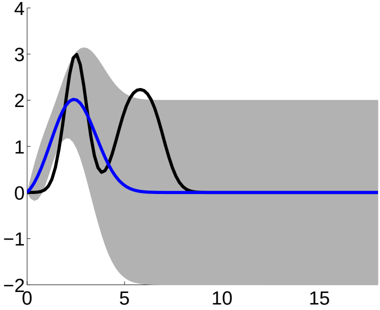
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



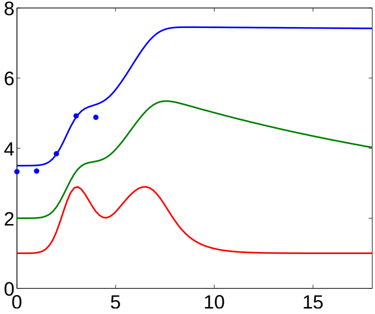
True “gene profiles” and noisy observations.



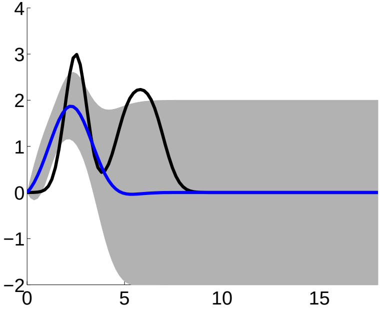
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



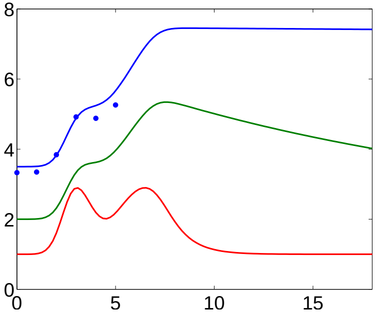
True "gene profiles" and noisy observations.



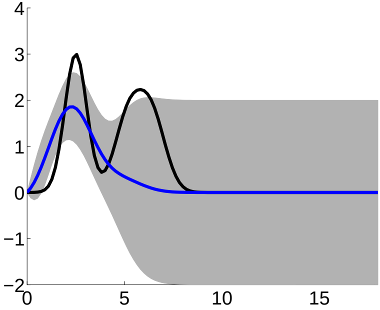
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



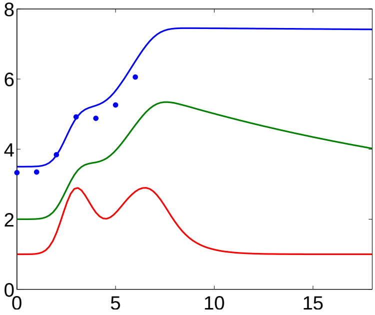
True "gene profiles" and noisy observations.



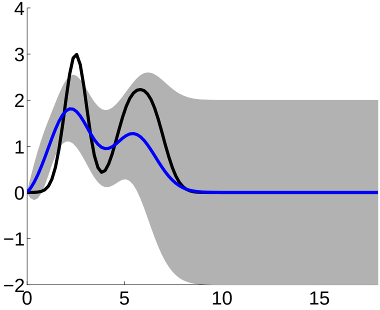
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



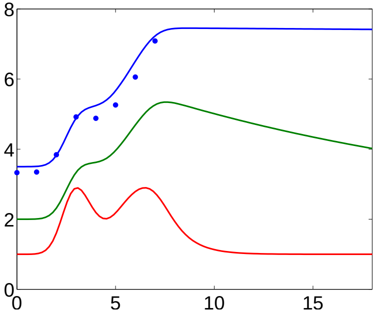
True “gene profiles” and noisy observations.



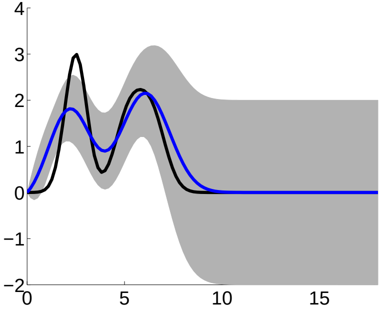
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



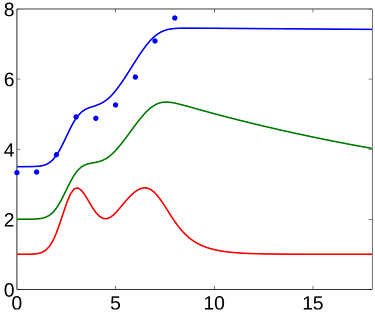
True “gene profiles” and noisy observations.



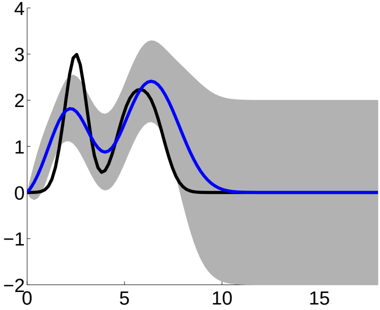
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



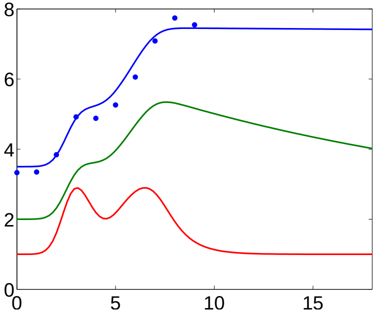
True "gene profiles" and noisy observations.



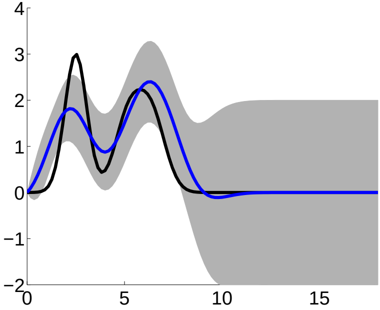
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



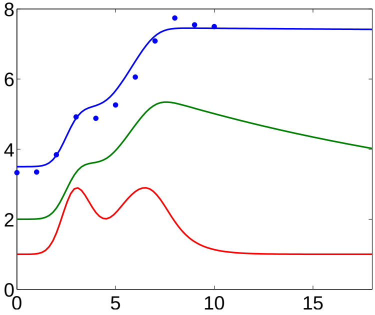
True "gene profiles" and noisy observations.



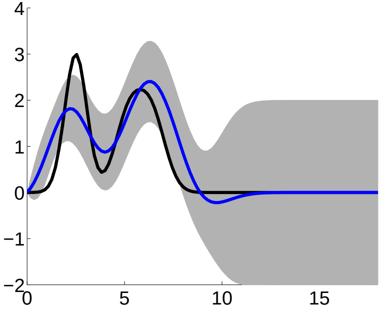
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



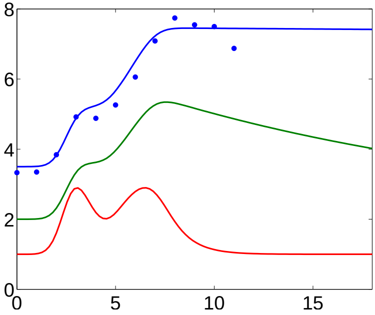
True "gene profiles" and noisy observations.



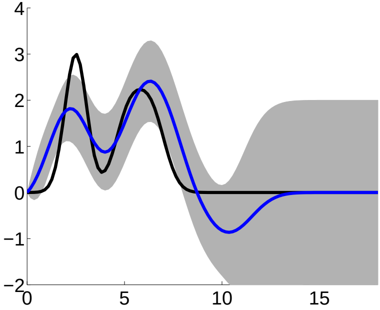
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



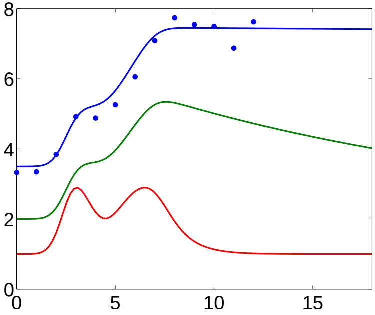
True "gene profiles" and noisy observations.



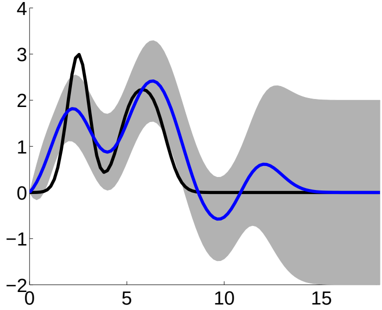
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



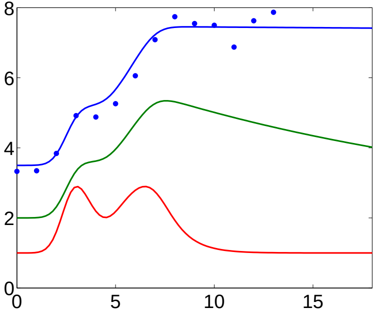
True "gene profiles" and noisy observations.



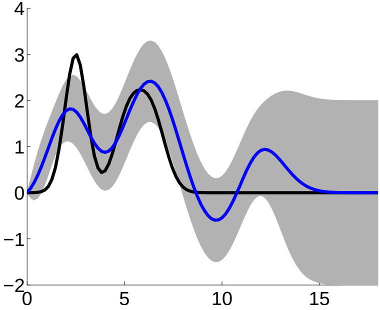
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



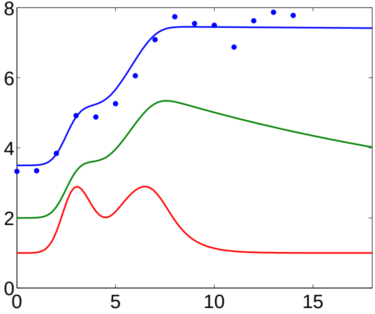
True "gene profiles" and noisy observations.



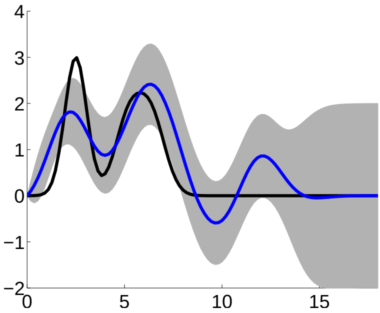
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



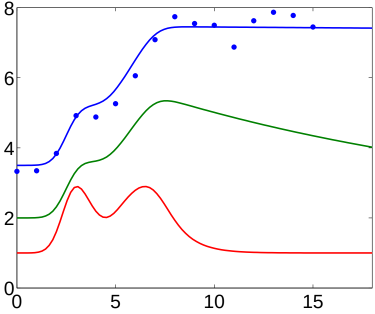
True “gene profiles” and noisy observations.



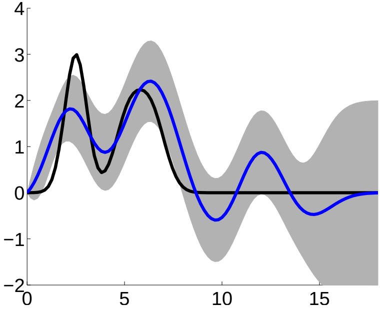
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



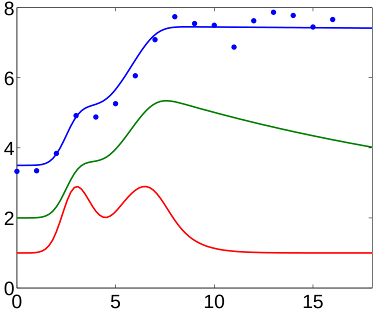
True “gene profiles” and noisy observations.



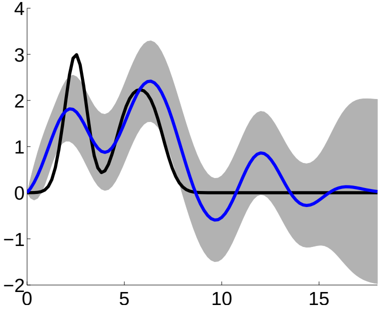
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



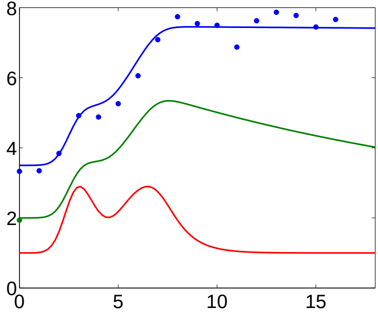
True “gene profiles” and noisy observations.



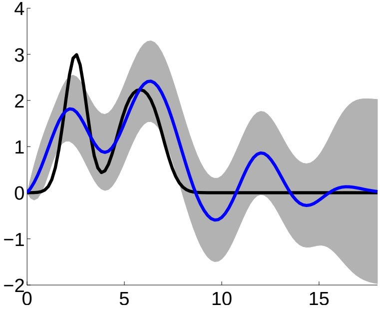
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



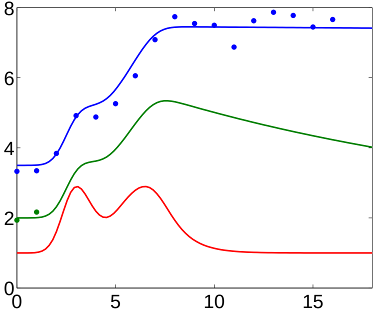
True "gene profiles" and noisy observations.



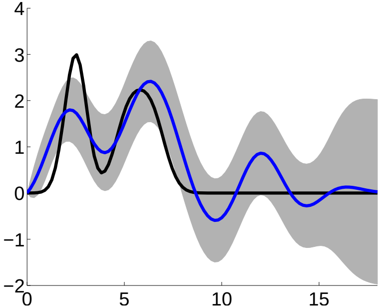
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



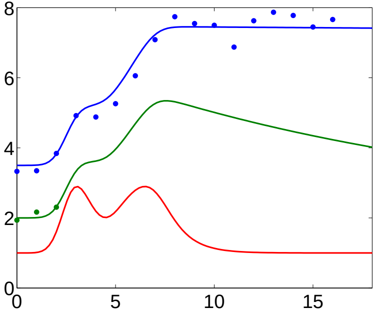
True “gene profiles” and noisy observations.



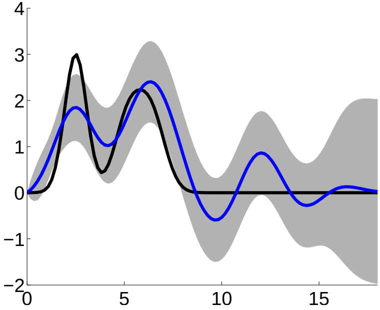
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



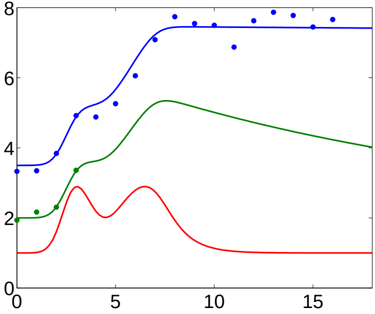
True “gene profiles” and noisy observations.



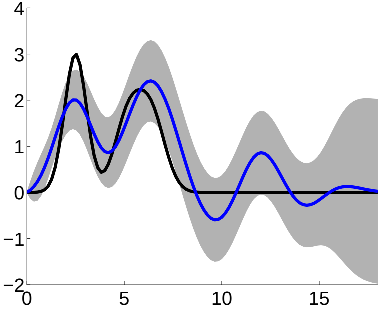
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



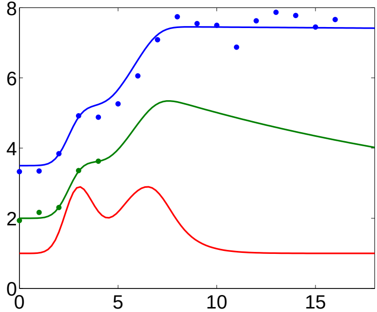
True "gene profiles" and noisy observations.



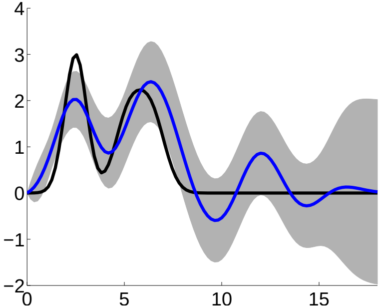
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



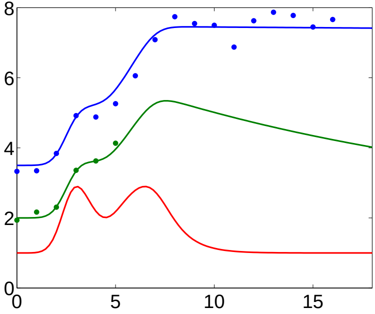
True "gene profiles" and noisy observations.



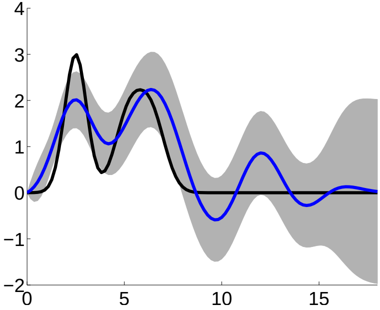
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



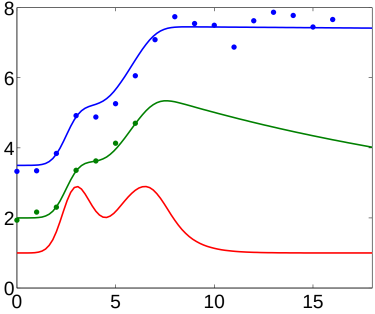
True "gene profiles" and noisy observations.



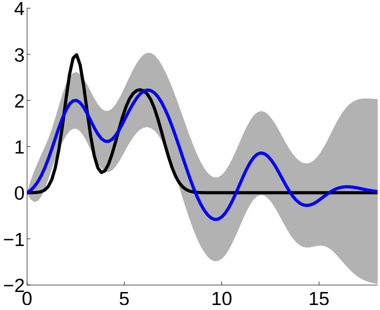
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



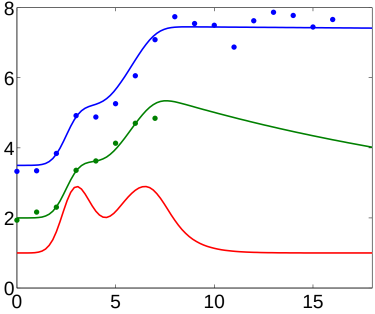
True "gene profiles" and noisy observations.



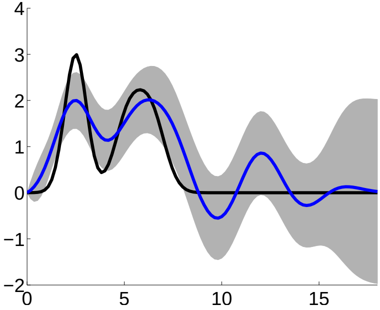
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



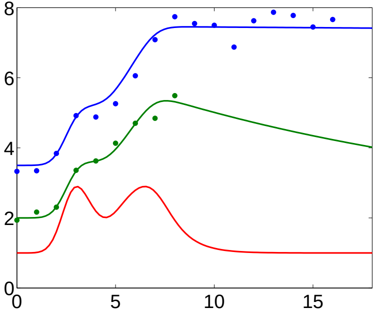
True “gene profiles” and noisy observations.



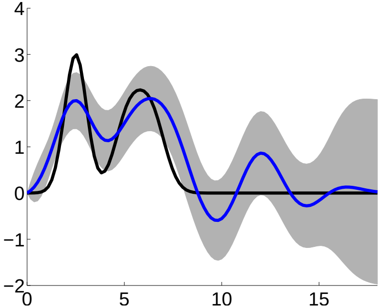
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



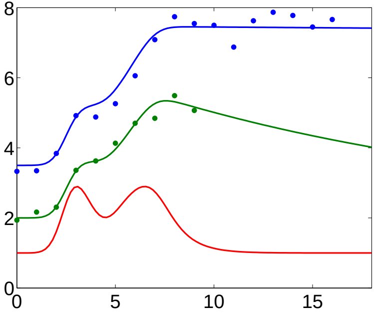
True "gene profiles" and noisy observations.



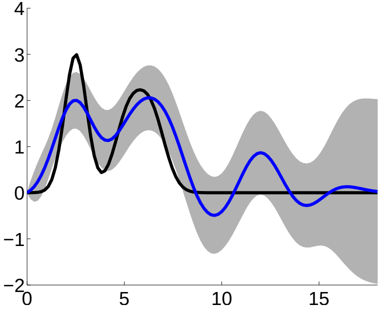
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



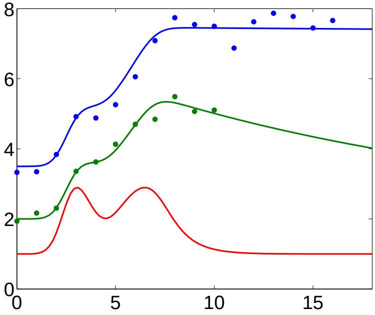
True “gene profiles” and noisy observations.



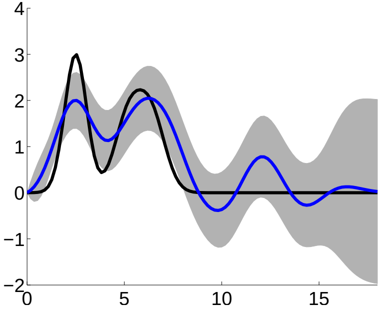
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



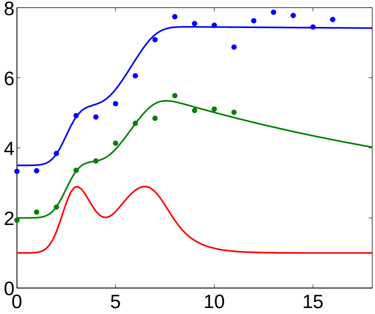
True "gene profiles" and noisy observations.



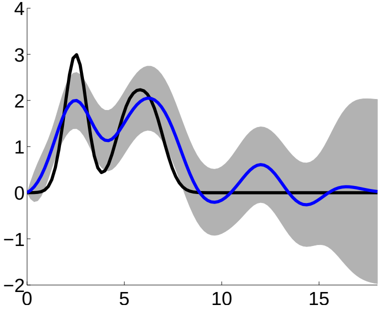
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



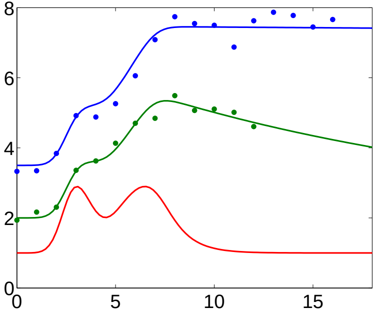
True “gene profiles” and noisy observations.



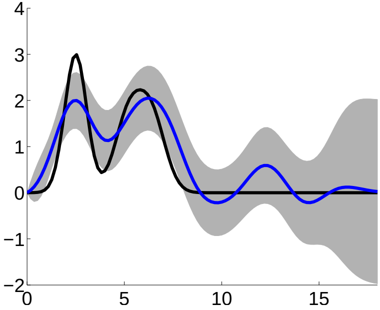
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



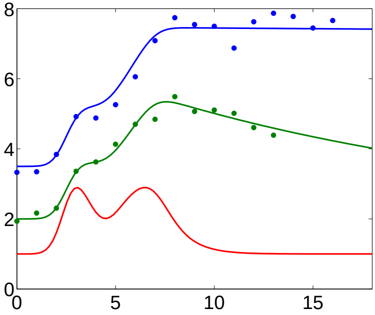
True “gene profiles” and noisy observations.



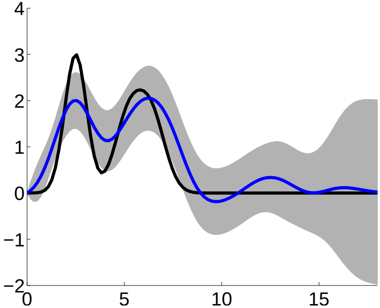
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



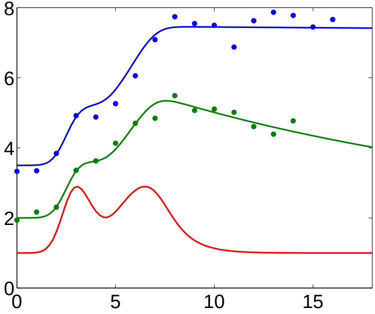
True "gene profiles" and noisy observations.



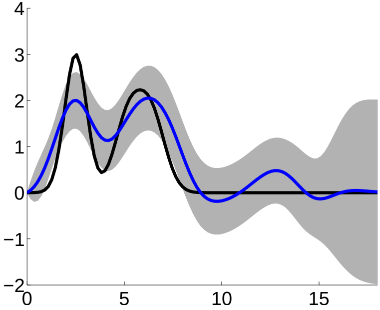
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



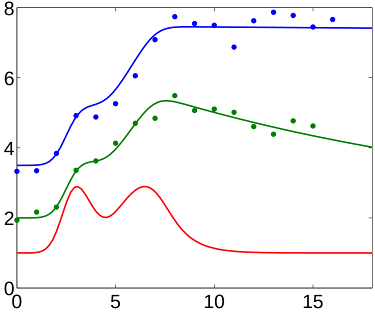
True "gene profiles" and noisy observations.



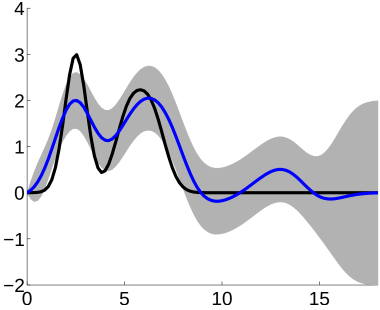
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



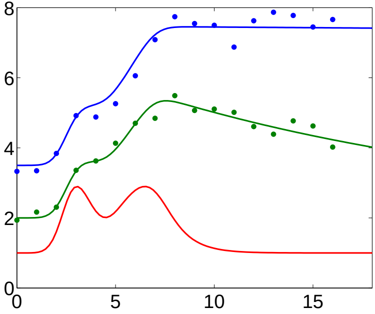
True “gene profiles” and noisy observations.



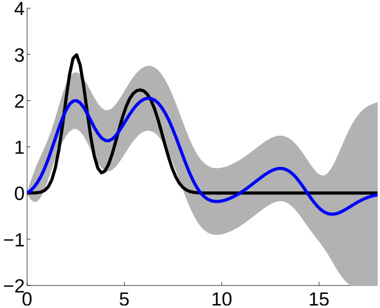
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



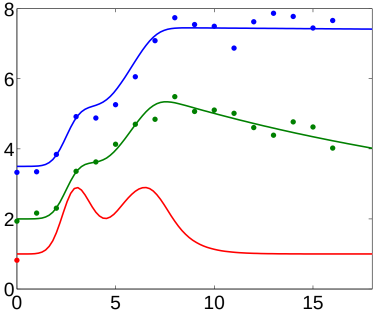
True "gene profiles" and noisy observations.



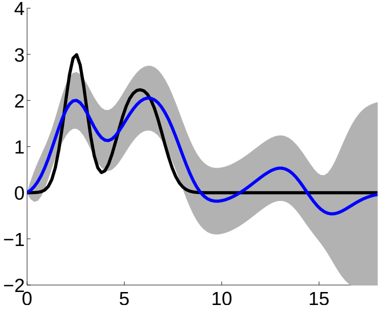
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



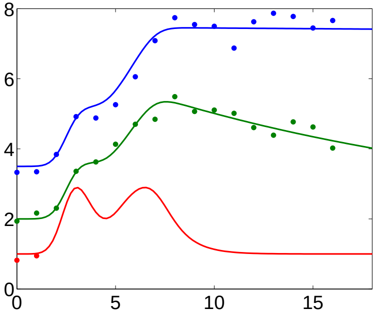
True “gene profiles” and noisy observations.



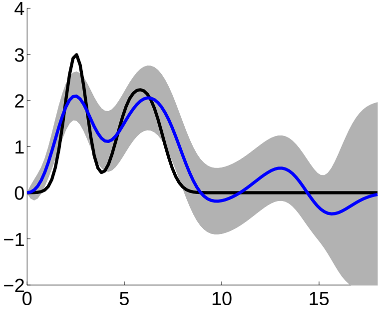
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



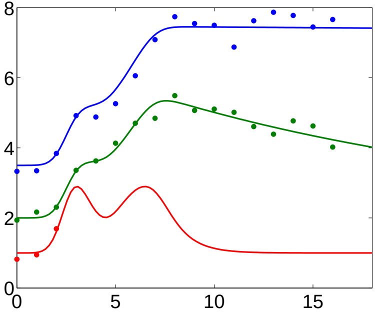
True "gene profiles" and noisy observations.



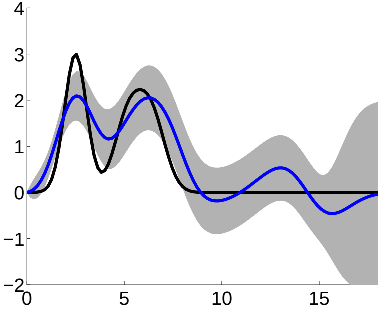
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



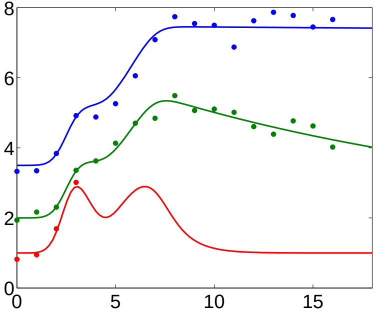
True "gene profiles" and noisy observations.



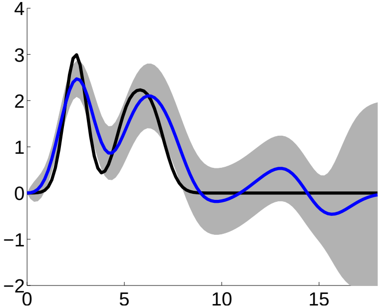
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



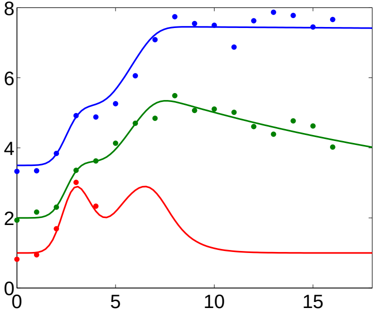
True "gene profiles" and noisy observations.



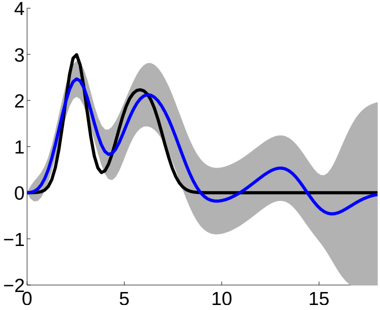
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



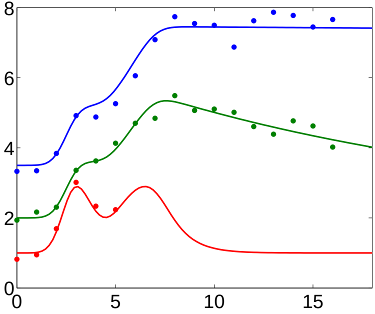
True “gene profiles” and noisy observations.



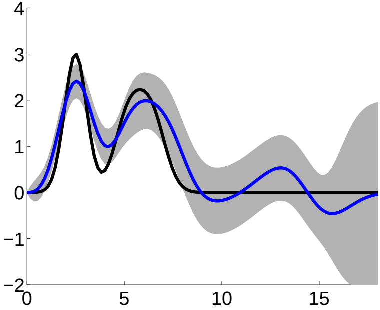
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



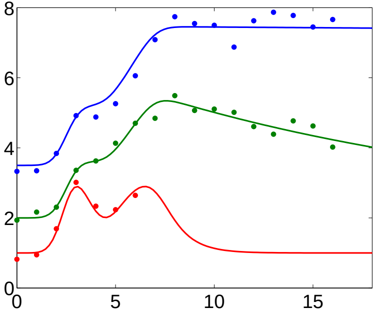
True "gene profiles" and noisy observations.



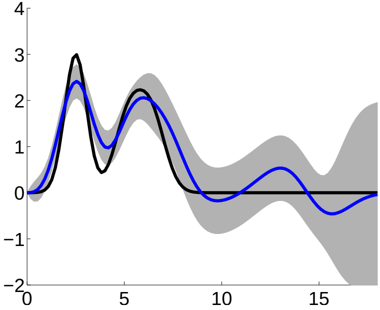
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



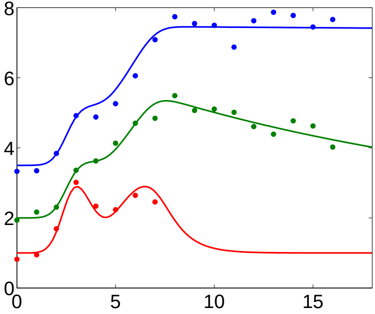
True "gene profiles" and noisy observations.



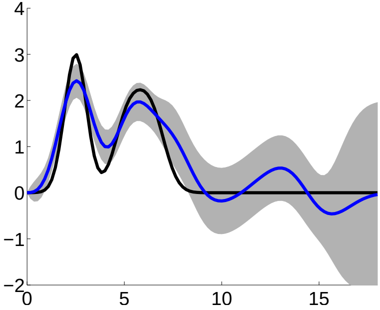
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



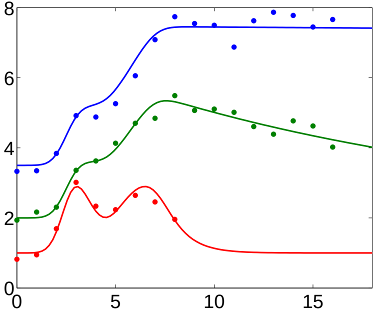
True “gene profiles” and noisy observations.



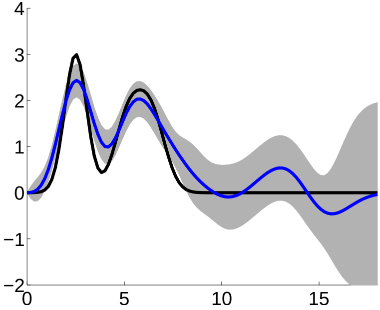
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



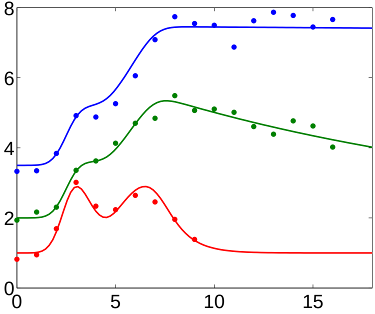
True "gene profiles" and noisy observations.



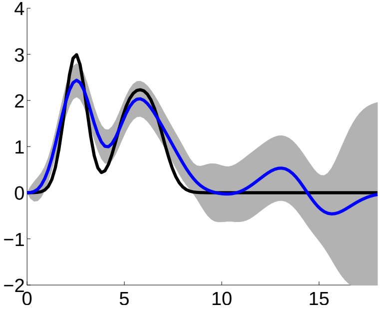
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



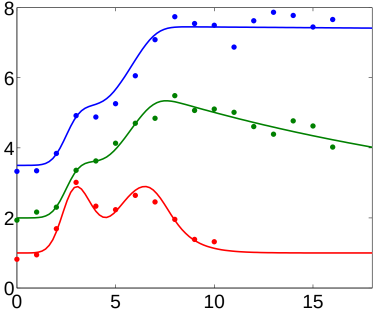
True “gene profiles” and noisy observations.



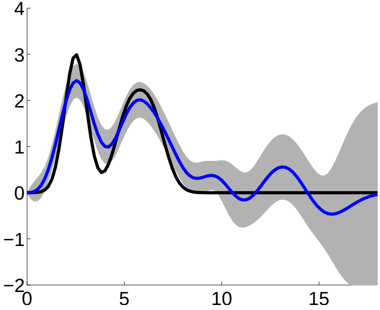
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



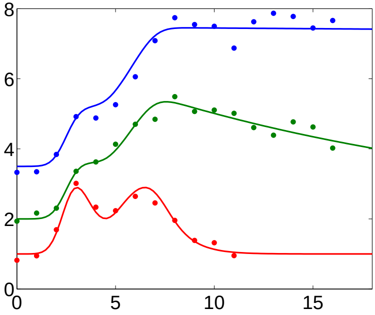
True "gene profiles" and noisy observations.



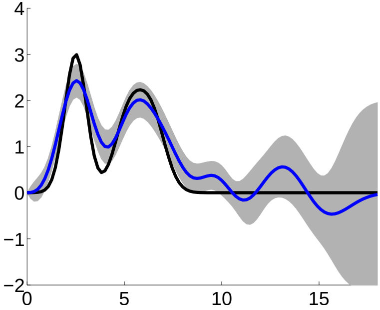
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



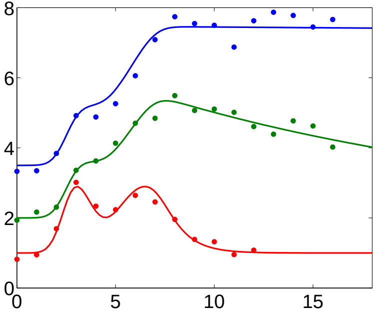
True "gene profiles" and noisy observations.



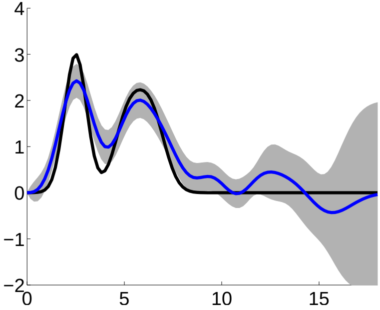
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



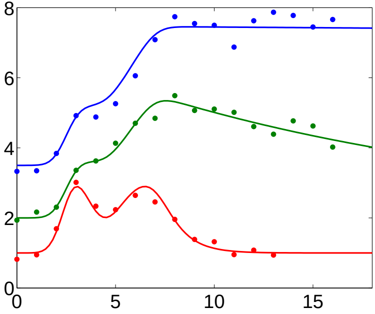
True "gene profiles" and noisy observations.



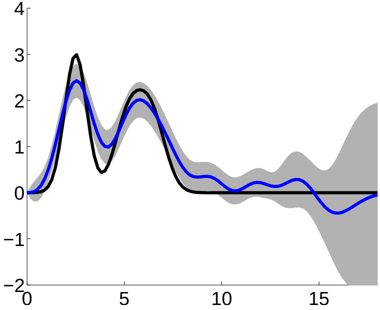
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



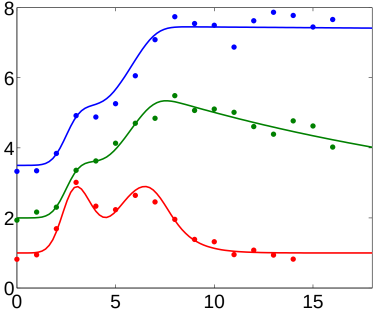
True "gene profiles" and noisy observations.



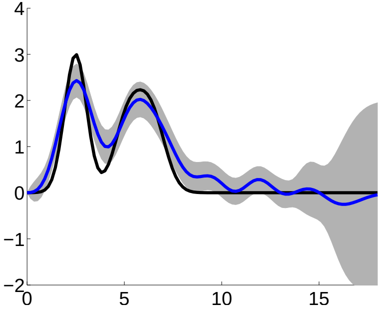
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



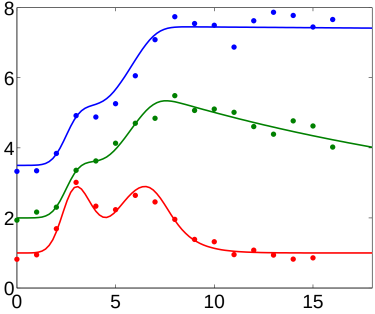
True “gene profiles” and noisy observations.



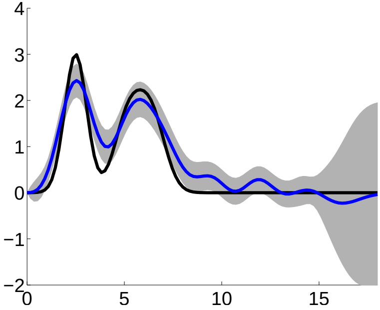
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



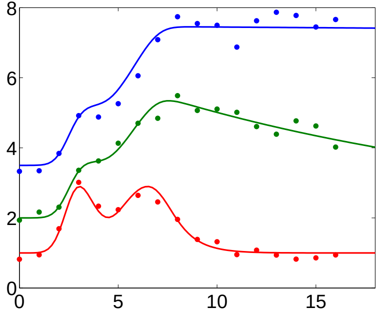
True “gene profiles” and noisy observations.



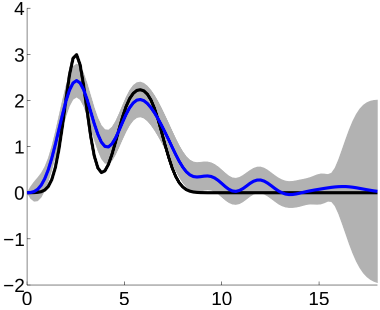
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



True “gene profiles” and noisy observations.



Inferred transcription factor activity.

Gene Expression Example

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

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

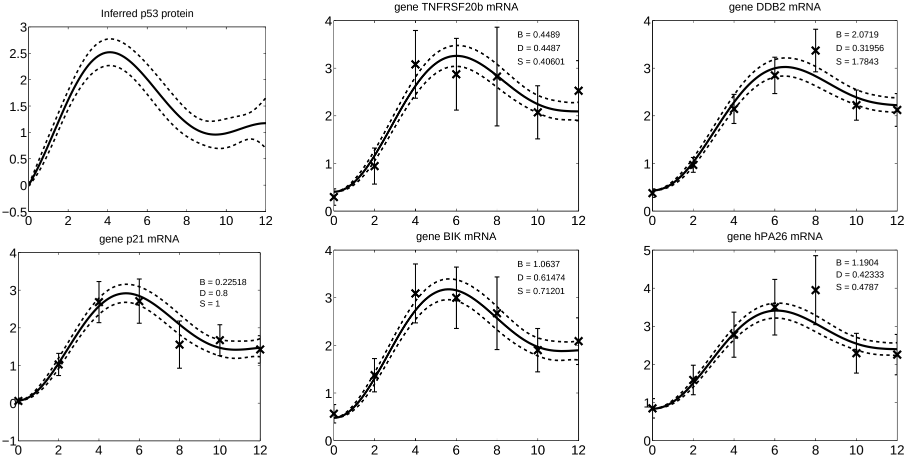
ABSTRACT

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

(?)

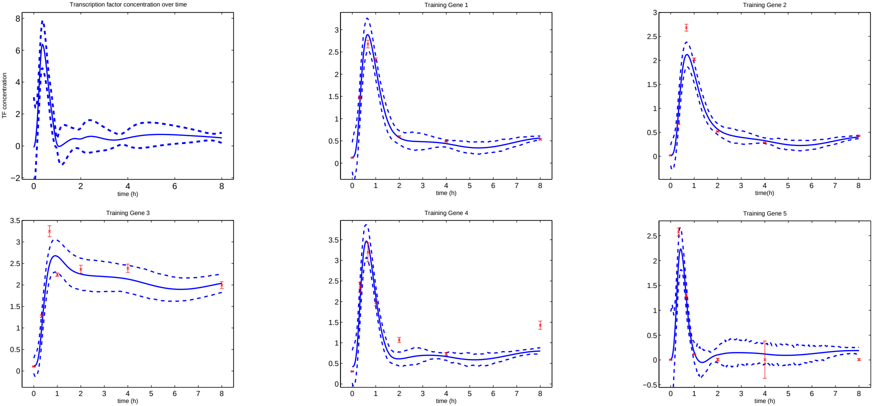


Ranking with ERK Signalling

- 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.

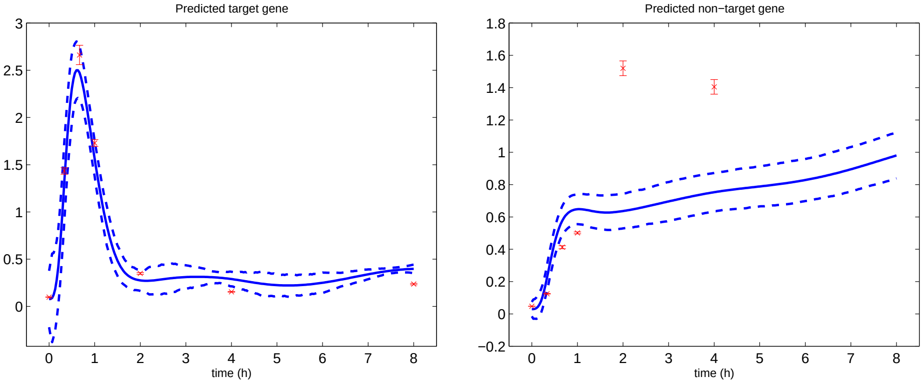
Elk-1 (MLP covariance)

Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



Outline

PNAS

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

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

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

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

We present a computational method for identifying potential targets of a transcription factor (TF) using wild-type gene expression time series data. For each putative target gene we fit a simple differential equation model of transcriptional regulation, and the used for genome-wide scoring of putative target gen
is required to apply our method is wild-type time seri
lected over a period where TF activity is changing. Ou
allows for complementary evidence from expression

Cascaded Differential Equations

(?)

- 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

(?)

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

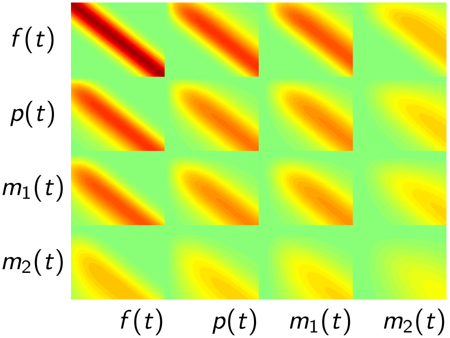
RBF covariance function for $f(t)$

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

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

- 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`

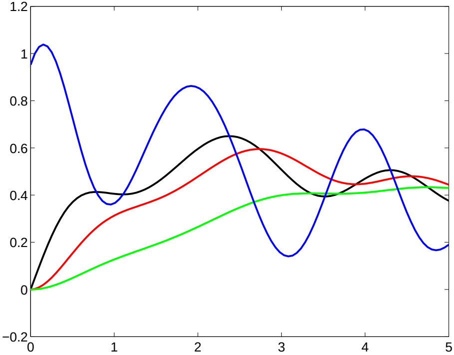


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

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

- `disimSample`

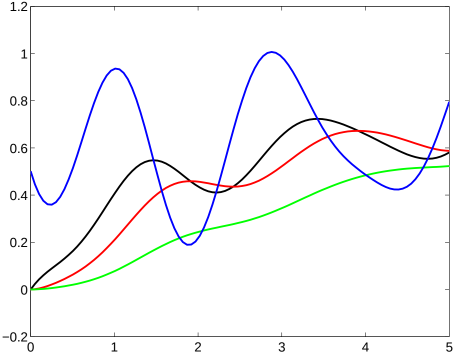


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

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

- `disimSample`

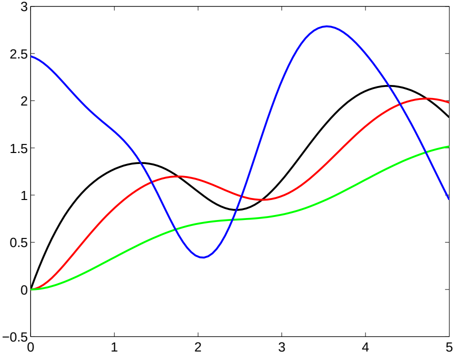


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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

- `disimSample`

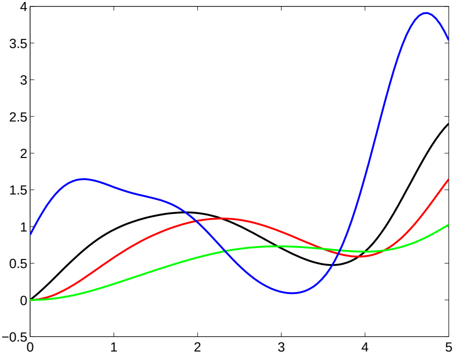


Figure: Joint samples from the ODE covariance, *blue*: $f(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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) (**?**).

Results for Twi using the Cascade model

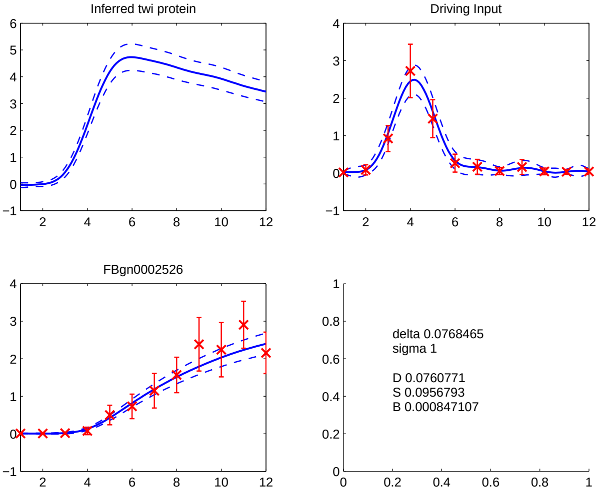


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

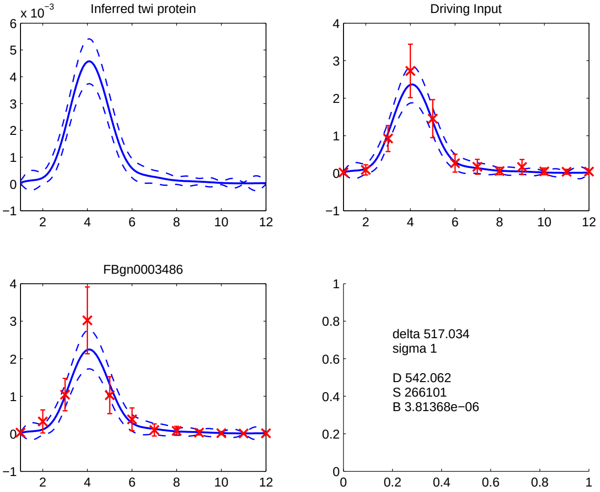


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

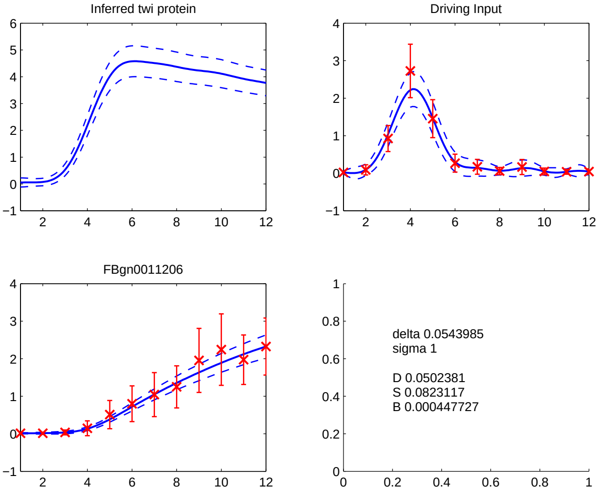


Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

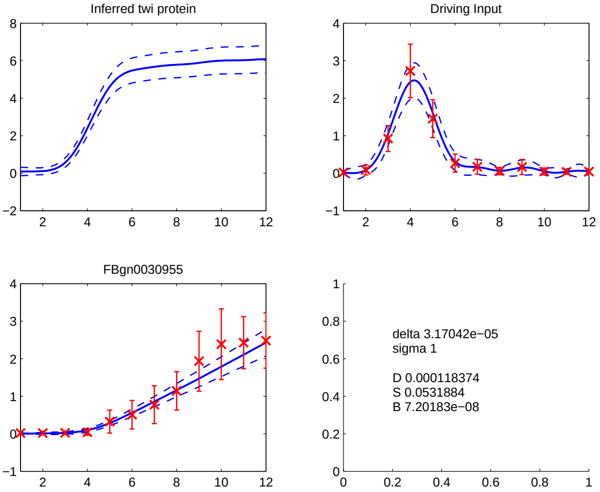


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

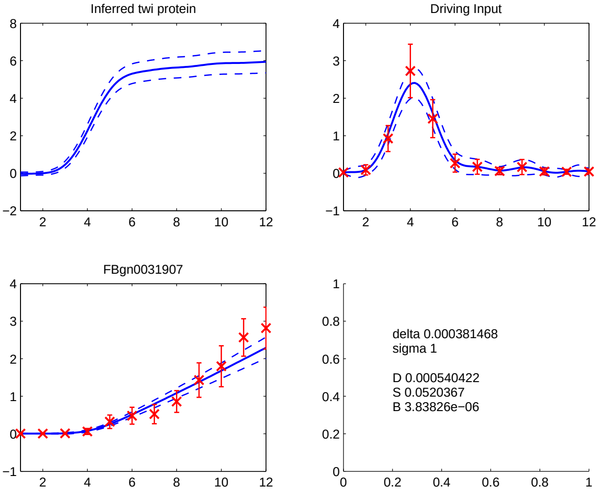


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

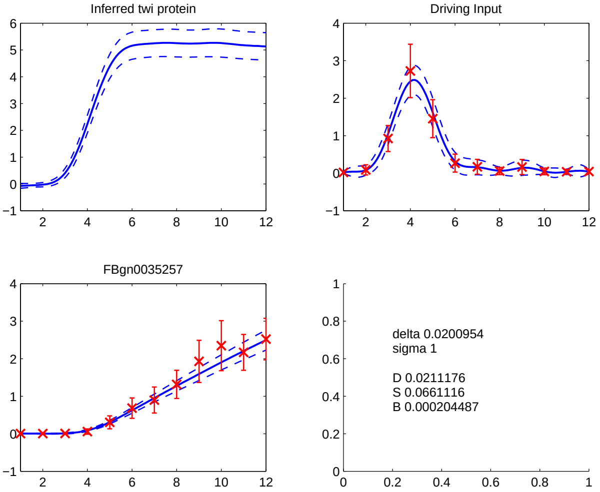


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

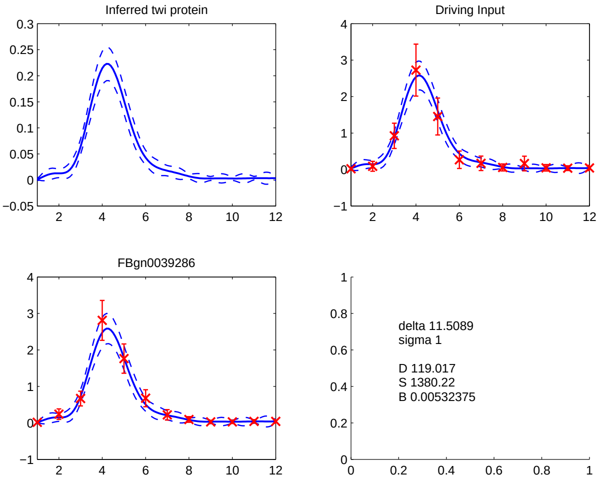
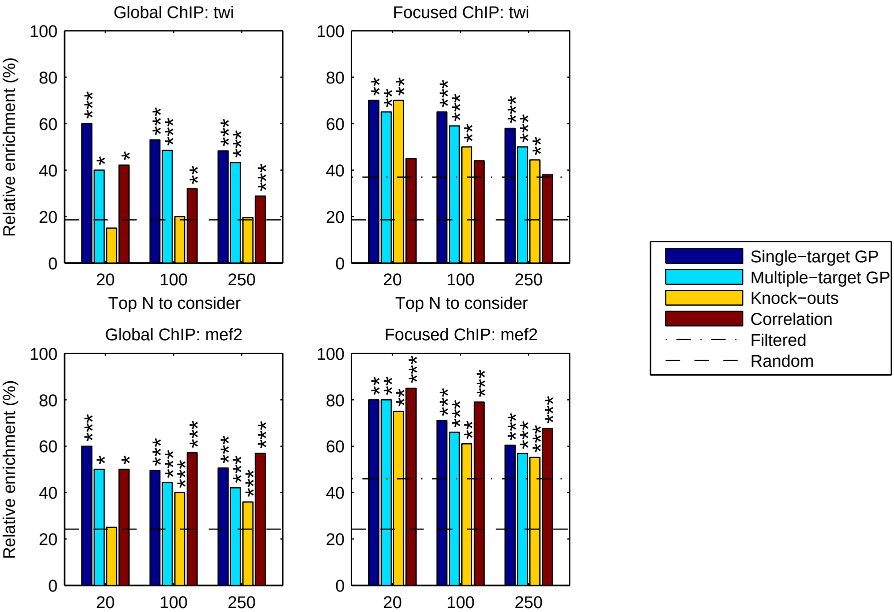


Figure: Model for flybase gene identity FBgn0039286.

Evaluation methods

- Evaluate the ranking methods by taking a number of top-ranked targets and record the number of "positives" (**?**):
  - 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



Global ChIP: twi
Focused ChIP: twi
Global ChIP: mef2
Focused ChIP: mef2
Relative enrichment (%)
Top N to consider
0
20
40
60
80
100
20
100
250
Single−target GP
Multiple−target GP
Knock−outs
Correlation
Filtered
Random

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

Multiple Transcription Factors

BMC Systems Biology

BioMed Central
The Open Access Publisher

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

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

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

Michalis K Titsias (mtitsias@well.ox.ac.uk)
Antti Honkela (antti.honkela@hiit.fi)
Neil D Lawrence (n.lawrence@sheffield.ac.uk)
Magnus Rattray (m.rattray@sheffield.ac.uk)

ISSN 1752-0509

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

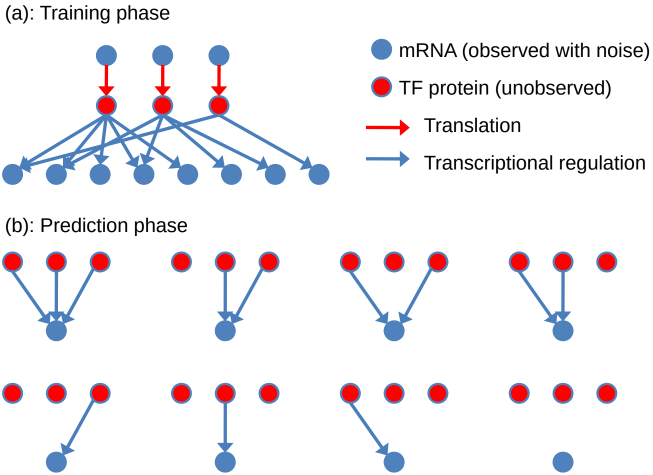
A “middle-out” approach for inferring regulatory networks

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

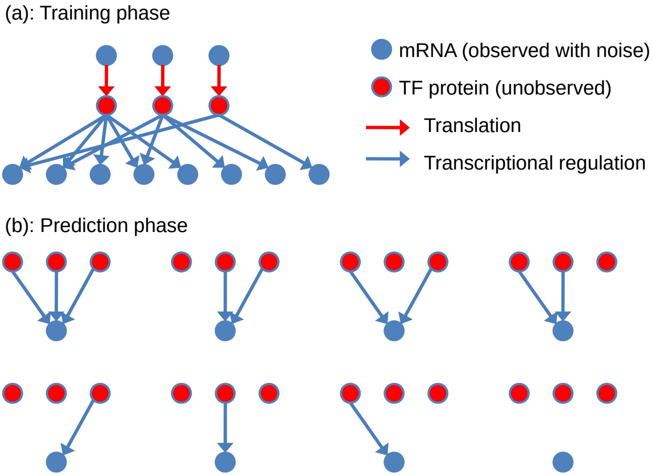
- Training stage: Parameter estimation on known network



- Scanning stage: Bayesian evidence model scoring for target inference

A “middle-out” approach for inferring regulatory networks

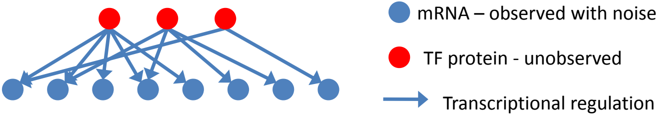
- Training stage: Parameter estimation on known network



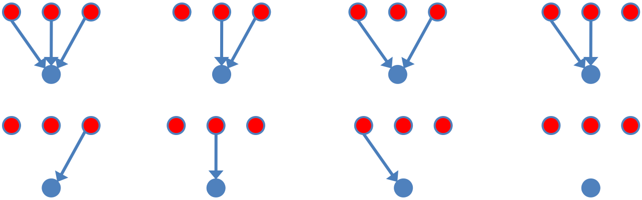
- Scanning stage: Bayesian evidence model scoring for target inference

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

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

$m_j(t)$ – 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

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

????

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

????

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

????

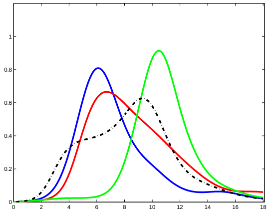
Gaussian process inference over latent functions

- 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

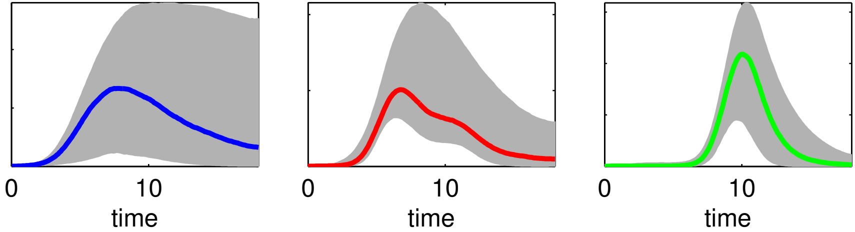
????

Artificial data: one experimental condition

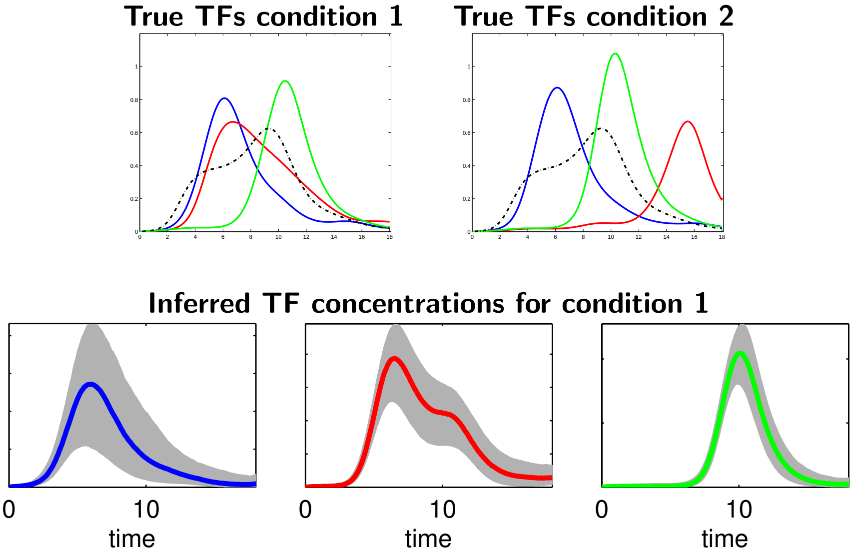
Ground Truth TFs



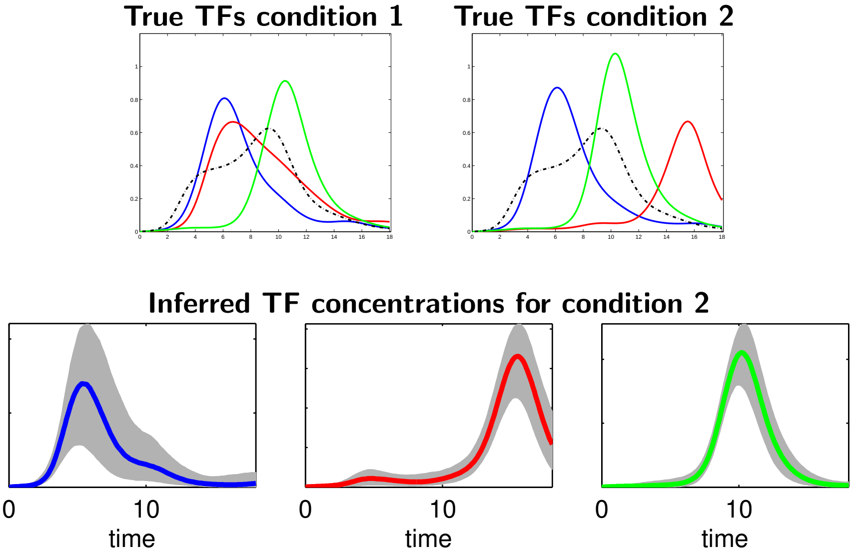
Inferred TF concentrations after training stage



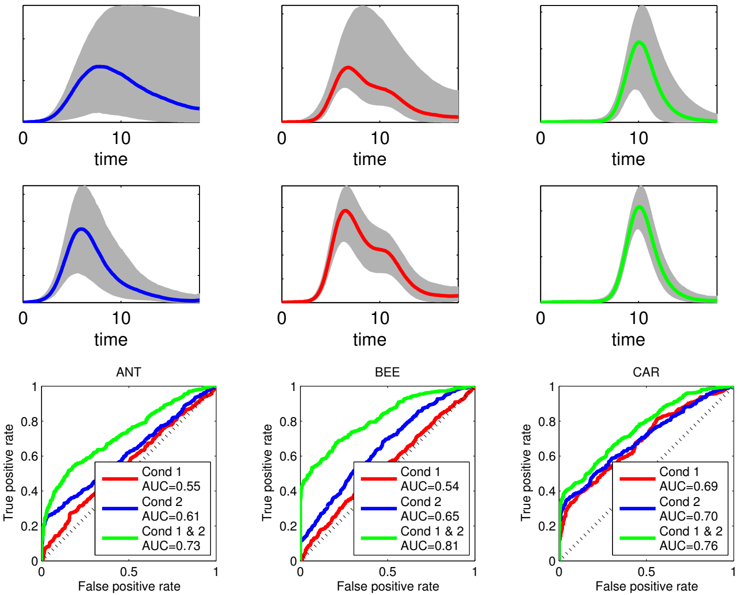
Artificial data: two experimental conditions



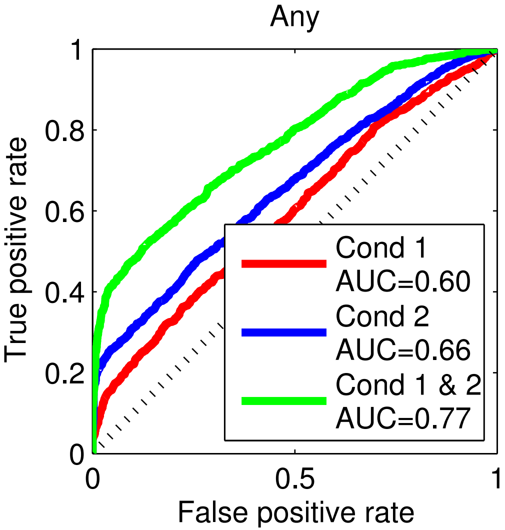
Artificial data: two experimental conditions



Artificial data: scanning performance for each TF

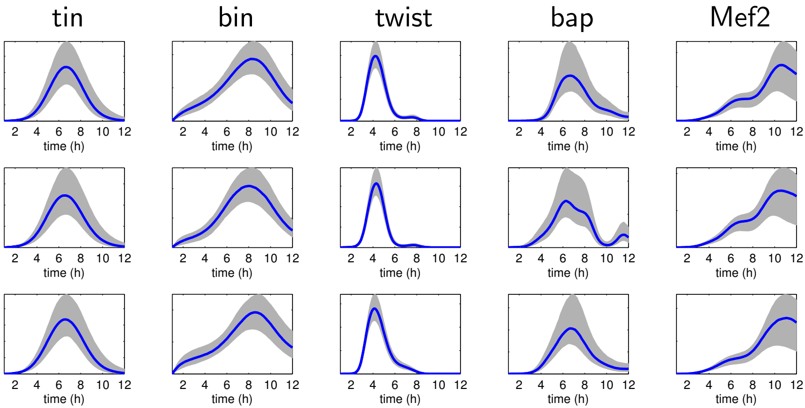


Artificial data: scanning performance for all TFs



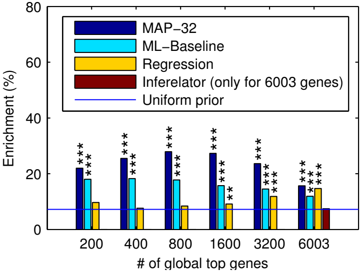
Drosophila training

- Sub-network of 96 genes targeted by 5 TFs during Drosophila mesoderm development (**?**).
- Data: wild-type times series, 3 replicates (**?**).

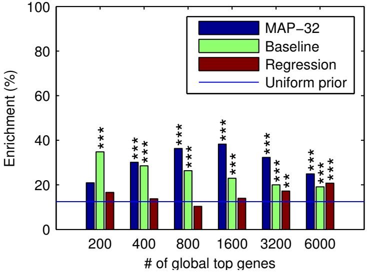


Drosophila scanning: model ranking

- 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 **?**.

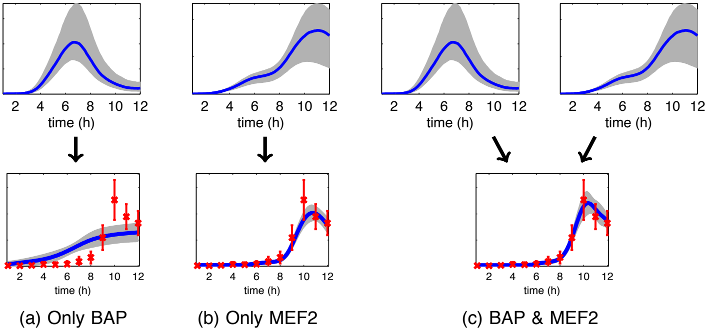


All "non-quiet" genes



All targets with in situ evidence

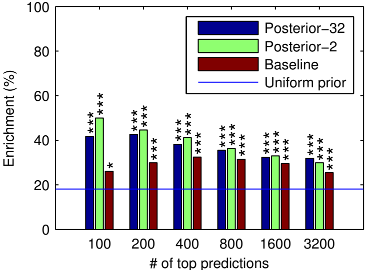
Coregulated Target Example



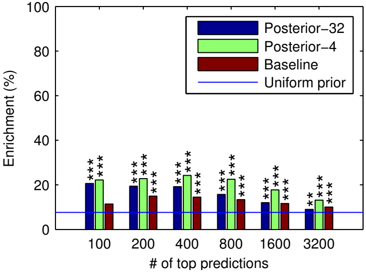
A highly ranked putative joint target of BAP amd MEF2. The candidate gene is confirmed as a joint target by independent ChIP-chip studies **?**.

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 **?**.



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 $\rightarrow$ better performance
- Robust to existence of some unknown regulating TFs
- Significant enrichment of inferred TF-target links for nearby ChIP-chip binding in drosophila development example

Summary and Conclusion

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

Summary and Conclusion

- Middle-out approach: sub-network training followed by genome-wide scanning
- Training: Bayesian inference of regulation model parameters and TF protein concentration functions
- Scanning: Bayesian model scoring for inferring TF-target link probabilities
- More informative conditions $\rightarrow$ 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

Outline

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.

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].

N. D. Lawrence, M. Girolami, M. Rattray, and G. Sanguinetti, editors. *Learning and Inference in Computational Systems Biology*. MIT Press, Cambridge, MA, 2010. [Google Books] .

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

M. K. Titsias, A. Honkela, N. D. Lawrence, and M. Rattray. Identifying targets of multiple co-regulated transcription factors from expression time-series by Bayesian model comparison. *BMC Systems Biology*, 6(53), 2012. [DOI].

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

P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin. Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, 3(12):RESEARCH0088, 2002.

R. P. Zinzen, C. Girardot, J. Gagneur, M. Braun, and E. E. M. Furlong. Combinatorial binding predicts spatio-temporal cis-regulatory activity. *Nature*, 462(7269):65–70, Nov 2009. [URL]. [DOI].