

# Latent Force Models

Neil D. Lawrence

GPRS

14th February 2014



# Outline

Motivation

Latent Force Models

ODE Model of Transcriptional Regulation

PDE Example

Efficient Approximations

# Outline

Motivation

Latent Force Models

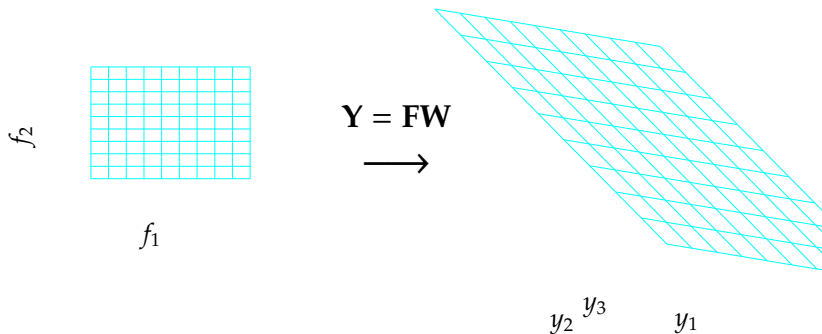
ODE Model of Transcriptional Regulation

PDE Example

Efficient Approximations

# Linear Dimensionality Reduction

- ▶ Find a lower dimensional plane embedded in a higher dimensional space.
- ▶ The plane is described by the matrix  $\mathbf{W} \in \mathbb{R}^{p \times q}$ .



# Dimensionality Reduction

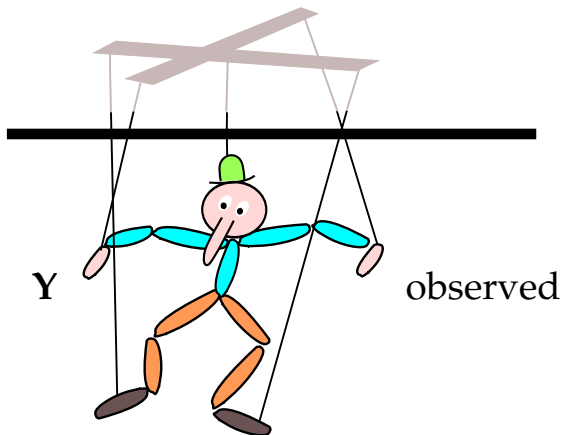
- ▶ Linear relationship between the data,  $\mathbf{Y}$ , and a reduced dimensional representation,  $\mathbf{F}$ .

$$\mathbf{Y} = \mathbf{F}\mathbf{W} + \epsilon,$$

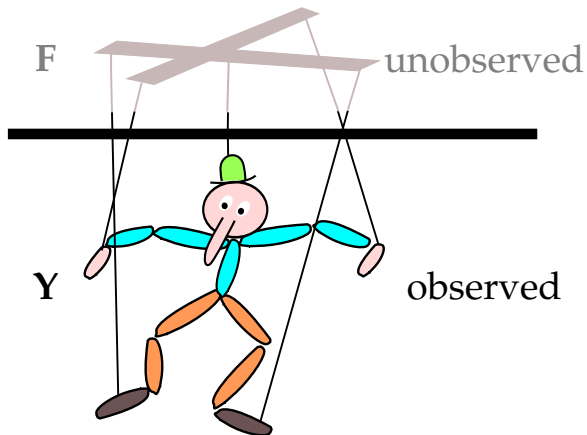
$$\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

- ▶ Problem is we don't know what  $\mathbf{F}$  should be!

# Marionette Analogy



# Marionette Analogy

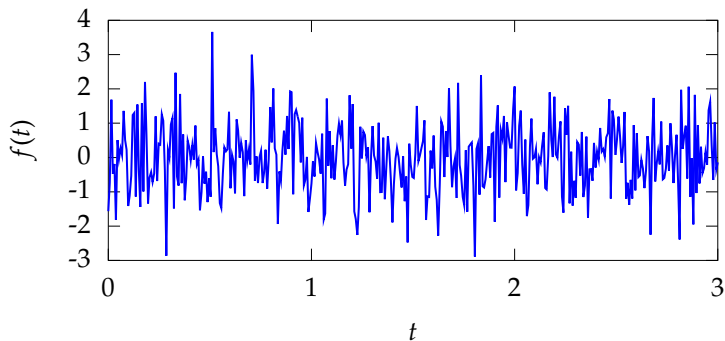


# F is a Latent Variable

- ▶ Define a *probability distribution* for  $\mathbf{F}$ .
- ▶ Marginalize out  $\mathbf{F}$  (integrate over).
- ▶ Optimize with respect to  $\mathbf{W}$ .
- ▶ For Gaussian distribution,  $\mathbf{F} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
  - ▶ and  $\Sigma = \sigma^2 \mathbf{I}$  we have probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998).
  - ▶ and  $\Sigma$  constrained to be diagonal, we have factor analysis.

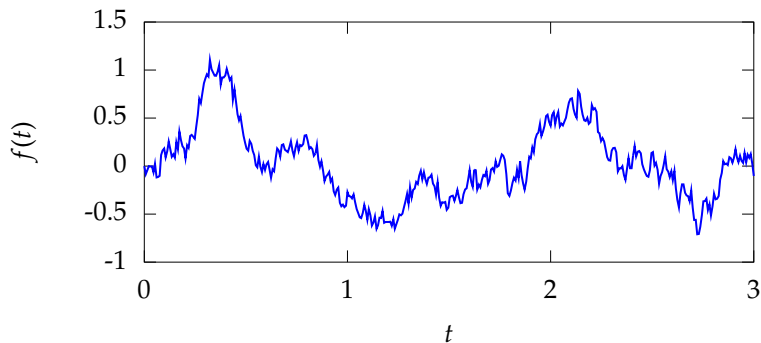


# Dimensionality Reduction: Temporal Data



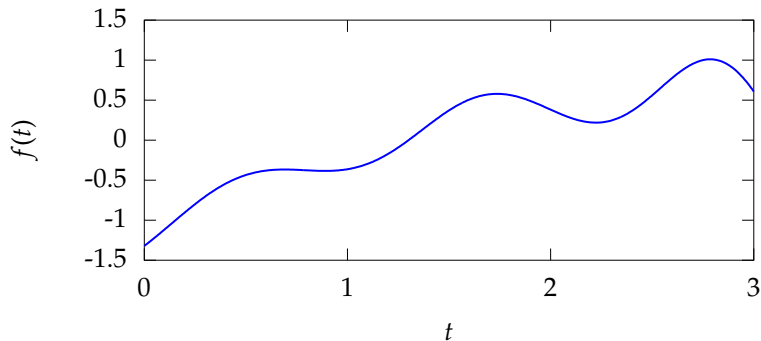
**Figure :** PCA: Pure sampling from a Gaussian does not retain temporal effects.

# Dimensionality Reduction: Temporal Data



**Figure :** Kalman filter (Rauch-Tung-Striebel smoother) is Markov-Gaussian (non smooth).

# Dimensionality Reduction: Temporal Data



**Figure :** General Gaussian processes allow for priors over *smooth* functions.

# Outline

Motivation

**Latent Force Models**

ODE Model of Transcriptional Regulation

PDE Example

Efficient Approximations

## Back to Mechanistic Models!

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

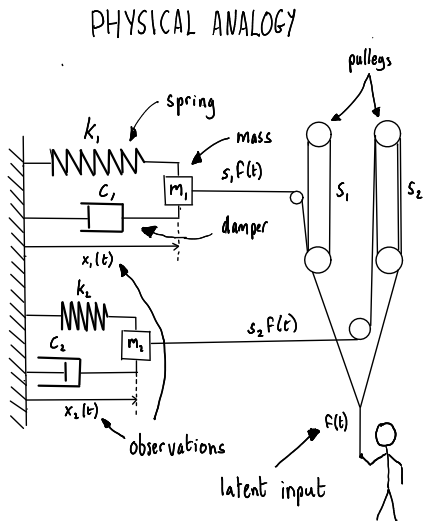
# Extend Model

- ▶ Add a damper and give the system mass.

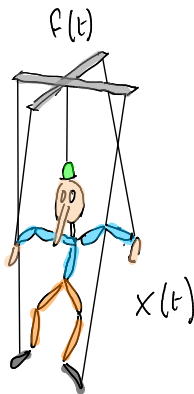
$$\mathbf{F}\mathbf{S} = \ddot{\mathbf{Y}}\mathbf{M} + \dot{\mathbf{Y}}\mathbf{C} + \mathbf{Y}\mathbf{D} + \boldsymbol{\epsilon}.$$

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

# Physical Analogy



## MARIONETTE



# Gaussian Process priors and Latent Force Models

## Driven Harmonic Oscillator

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

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

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

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

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



# Covariance for ODE Model

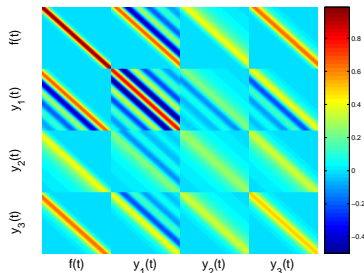
- Exponentiated Quadratic Covariance function for  $f(t)$

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

- Joint distribution for  $y_1(t)$ ,  $y_2(t)$ ,  $y_3(t)$  and  $f(t)$ .

Damping ratios:

$\zeta_1$	$\zeta_2$	$\zeta_3$
0.125	2	1



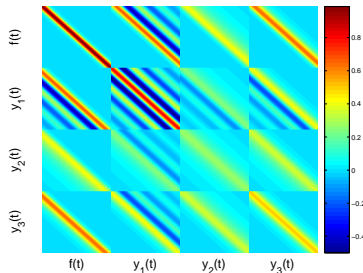
# Covariance for ODE Model

- Analogy

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

- Joint distribution for  $y_1(t)$ ,  $y_2(t)$ ,  $y_3(t)$  and  $f(t)$ .  
Damping ratios:

$\zeta_1$	$\zeta_2$	$\zeta_3$
0.125	2	1



# Covariance for ODE Model

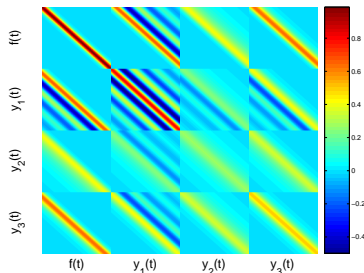
- ▶ Exponentiated Quadratic Covariance function for  $f(t)$

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

- ▶ Joint distribution for  $y_1(t)$ ,  $y_2(t)$ ,  $y_3(t)$  and  $f(t)$ .

Damping ratios:

$\zeta_1$	$\zeta_2$	$\zeta_3$
0.125	2	1



# Joint Sampling of $y(t)$ and $f(t)$

## ► 1fmSample

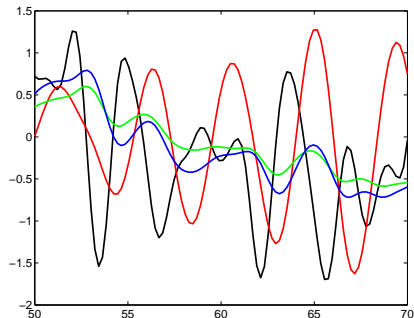


Figure : Joint samples from the ODE covariance, *black*:  $f(t)$ , *red*:  $y_1(t)$  (underdamped), *green*:  $y_2(t)$  (overdamped), and *blue*:  $y_3(t)$  (critically damped).

# Joint Sampling of $y(t)$ and $f(t)$

## ► 1fmSample

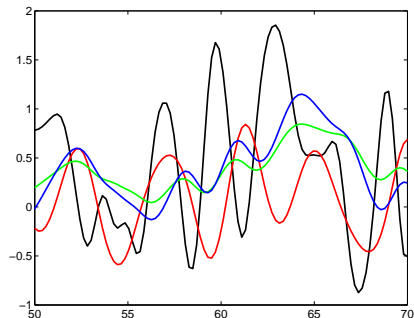


Figure : Joint samples from the ODE covariance, *black*:  $f(t)$ , *red*:  $y_1(t)$  (underdamped), *green*:  $y_2(t)$  (overdamped), and *blue*:  $y_3(t)$  (critically damped).

# Joint Sampling of $y(t)$ and $f(t)$

## ► 1fmSample

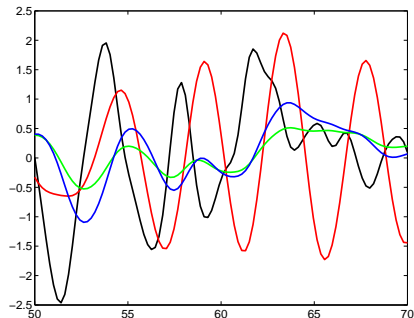


Figure : Joint samples from the ODE covariance, *black*:  $f(t)$ , *red*:  $y_1(t)$  (underdamped), *green*:  $y_2(t)$  (overdamped), and *blue*:  $y_3(t)$  (critically damped).

# Joint Sampling of $y(t)$ and $f(t)$

► 1fmSample

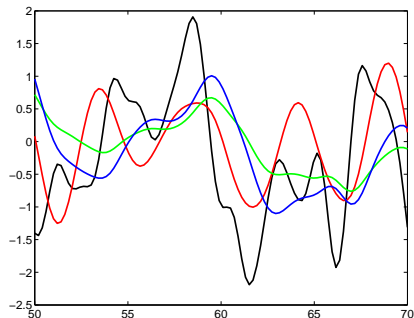


Figure : Joint samples from the ODE covariance, *black*:  $f(t)$ , *red*:  $y_1(t)$  (underdamped), *green*:  $y_2(t)$  (overdamped), and *blue*:  $y_3(t)$  (critically damped).

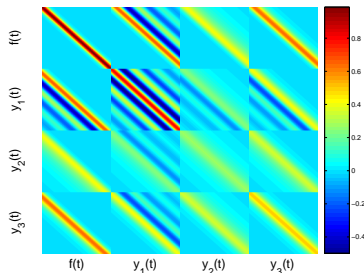
# Covariance for ODE

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

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

- ▶ Joint distribution for  $y_1(t)$ ,  $y_2(t)$ ,  $y_3(t)$  and  $f(t)$ .
- ▶ Damping ratios:

$\zeta_1$	$\zeta_2$	$\zeta_3$
0.125	2	1





## Example: Motion Capture

**Mauricio Alvarez and David Luengo (Álvarez et al., 2009, 2013)**

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

## Example: Motion Capture

**Mauricio Alvarez and David Luengo (Álvarez et al., 2009, 2013)**

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

## Example: Motion Capture

**Mauricio Alvarez and David Luengo (Álvarez et al., 2009, 2013)**

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

## Example: Motion Capture

**Mauricio Alvarez and David Luengo (Álvarez et al., 2009, 2013)**

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

# Prediction of Test Motion

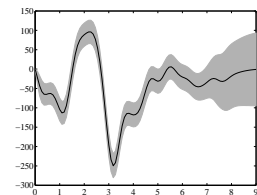
- ▶ Model left arm only.
- ▶ 3 balancing motions (18, 19, 20) from subject 49.
- ▶ 18 and 19 are similar, 20 contains more dramatic movements.
- ▶ Train on 18 and 19 and testing on 20
- ▶ Data was down-sampled by 32 (from 120 fps).
- ▶ Reconstruct motion of left arm for 20 given other movements.
- ▶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

# Mocap Results

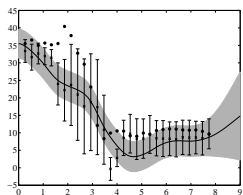
**Table :** Root mean squared (RMS) angle error for prediction of the left arm's configuration in the motion capture data. Prediction with the latent force model outperforms the prediction with regression for all apart from the radius's angle.

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

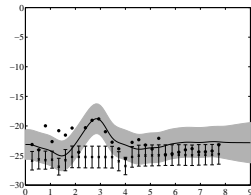
# Mocap Results II



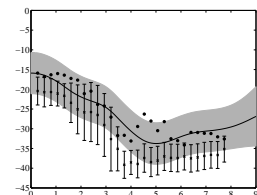
(a) Inferred Latent Force



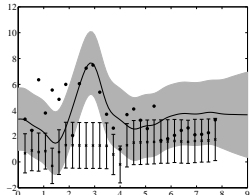
(b) Wrist



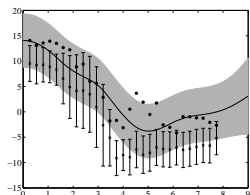
(c) Hand X Rotation



(d) Hand Z Rotation



(e) Thumb X Rotation



(f) Thumb Z Rotation

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

# Motion Capture Experiments

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



# Motion Capture Results

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

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

# Outline

Motivation

Latent Force Models

ODE Model of Transcriptional Regulation

PDE Example

Efficient Approximations

# Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

# Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

# Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

# Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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



## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

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

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

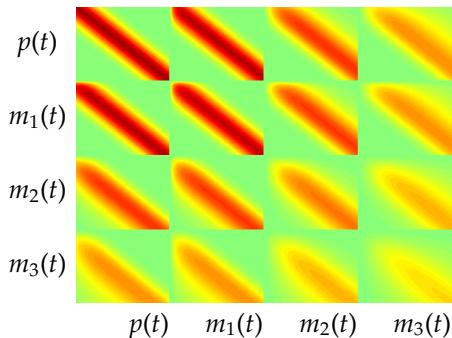
# Covariance for Transcription Model

## RBF covariance function for $p(t)$

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

- ▶ 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



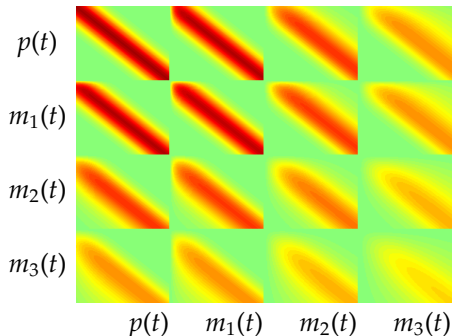
# Covariance for Transcription Model

**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



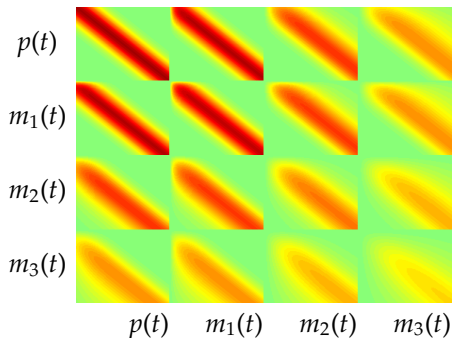
# Covariance for Transcription Model

## RBF covariance function for $p(t)$

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

- ▶ 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 $p(t)$ and $m(t)$

► `simSample`

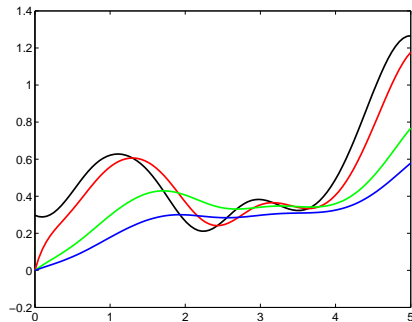


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

► `simSample`

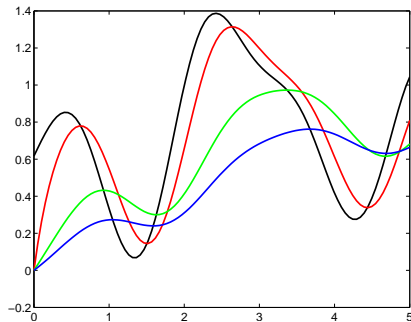


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

## ► `simSample`

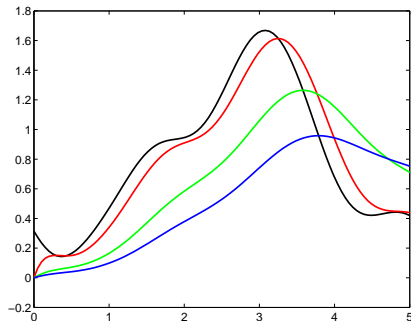


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

► `simSample`

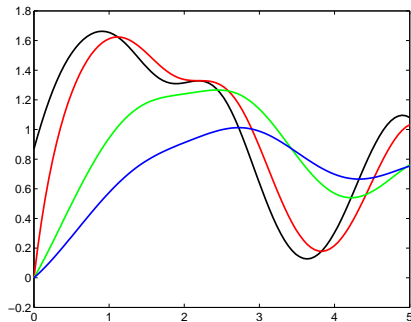
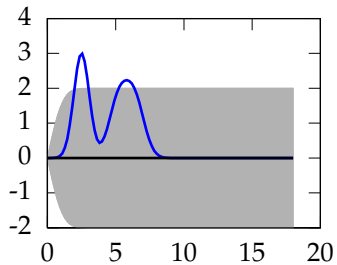
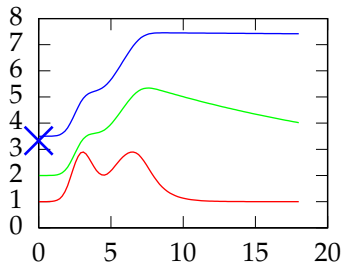
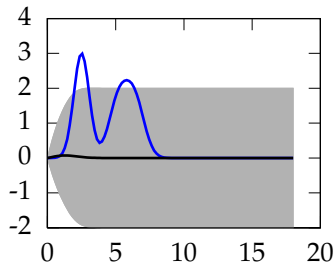
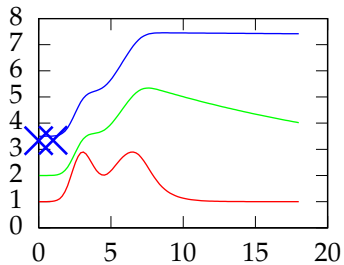
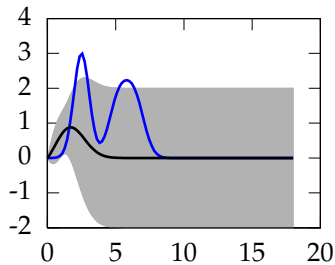
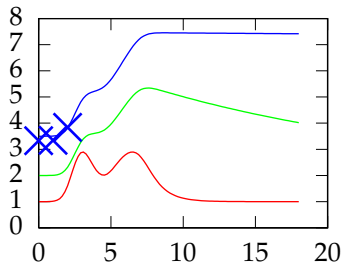


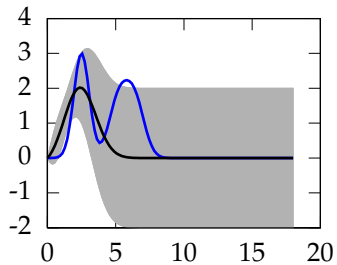
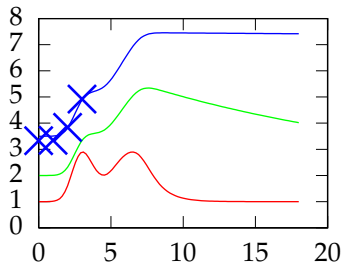
Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

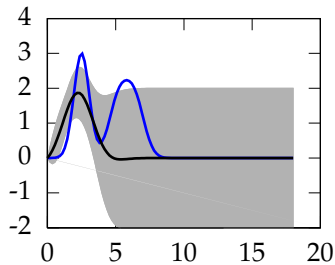
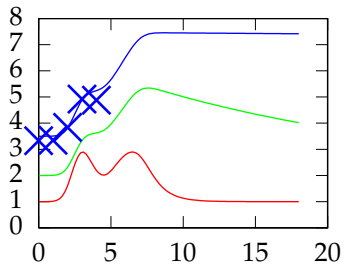


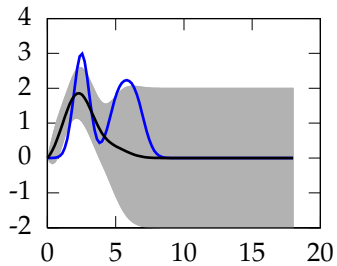
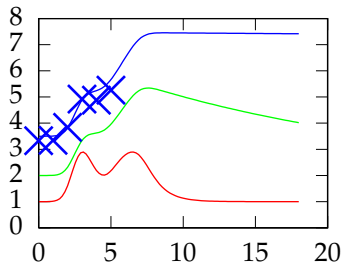


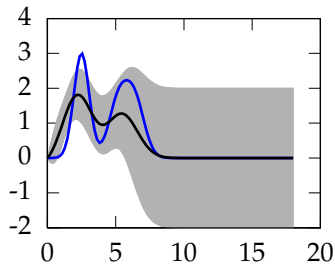
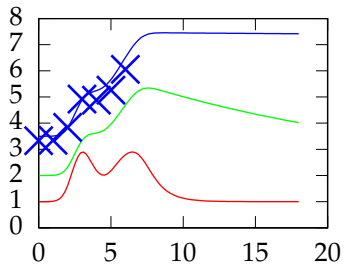


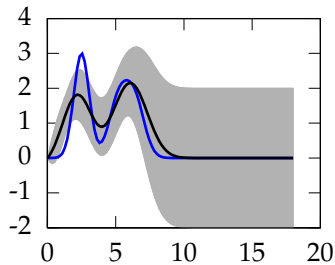
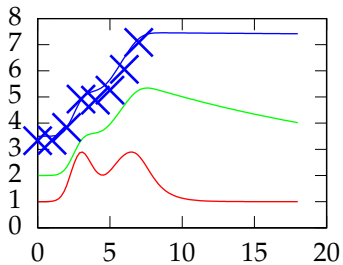




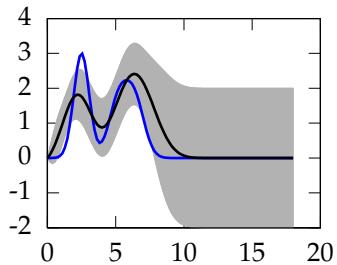
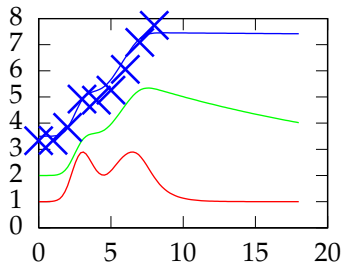


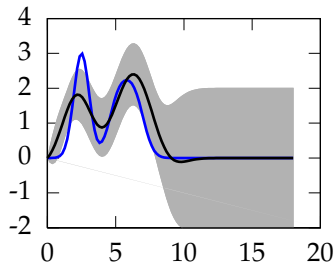
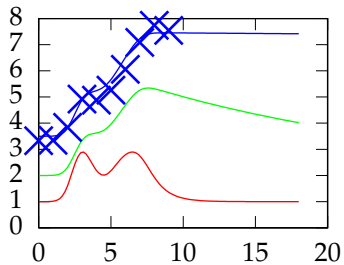


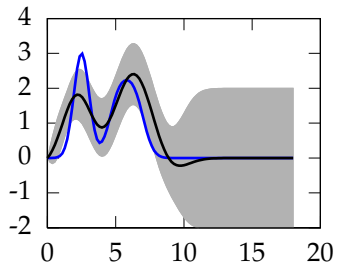
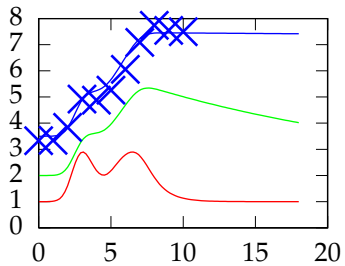


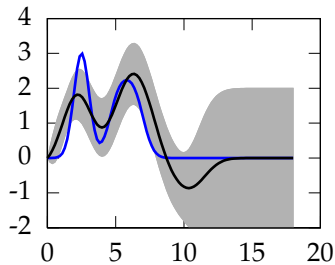
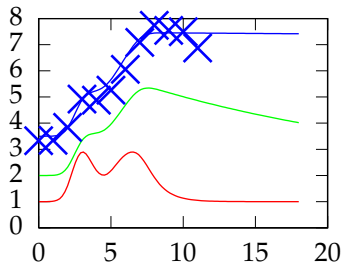


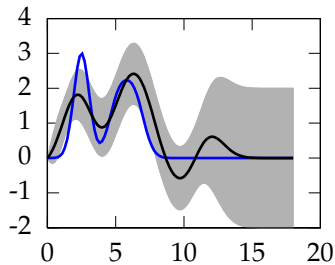
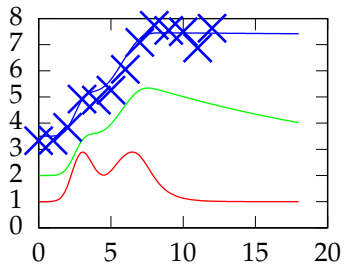


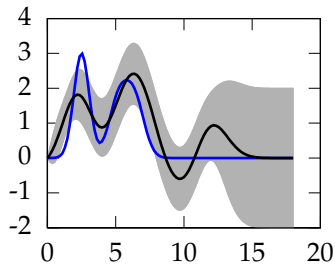
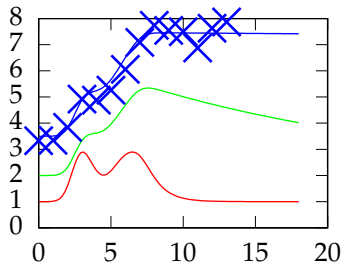


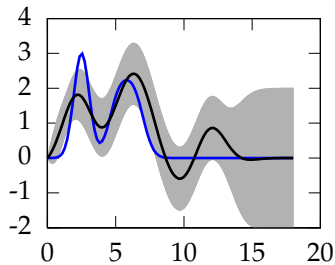
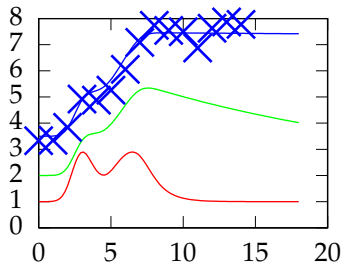


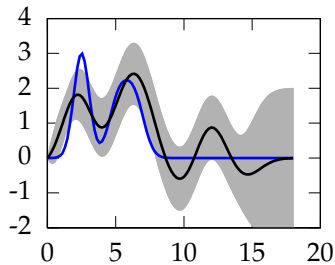
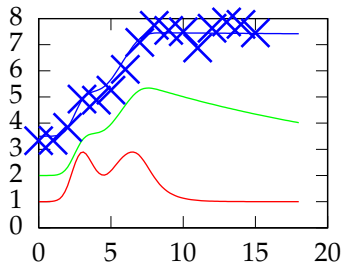




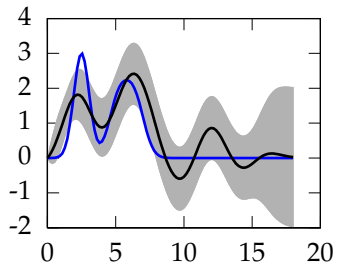
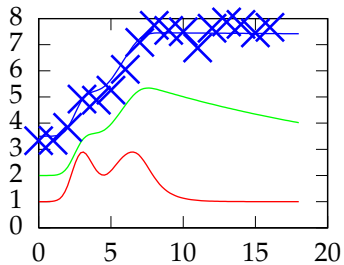


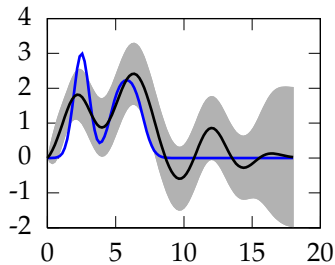
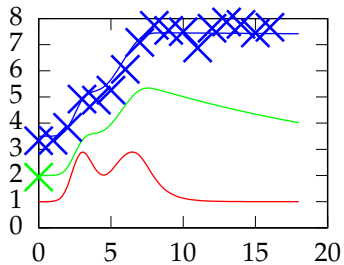


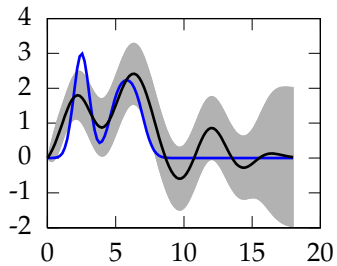
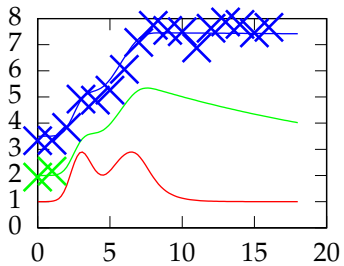


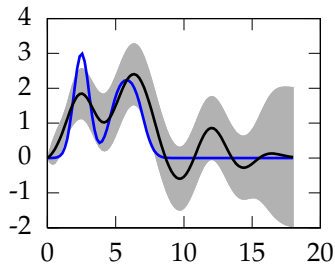
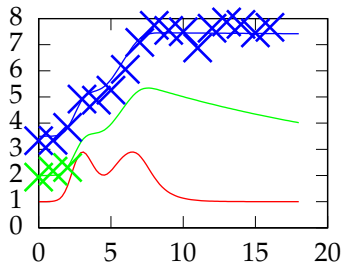


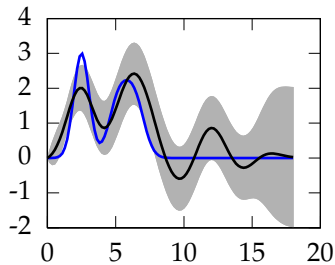
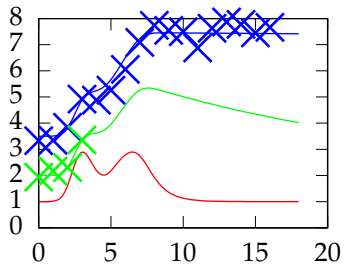


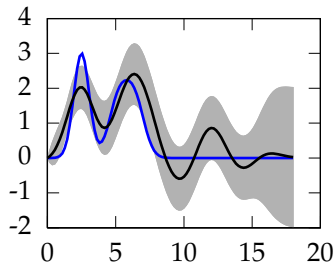
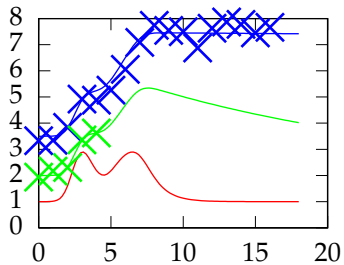


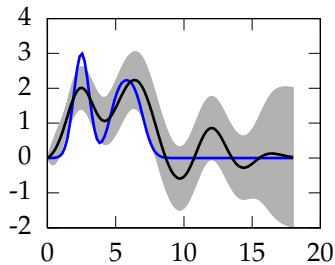
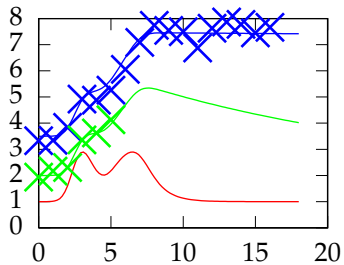


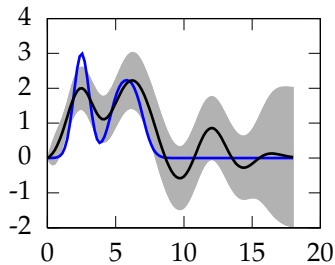
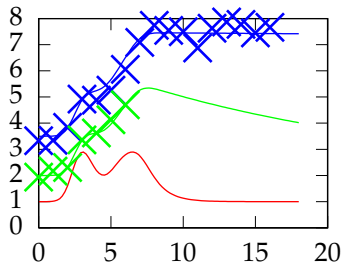




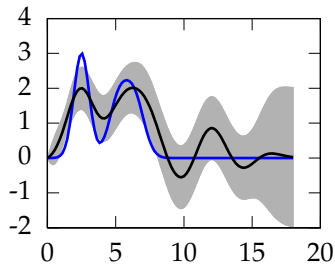
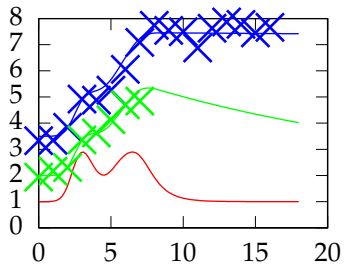


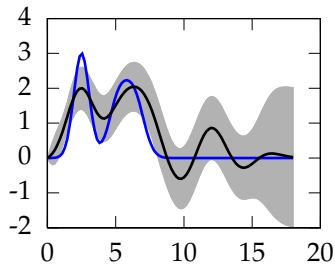
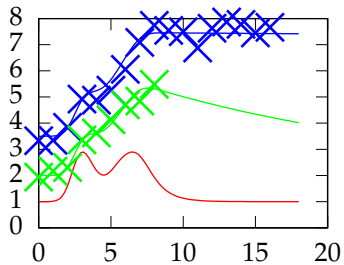


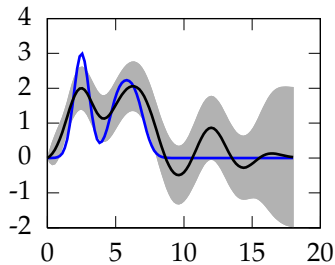
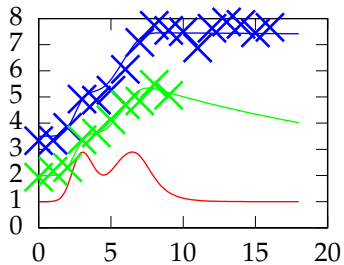


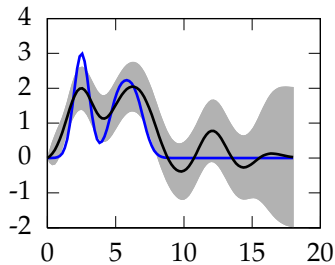
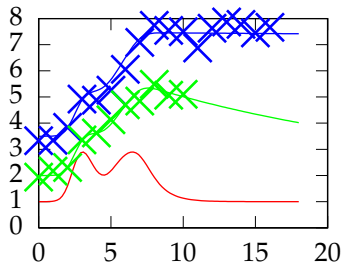


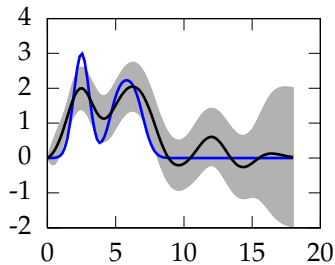
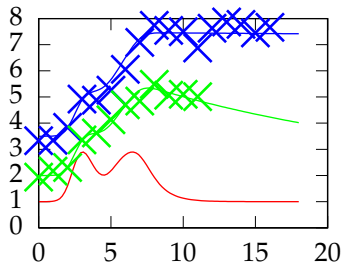


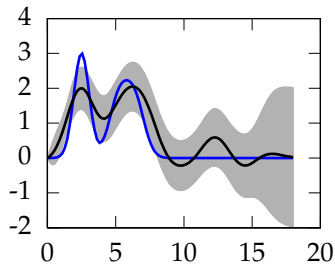
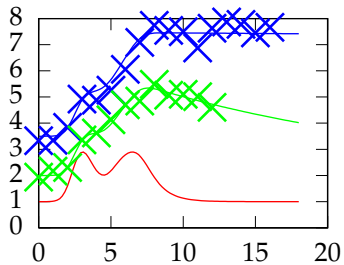


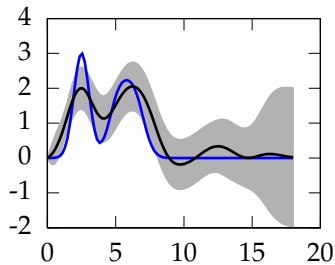
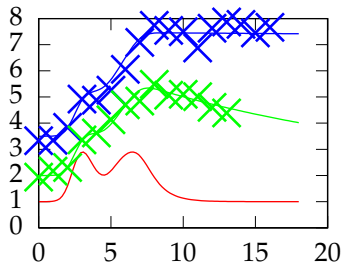


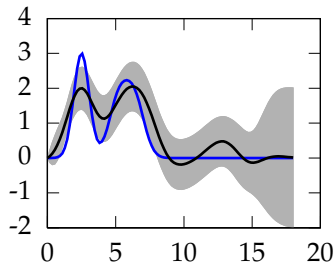
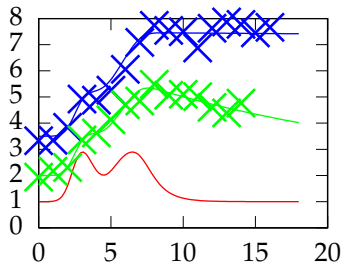




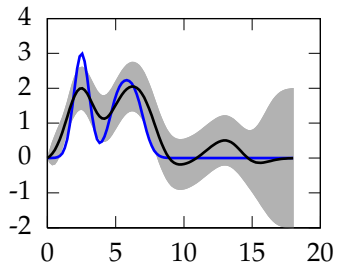
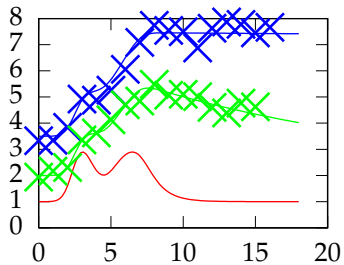


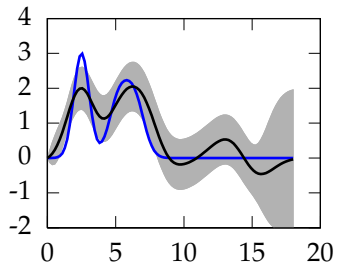
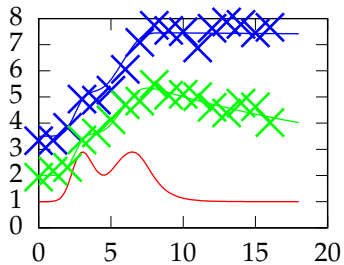


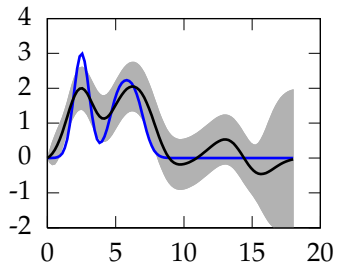
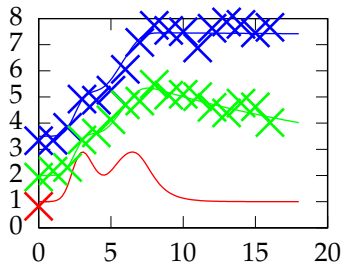


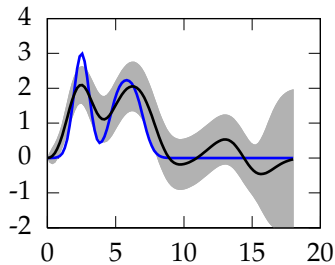
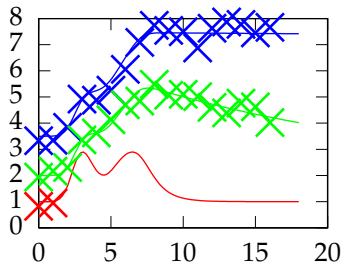


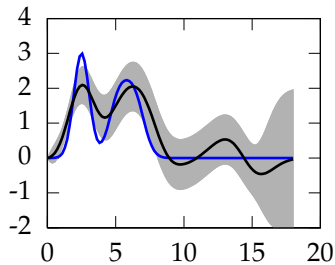
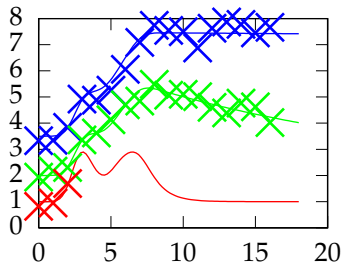


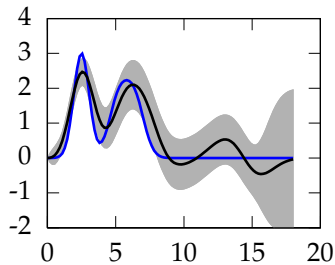
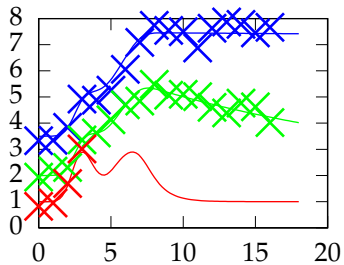


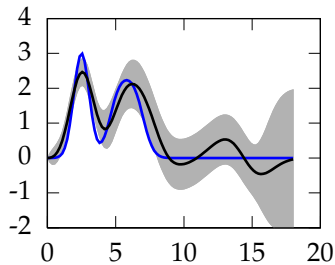
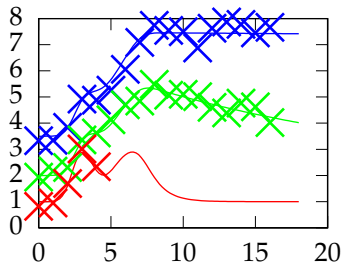


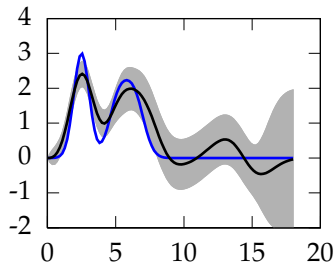
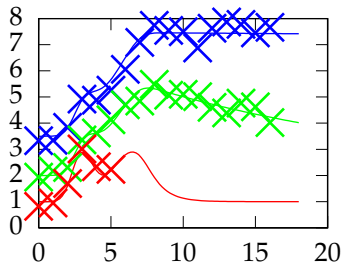




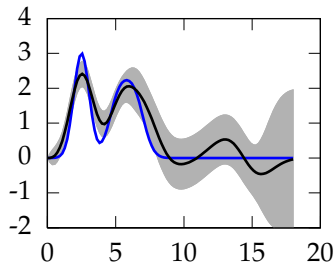
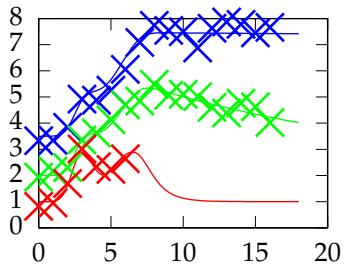


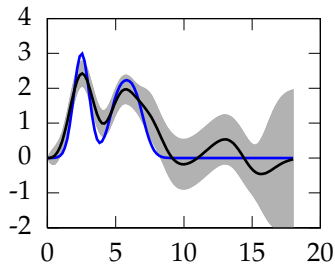
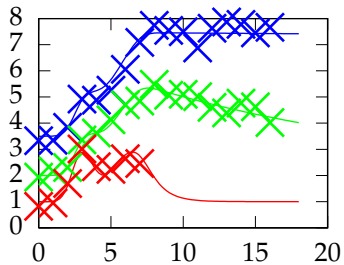


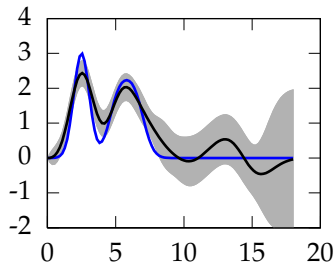
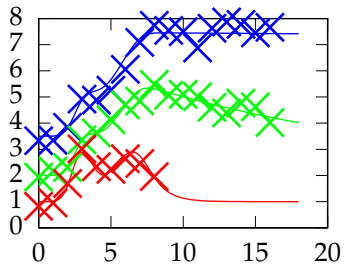


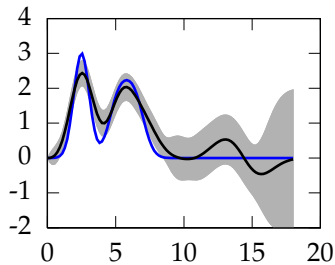
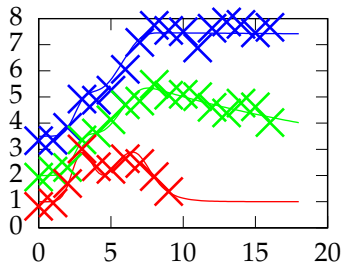


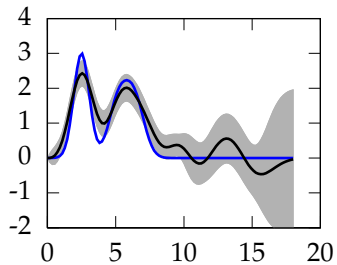
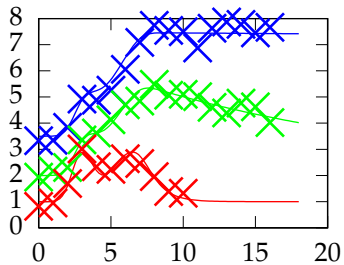


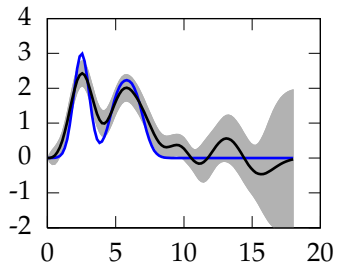
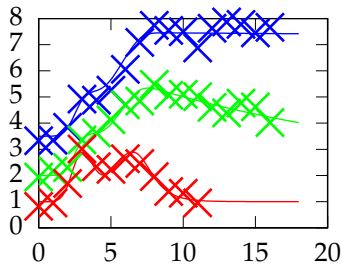


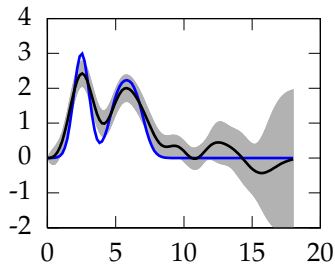
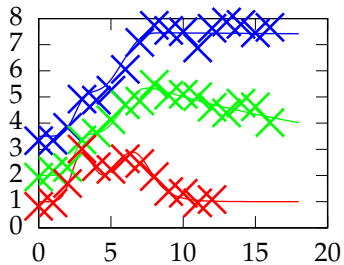


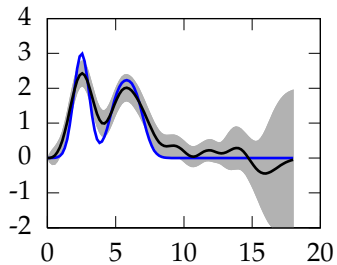
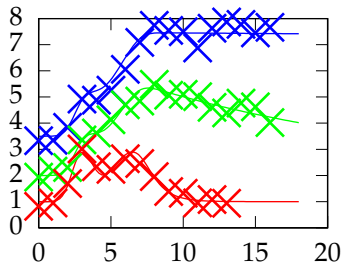




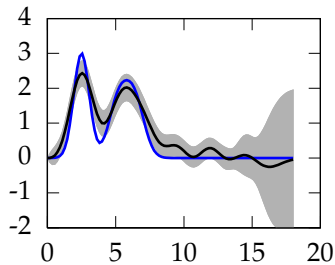
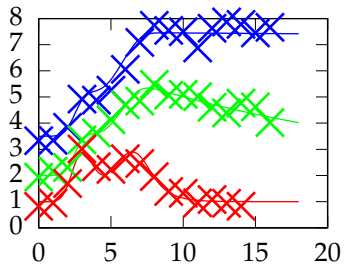


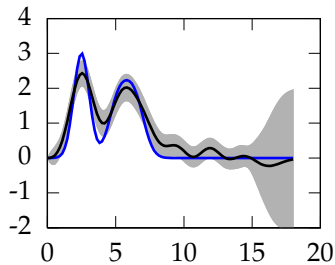
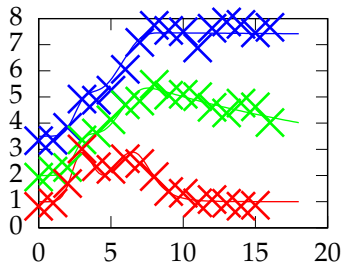


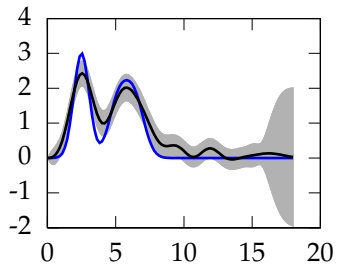
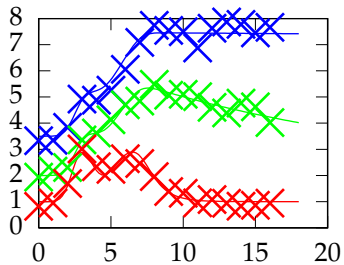












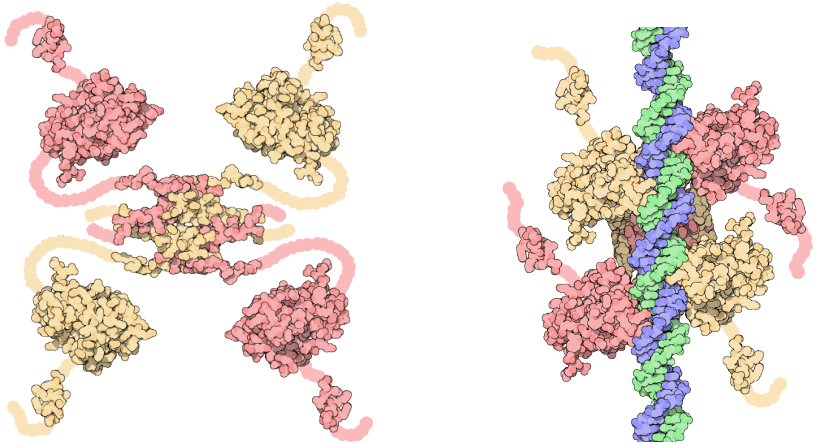
# Radiation Damage in the Cell

- ▶ Radiation can damage 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 meiosis, chromosomes have divided.
  - ▶ S: Cell is undergoing meiosis (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 feedback loop with NF- $\kappa$ B.

# 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).

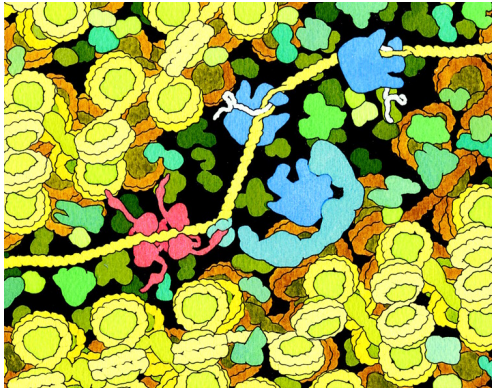


Figure : Repair of DNA damage by p53. Image from Goodsell (1999).

## Some p53 Targets

*DDB2* DNA Damage Specific DNA Binding Protein 2.  
(also governed by C/EBP-beta, E2F1, E2F3,...).

*p21* Cycline-dependent kinase inhibitor 1A  
(CDKN1A). A regulator of cell cycle progression.  
(also governed by SREBP-1a, Sp1, Sp3,... ).

*hPA26/SESN1* sestrin 1 Cell Cycle arrest.

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

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



# Modelling Assumption

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

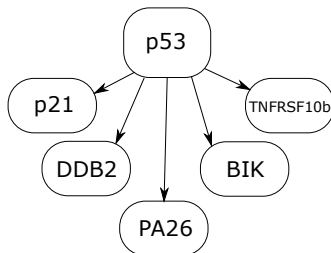


Figure : p53 SIM network motif as modelled by Barenco et al. 2006.

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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

# Ordinary Differential Equation Model

- ▶ First Order Differential Equation

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

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



---

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

Pei Gao<sup>1</sup>, Antti Honkela<sup>2</sup>, Magnus Rattray<sup>1</sup> and Neil D. Lawrence<sup>1,\*</sup>

<sup>1</sup>School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester, M13 9PL and

<sup>2</sup>Adaptive Informatics Research Centre, Helsinki University of Technology, PO Box 5400, FI-02015 TKK, Finland

---

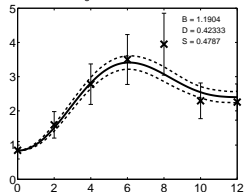
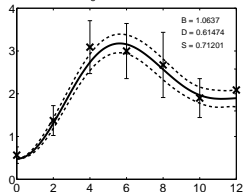
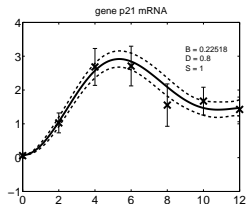
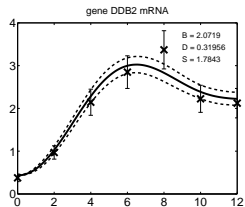
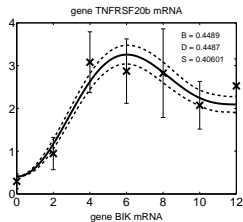
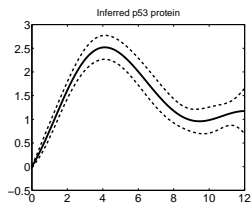
### ABSTRACT

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

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

# p53 Results with GP

(Gao et al., 2008)

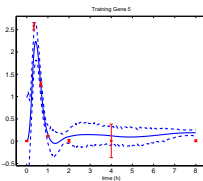
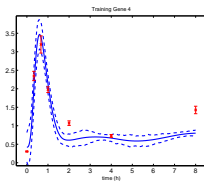
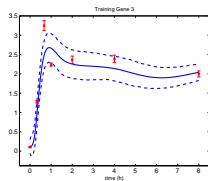
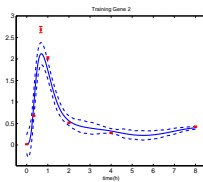
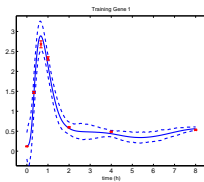
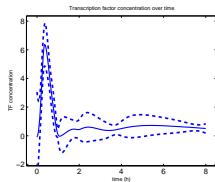


# Ranking with ERK Signalling

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

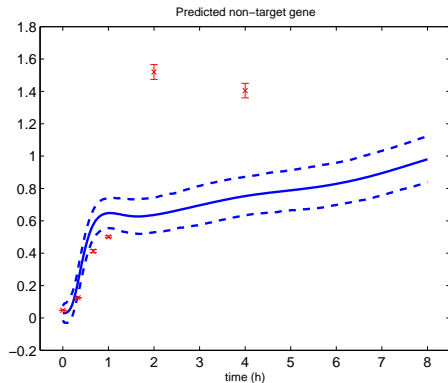
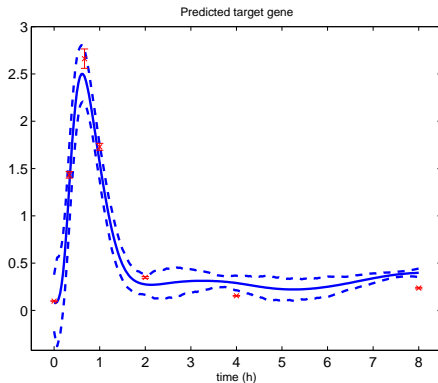
# Elk-1 (MLP covariance)

Jennifer Withers



# Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



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

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

<sup>a</sup>Department of Information and Computer Science, Aalto University School of Science and Technology, Helsinki, Finland; <sup>b</sup>Genome Biology U European Molecular Biology Laboratory, Heidelberg, Germany; and <sup>c</sup>School of Computer Science, University of Manchester, Manchester, United Kingdom

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 genes. The method is required to apply our method is wild-type time series data collected over a period where TF activity is changing. Our method allows for complementary evidence from expression**

# Cascaded Differential Equations

**(Honkela et al., 2010)**

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

# Covariance for Translation/Transcription Model

## RBF covariance function for $f(t)$

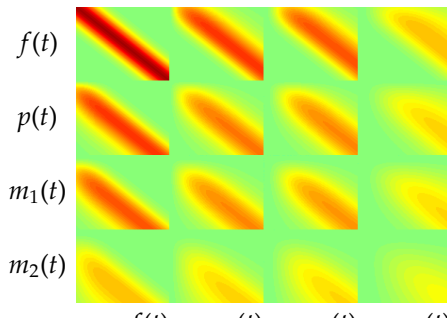
$$p(t) = \sigma \exp(-\delta t) \int_0^t f(u) \exp(\delta u) du$$

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

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

- ▶ Here:

$\delta$	$d_1$	$s_1$	$d_2$	$s_2$
1	5	5	0.5	0.5





# Twist Results

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

# Results for Twi using the Cascade model

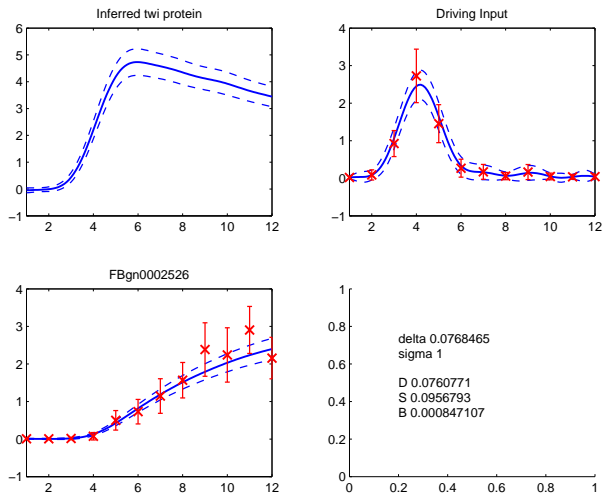


Figure : Model for flybase gene identity FBgn0002526.

# Results for Twi using the Cascade model

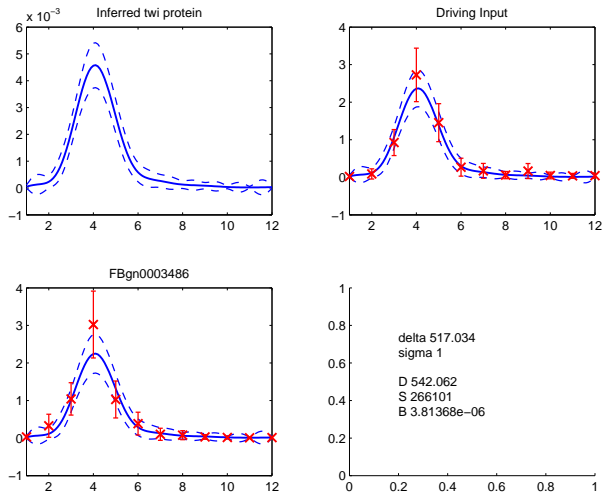


Figure : Model for flybase gene identity FBgn0003486.

# Results for Twi using the Cascade model

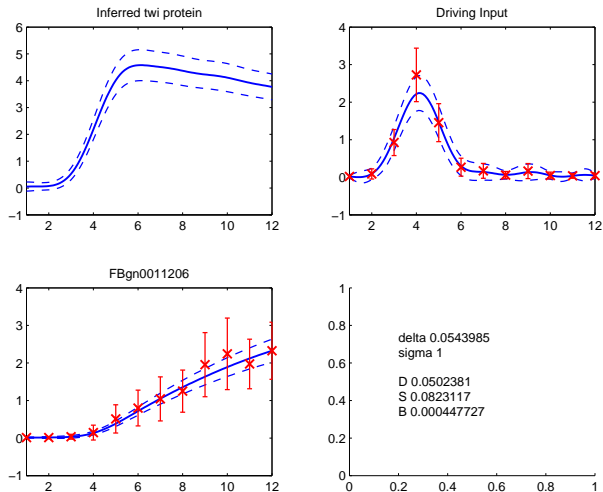


Figure : Model for flybase gene identity FBgn0011206.

# Results for Twi using the Cascade model

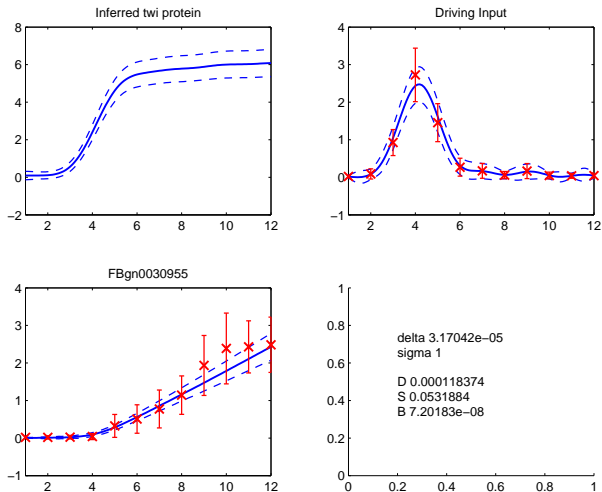


Figure : Model for flybase gene identity FBgn00309055.

# Results for Twi using the Cascade model

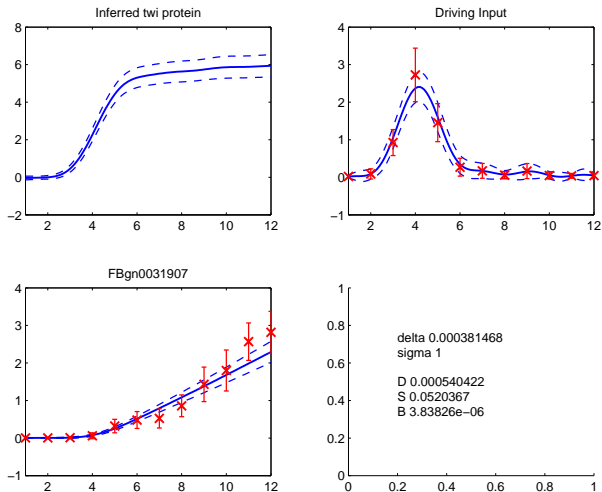


Figure : Model for flybase gene identity FBgn0031907.

# Results for Twi using the Cascade model

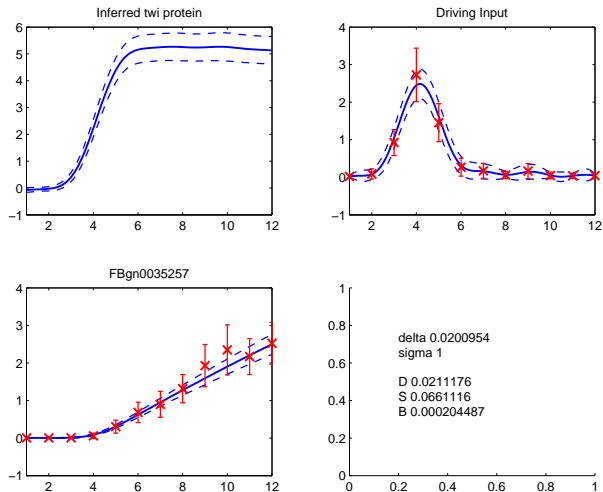


Figure : Model for flybase gene identity FBgn0035257.

# Results for Twi using the Cascade model

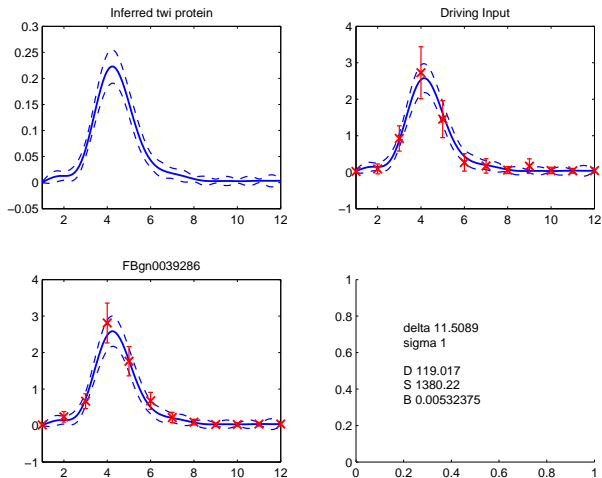


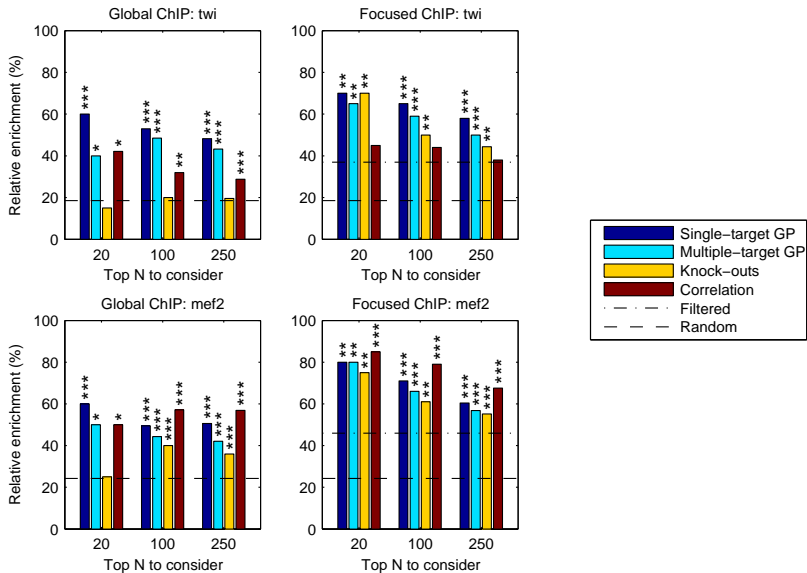
Figure : Model for flybase gene identity FBgn0039286.



# Evaluation methods

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

# Results



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

# Summary

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

# Outline

Motivation

Latent Force Models

ODE Model of Transcriptional Regulation

**PDE Example**

Efficient Approximations

# Partial Differential Equations and Latent Forces

**Mauricio Alvarez**

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

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

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

# Solution to the PDE

**Mauricio Alvarez**

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

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

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

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

with  $T_q = \kappa_q t$ .

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

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

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

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

# Prediction of Metal Concentrations

**Mauricio Alvarez**

- ▶ Replicate experiments in (Goovaerts, 1997, pp. 248,249):
  - ▶ *Primary variable* (Cd, Cu, Pb, Co) predicted in conjunction with *secondary variables* (Ni and Zn for Cd; Pb, Ni, and Zn for Cu; Cu, Ni, and Zn for Pb; Ni and Zn for Co).<sup>2</sup>
- ▶ Condition on the secondary variables to improve prediction for primary variables.
- ▶ Compare results for the diffusion kernel with independent GPs and “ordinary co-kriging” (Goovaerts, 1997, pp. 248,249).



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

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

# Outline

Motivation

Latent Force Models

ODE Model of Transcriptional Regulation

PDE Example

Efficient Approximations

# Convolutions and Computational Complexity

**Mauricio Alvarez**

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

$$x_i(t) = \int f(u) k_i(u-t) du + h_i(t)$$

$$x_i(t) = \int_0^t f(u) g_i(u) du + h_i(t)$$

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

# Independence Assumption

**Mauricio Alvarez**

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

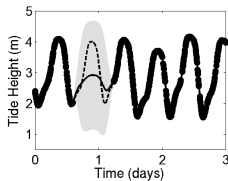
# Tide Sensor Network

**Mauricio Alvarez**

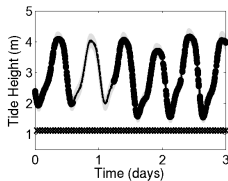
- ▶ Network of tide height sensors in the solent — tide heights are correlated.
- ▶ Data kindly provided by Alex Rogers (see Osborne et al., 2008).
- ▶  $d = 3$  and  $N = 1000$  of the 4320 for the training set.
- ▶ Simulate sensor failure by knocking out one sensor for a given time.
- ▶ For the other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

# Tide Height Results

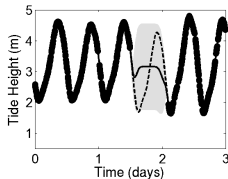
Mauricio Alvarez



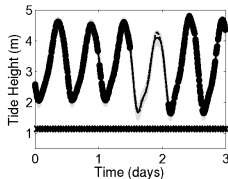
(a) Bramblemet Independent



(b) Bramblemet PITC



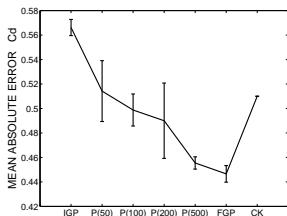
(c) Cambermet Independent



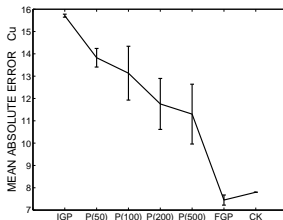
(d) Cambermet PITC

**Mauricio Alvarez**

- ▶ Jura dataset — concentrations of several heavy metals (Atteia et al., 1994).
- ▶ Prediction 259 data, validation 100 data points.
- ▶ Predict *primary variables* (cadmium and copper) at prediction locations in conjunction with some *secondary variables* (nickel and zinc for cadmium; lead, nickel and zinc for copper) (Goovaerts, 1997, p. 248,249).



(a) Cadmium



(b) Copper

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



# Discussion and Future Work

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

# Acknowledgements

**Investigators** Neil Lawrence and Magnus Rattray

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

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

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

Antti Honkela visits to Manchester funded by PASCAL I & II EU Networks of excellence.

# References I

- M. A. Álvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 57–64, Cambridge, MA, 2009. MIT Press. [PDF].
- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Latent force models. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 9–16, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5. [PDF].
- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. Technical report, University of Sheffield, [PDF].
- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, 2013. [PDF].
- M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Y. W. Teh and D. M. Titterton, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010a. JMLR W&CP 9. [PDF].
- M. A. Álvarez, J. Peters, B. Schölkopf, and N. D. Lawrence. Switched latent force models for movement segmentation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 55–63, Cambridge, MA, 2010b. MIT Press.
- O. Atteia, J.-P. Dubois, and R. Webster. Geostatistical analysis of soil contamination in the Swiss Jura. *Environ Pollut*, 86(3):315–327, 1994.
- M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25, 2006.
- P. Boyle and M. Frean. Dependent Gaussian processes. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 217–224, Cambridge, MA, 2005. MIT Press.
- 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].

# References II

- 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.
- P. Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997. [Google Books].
- D. M. Higdon. Space and space-time modelling using process convolutions. In C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, editors, *Quantitative methods for current environmental issues*, pages 37–56. Springer-Verlag, 2002.
- A. Honkela, C. Girardot, E. H. Gustafson, Y.-H. Liu, E. E. M. Furlong, N. D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proc. Natl. Acad. Sci. USA*, 107(17): 7793–7798, Apr 2010. [DOI].
- M. A. Osborne, A. Rogers, S. D. Ramchurn, S. J. Roberts, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.
- A. D. Polyanin. *Handbook of Linear Partial Differential Equations for Engineers and Scientists*. Chapman & Hall/CRC, 1 edition, 2002.
- J. Quiñero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
- S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.
- E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.
- M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6 (3):611–622, 1999. [PDF]. [DOI].
- 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].