Latent Force Models and Multiple Output Gaussian Processes

Neil D. Lawrence

work with Magnus Rattray, Mauricio Alvarez, Pei Gao, Antti Honkela, David Luengo, Guido Sanguinetti, Michalis Titsias, Jennifer Withers

SLIM Meeting

23rd July 2009

Outline

Outline

Dimensionality Reduction I

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

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

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

- Integrate out $\mathbf{F}$, optimize with respect to $\mathbf{W}$.
- For temporal data and a particular Gaussian prior in the latent space: Kalman filter/smoother.
- More generally consider a Gaussian process (GP) prior,

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

Dimensionality Reduction II

- Given the covariance functions for $\{f_i(t)\}$ the implied covariance functions for $\{x_i(t)\}$ — semi-parametric latent factor model (Teh et al., 2005). Linear Models of Coregionalization.
- Kalman filter/smoother approach has been preferred
  - linear computational complexity in N.
  - Advances in sparse approximations have made the general GP framework practical. (Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005; Titsias, 2009).

Mechanical Analogy

- 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 d springs to the forces.,
  - Interpret system as the force balance equation, $\mathbf{XD} = \mathbf{FS} + \boldsymbol{\epsilon}$.
  - Forces act, e.g. through levers — a matrix of sensitivities, $\mathbf{S} \in \Re^{q \times d}$.
  - Diagonal matrix of spring constants, $\mathbf{D} \in \Re^{d \times d}$.
  - Original System: $\mathbf{W} = \mathbf{SD}^{-1}$.

Extend Model

- Add a damper and give the system mass.

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

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

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

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

  where, m_k is the kth diagonal element from $\mathbf{M}$ and similarly for c_k and d_k. s_{ik} is the i, kth element of $\mathbf{S}$.
- Model the latent forces as q independent, GPs with RBF covariances

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

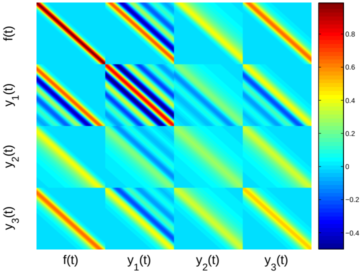
Covariance for ODE Model

- RBF Kernel function for $f(t)$

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



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

- `demLfmSample`

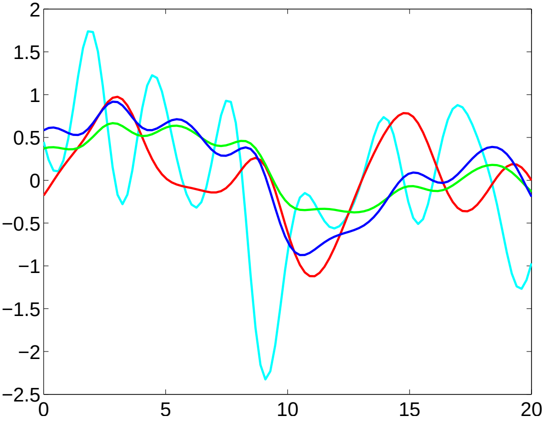


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $x_1(t)$(underdamped) and *green*: $x_2(t)$ (overdamped) and *blue*: $x_3(t)$ (critically damped).

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

- `demLfmSample`

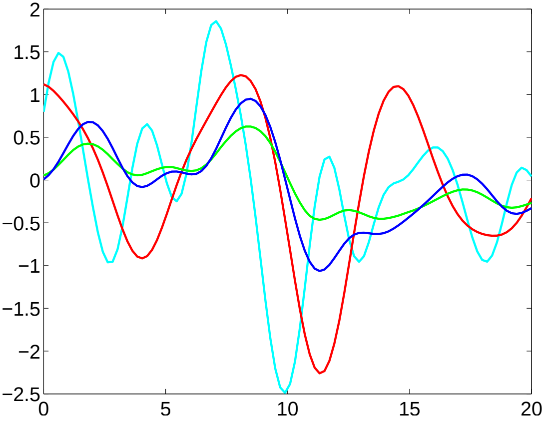


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $x_1(t)$(underdamped) and *green*: $x_2(t)$ (overdamped) and *blue*: $x_3(t)$ (critically damped).

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

- `demLfmSample`

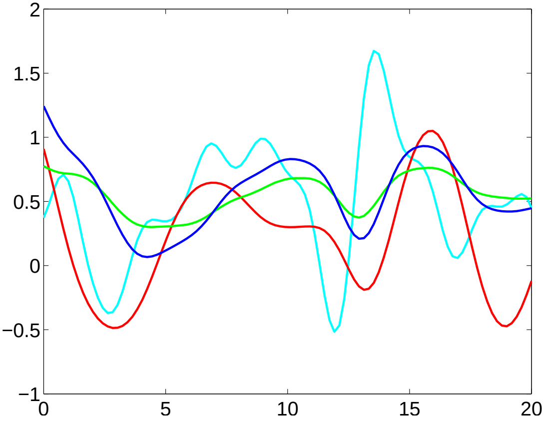


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $x_1(t)$(underdamped) and *green*: $x_2(t)$ (overdamped) and *blue*: $x_3(t)$ (critically damped).

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

- `demLfmSample`

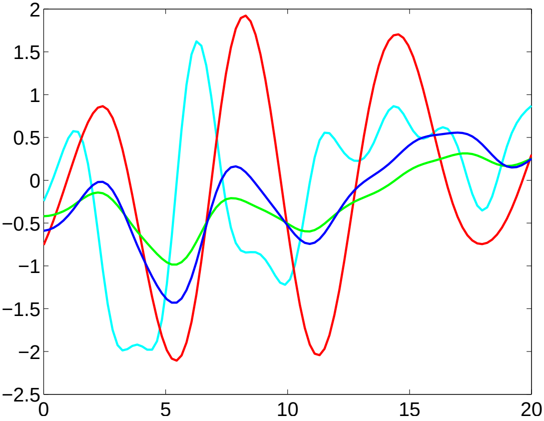


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $x_1(t)$(underdamped) and *green*: $x_2(t)$ (overdamped) and *blue*: $x_3(t)$ (critically damped).

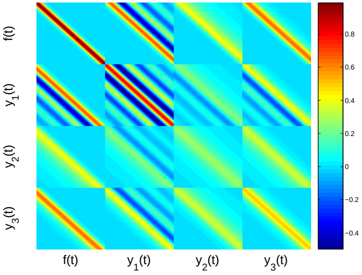
Covariance for ODE

- RBF Kernel function for $f(t)$

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



Example: Motion Capture

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

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

Example: Motion Capture

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

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

Example: Motion Capture

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

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

Example: Motion Capture

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

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

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

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

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

Example: Transcriptional Regulation

r

- First Order Differential Equation

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

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

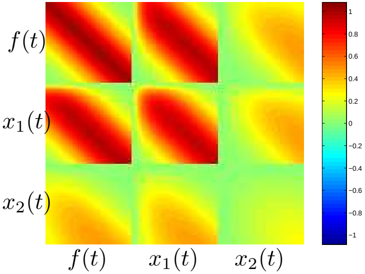
[labels=skipGPProperties]Covariance for Transcription Model

RBF covariance function for $f(t)$

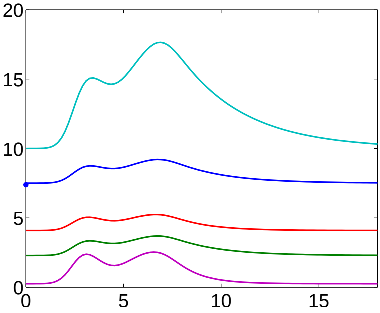
$$x_i(t) = \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u) \, \mathrm{d}u.$$

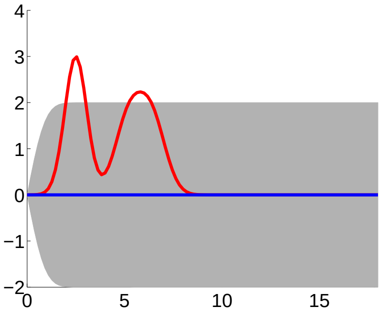
- Joint distribution for $x_1(t)$, $x_2(t)$ and $f(t)$.
  - Here:

D_1	S_1	D_2	S_2
5	5	0.5	0.5

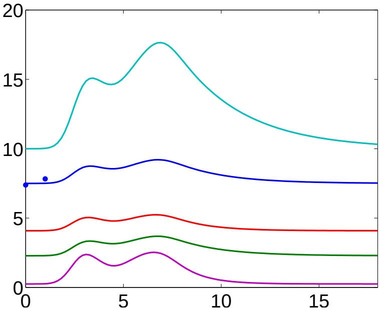


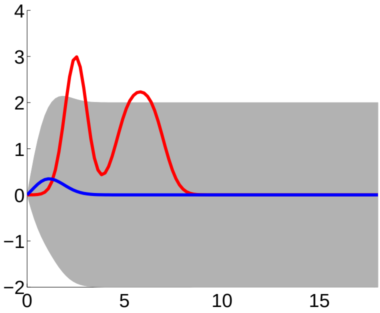
Artificial Example: Inferring $f(t)$



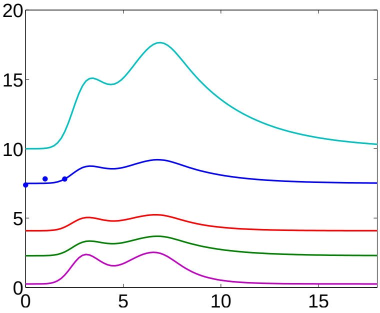


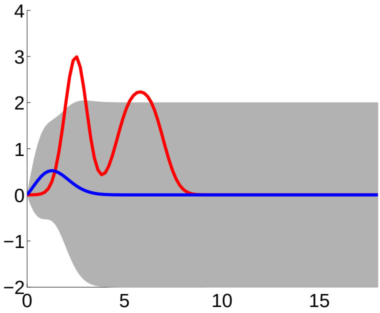
Artificial Example: Inferring $f(t)$



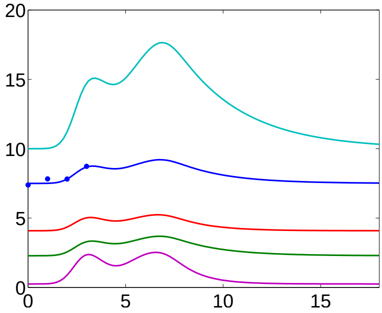


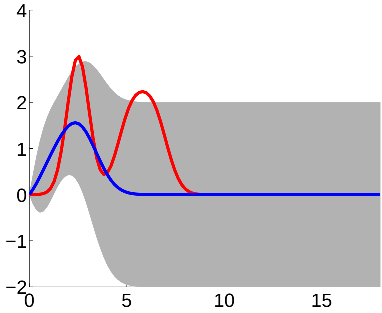
Artificial Example: Inferring $f(t)$



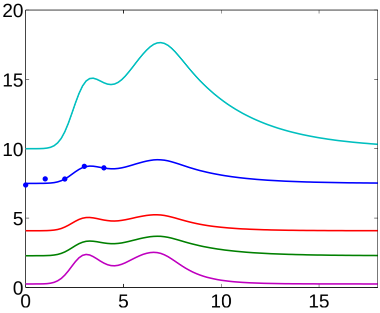


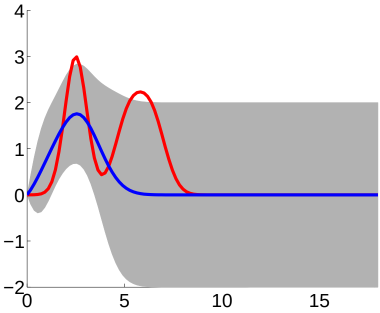
Artificial Example: Inferring $f(t)$



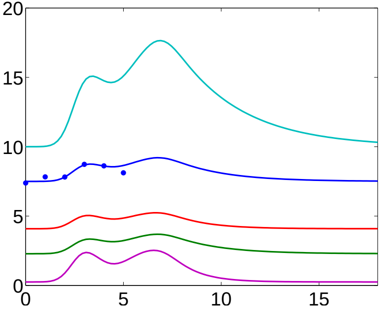


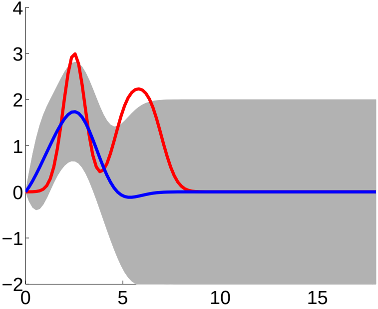
Artificial Example: Inferring $f(t)$



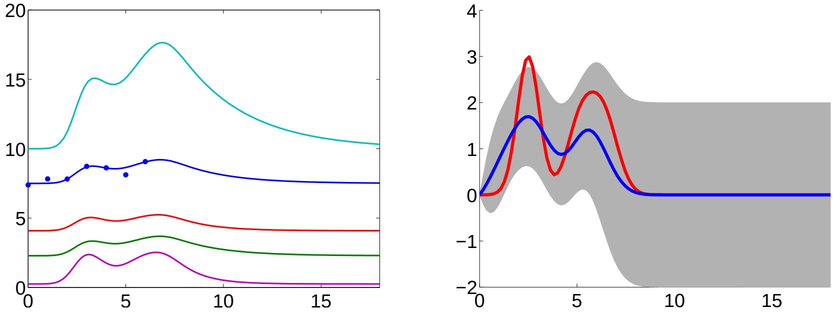


Artificial Example: Inferring $f(t)$

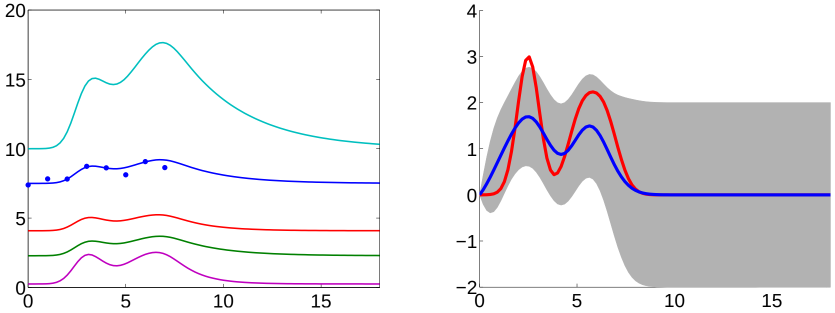




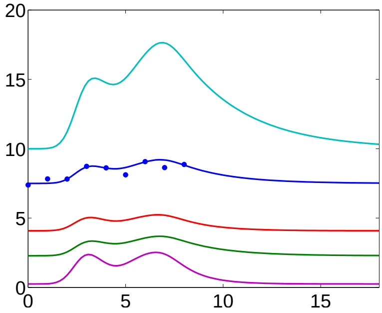
Artificial Example: Inferring $f(t)$

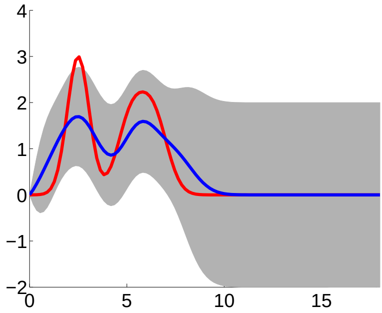


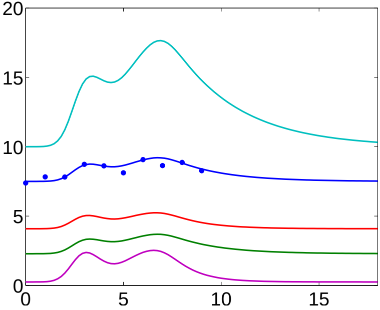
Artificial Example: Inferring $f(t)$

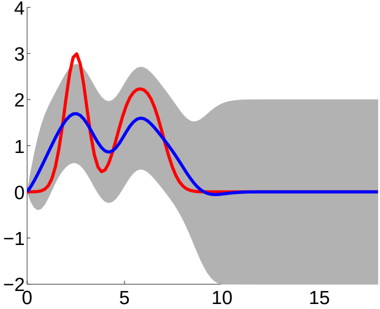


Artificial Example: Inferring $f(t)$

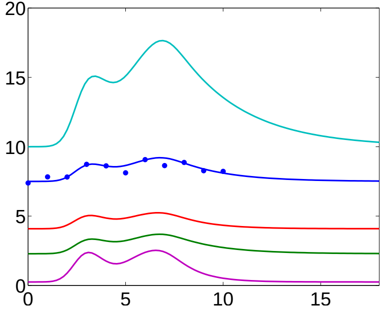


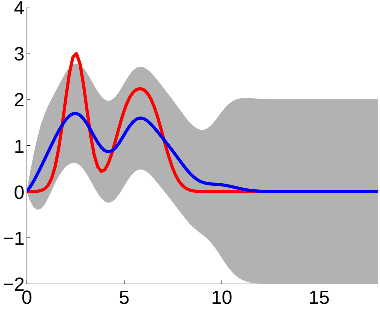



20
15
10
5
0
0
5
10
15

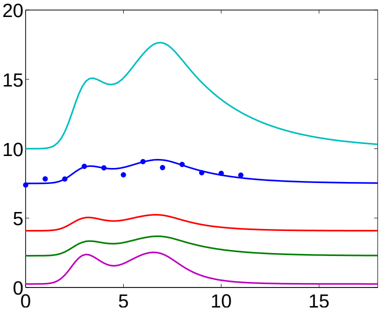

4
3
2
1
0
−1
−2
0
5
10
15

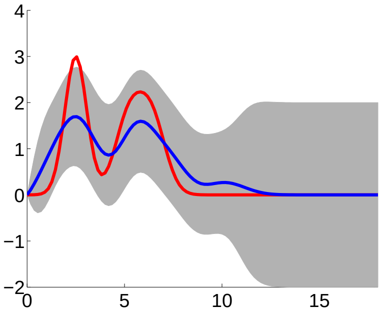
Artificial Example: Inferring $f(t)$



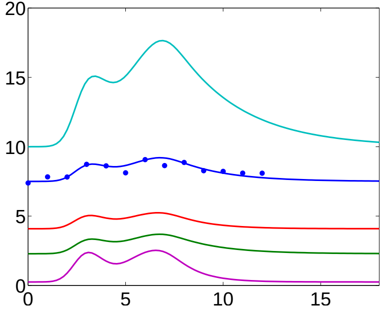


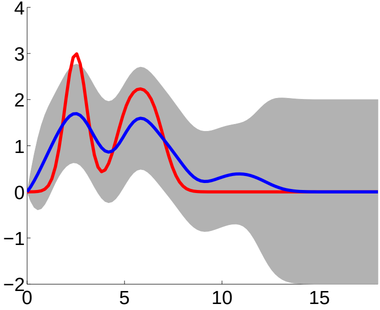
Artificial Example: Inferring $f(t)$

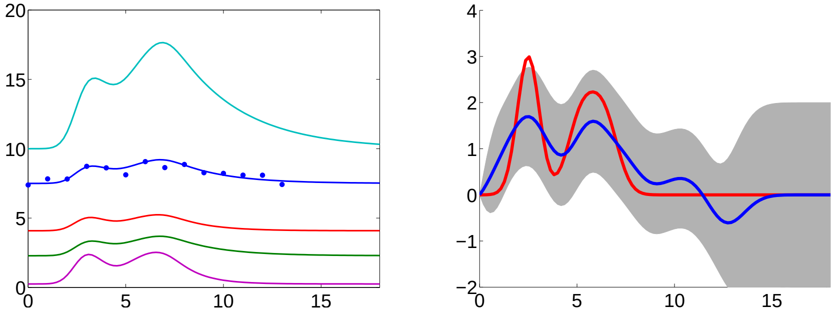




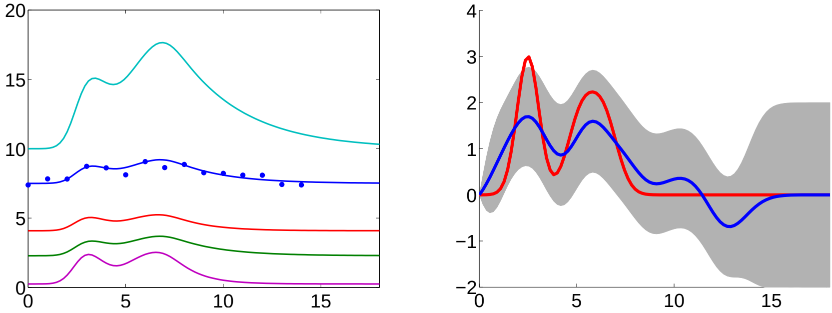
Artificial Example: Inferring $f(t)$



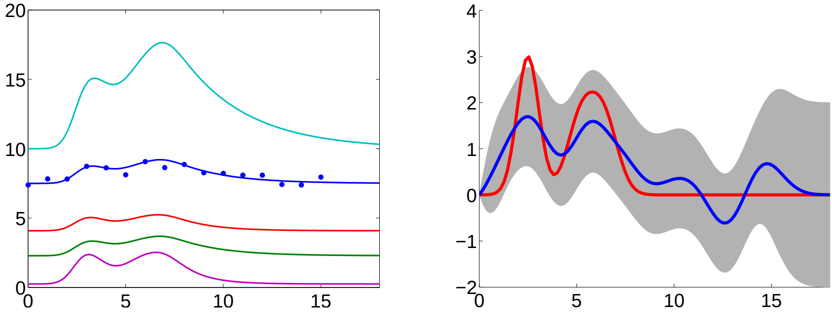



20
15
10
5
0
0
5
10
15
4
3
2
1
0
−1
−2
0
5
10
15

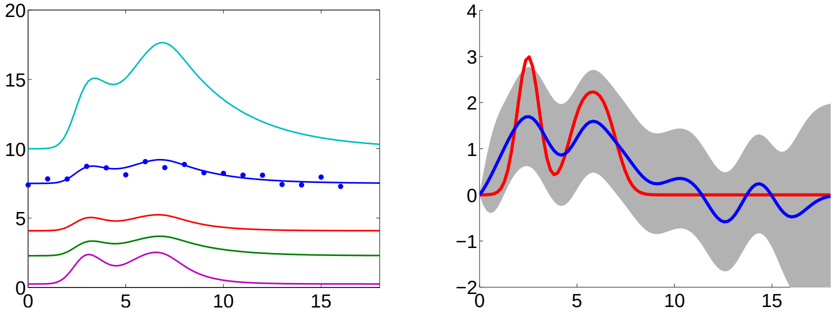
Artificial Example: Inferring $f(t)$



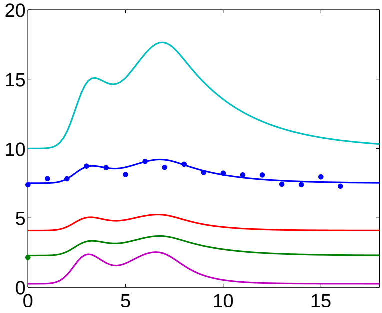
Artificial Example: Inferring $f(t)$

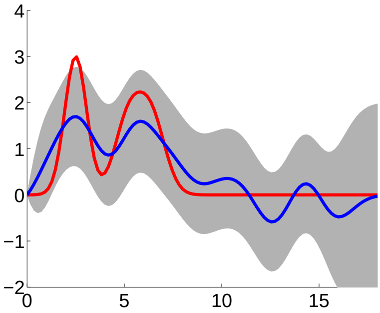


Artificial Example: Inferring $f(t)$

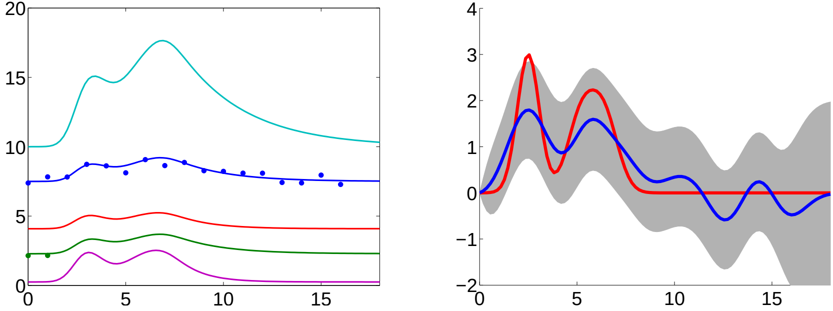


Artificial Example: Inferring $f(t)$

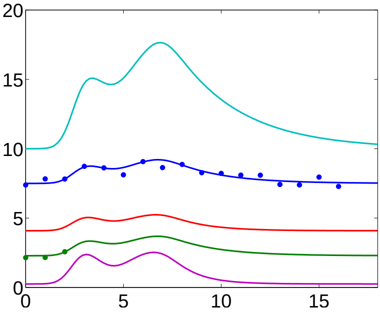


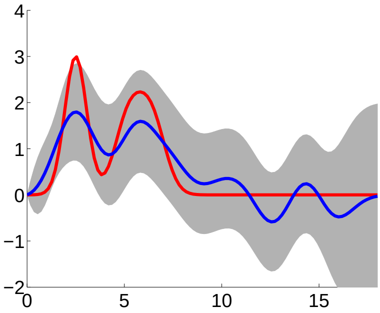


Artificial Example: Inferring $f(t)$

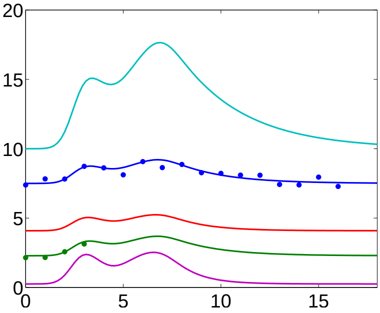


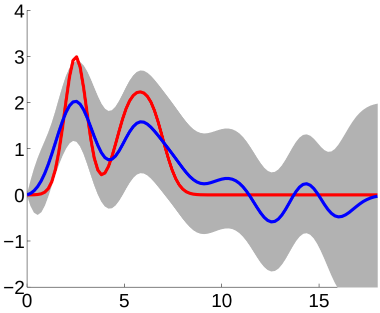
Artificial Example: Inferring $f(t)$



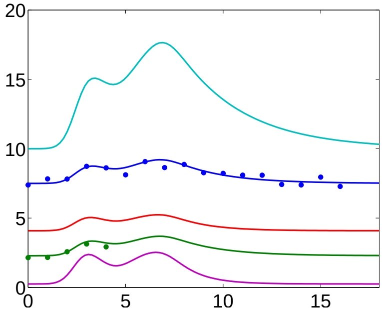


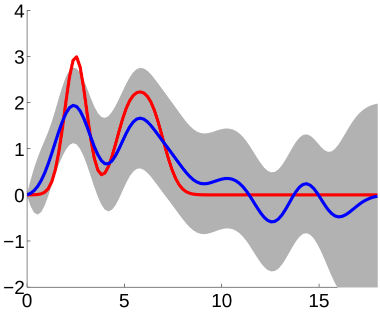
Artificial Example: Inferring $f(t)$



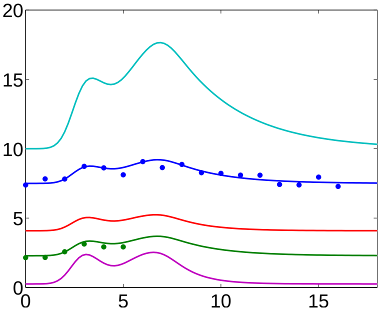


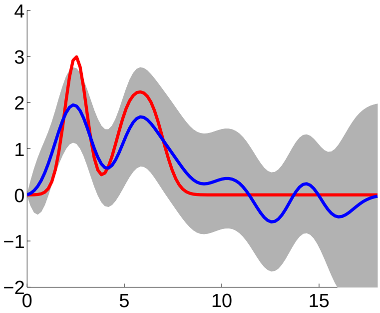
Artificial Example: Inferring $f(t)$



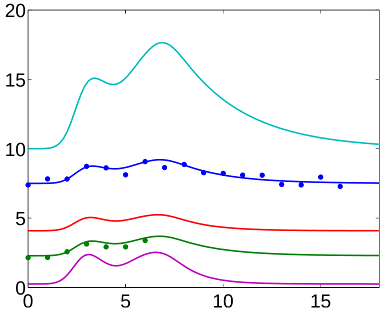


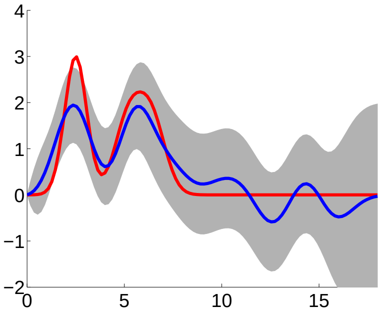
Artificial Example: Inferring $f(t)$



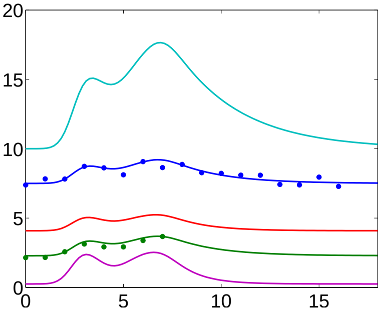


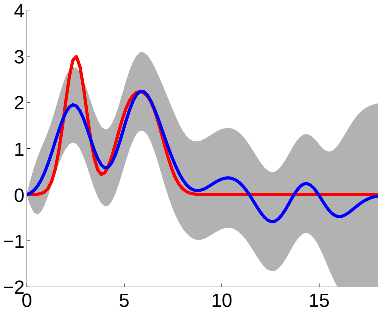
Artificial Example: Inferring $f(t)$



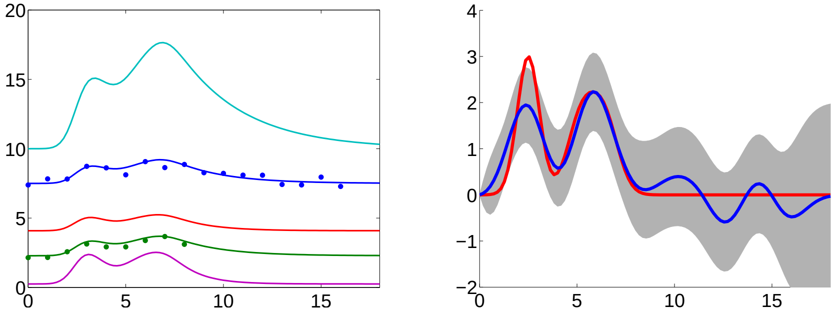


Artificial Example: Inferring $f(t)$

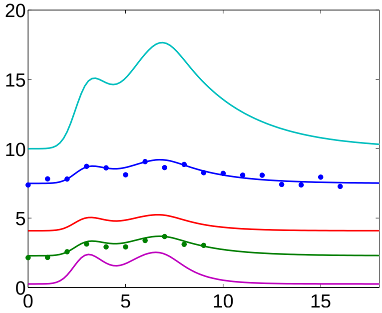


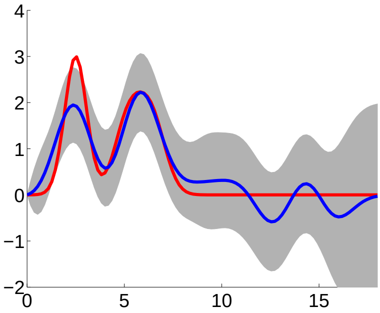


Artificial Example: Inferring $f(t)$

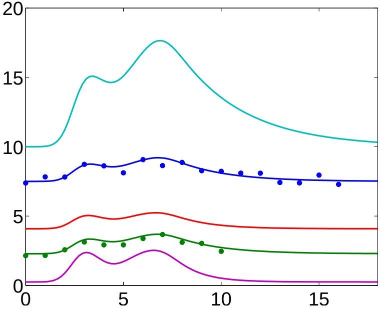


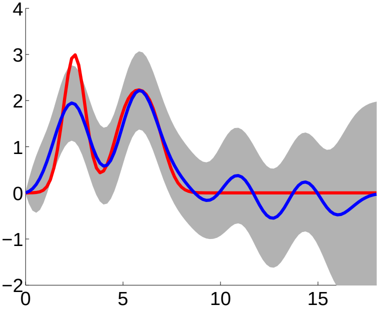
Artificial Example: Inferring $f(t)$



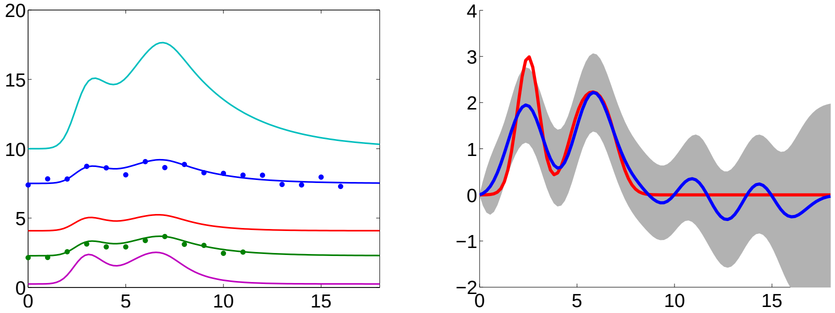


Artificial Example: Inferring $f(t)$

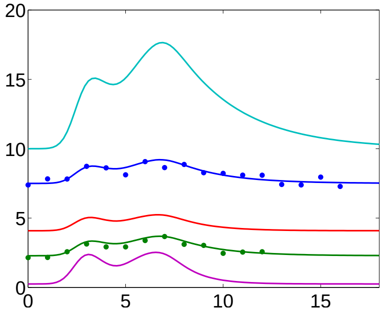


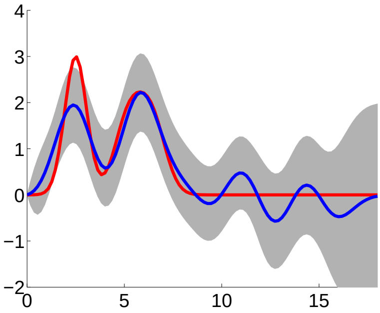


Artificial Example: Inferring $f(t)$

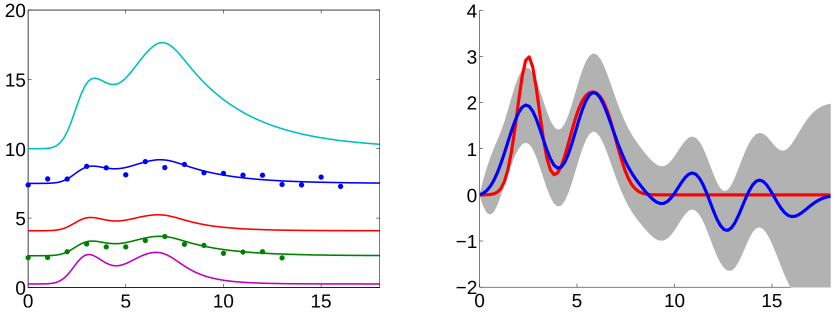


Artificial Example: Inferring $f(t)$

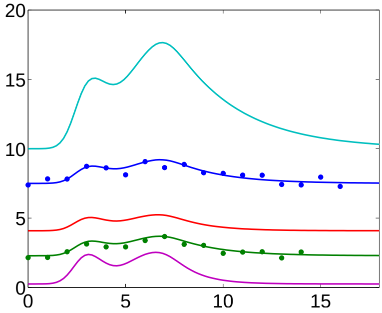


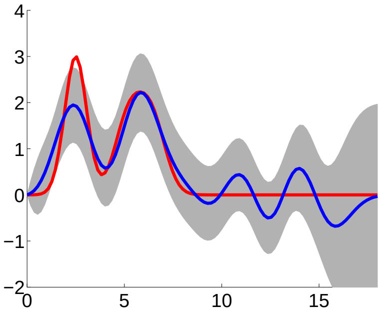


Artificial Example: Inferring $f(t)$

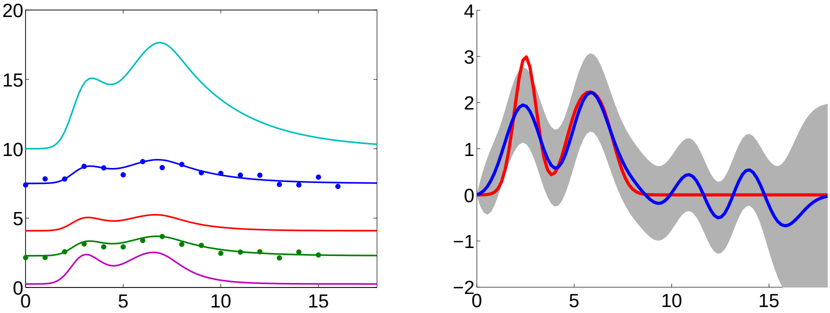


Artificial Example: Inferring $f(t)$

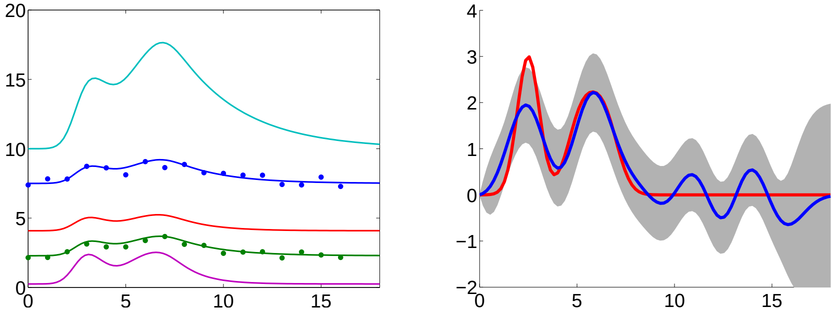




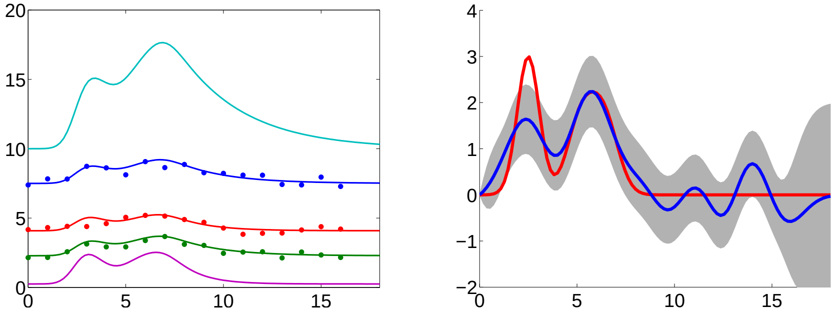
Artificial Example: Inferring $f(t)$



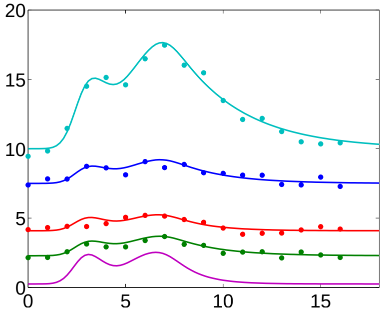
Artificial Example: Inferring $f(t)$

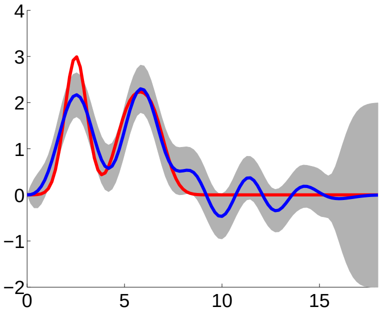


Artificial Example: Inferring $f(t)$

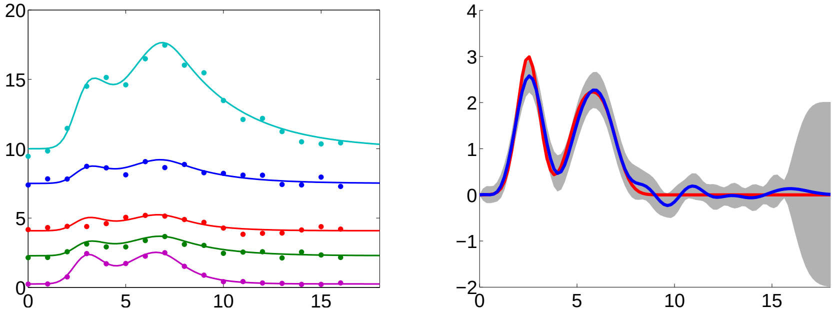


Artificial Example: Inferring $f(t)$





Artificial Example: Inferring $f(t)$



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

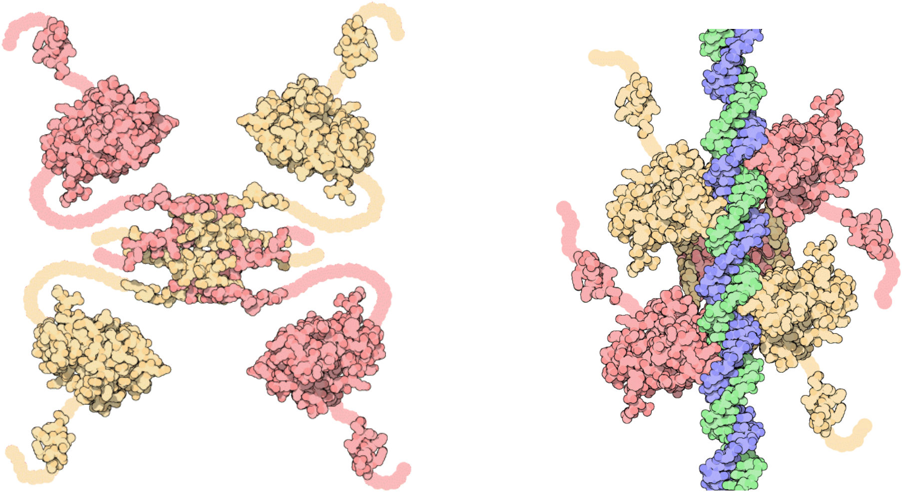


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the “Molecule of the Month” feature).

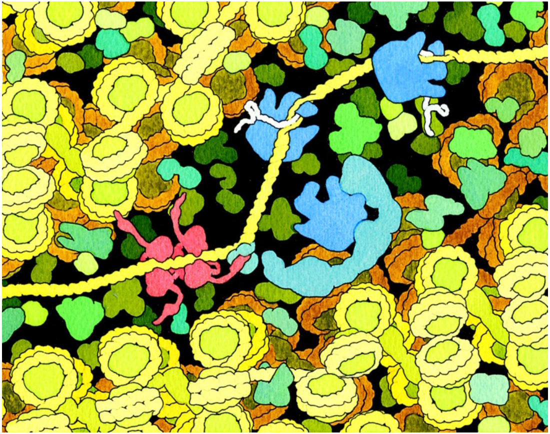


Figure: Repair of DNA damage by p53. Image fromGoodsell (1999).

Modelling Assumption

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

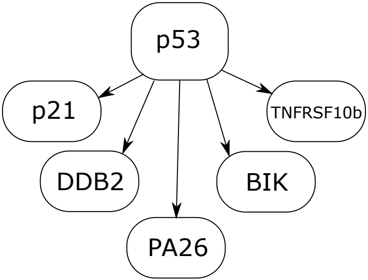
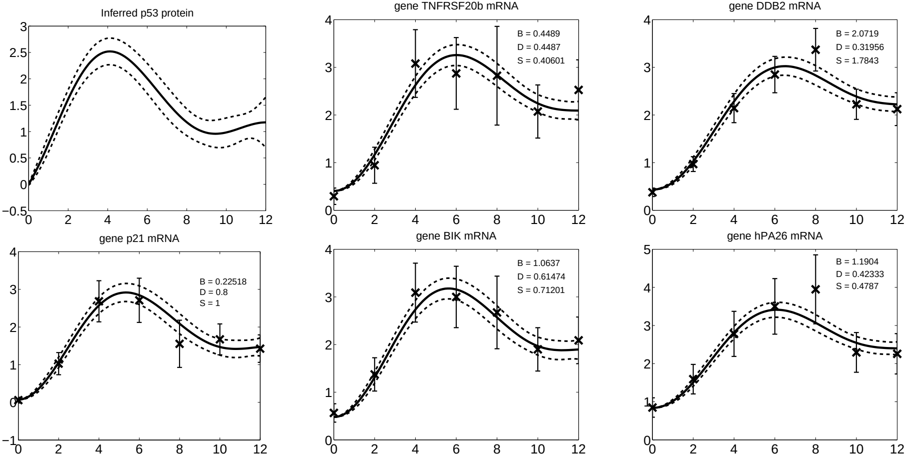


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

p53 (RBF covariance)

Pei Gao

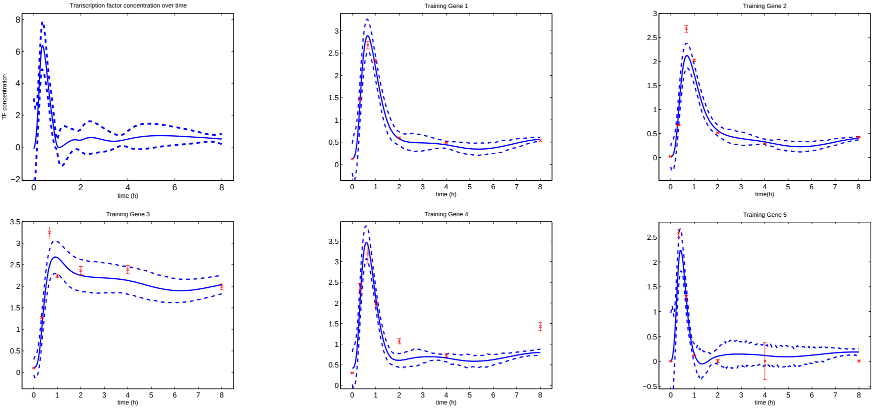


Ranking with ERK Signalling

- 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.

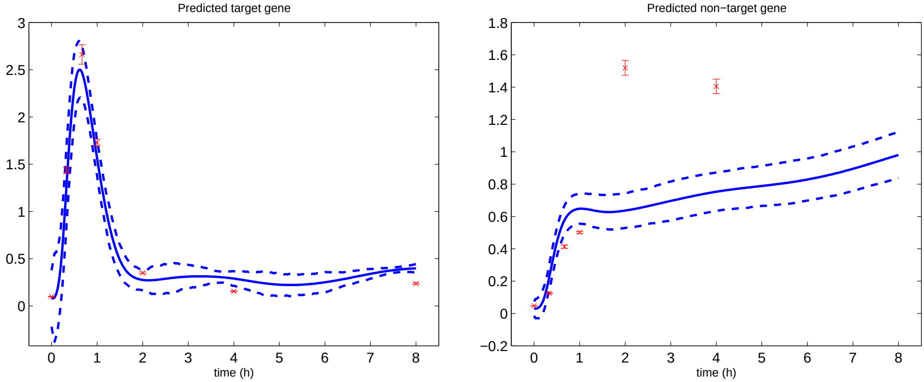
Elk-1 (MLP covariance)

Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



Roadmap

Cascaded Differential Equations

Antti Honkela

- 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

Antti Honkela

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

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = \sigma y(t) - \delta f(t)$$
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

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

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

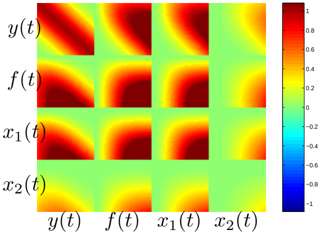
Covariance for Translation/Transcription Model

RBF covariance function for $y(t)$

$$
\begin{aligned}
f(t) &= \sigma \exp(-\delta t) \int_0^t y(u) \exp(\delta u) \,\mathrm{d}u \\
x_i(t) &= \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u) \,\mathrm{d}u.
\end{aligned}
$$

- Joint distribution for $x_1(t)$, $x_2(t)$, $f(t)$ and $y(t)$.
- Here:

δ	D_1	S_1	D_2	S_2
0.1	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.

Results for Twi using the Cascade model

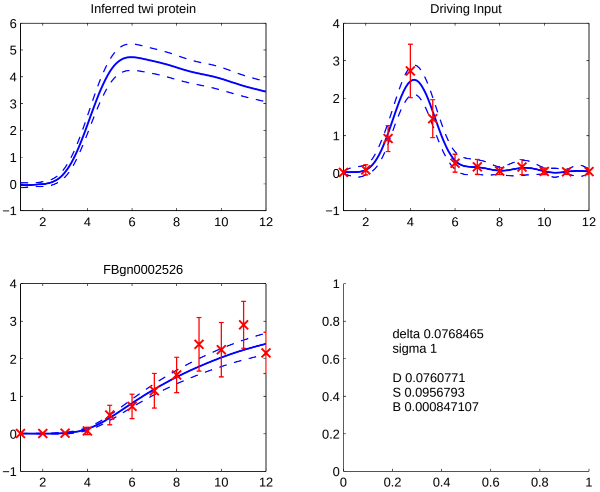


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

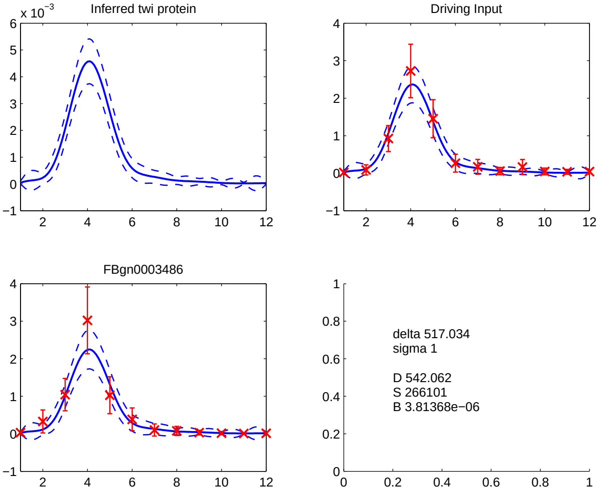


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

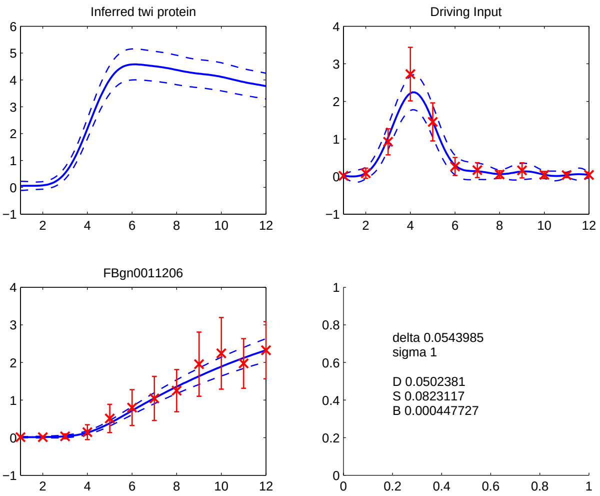


Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

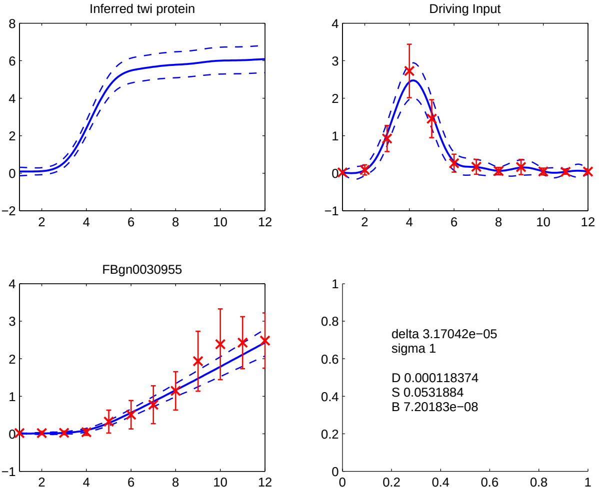


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

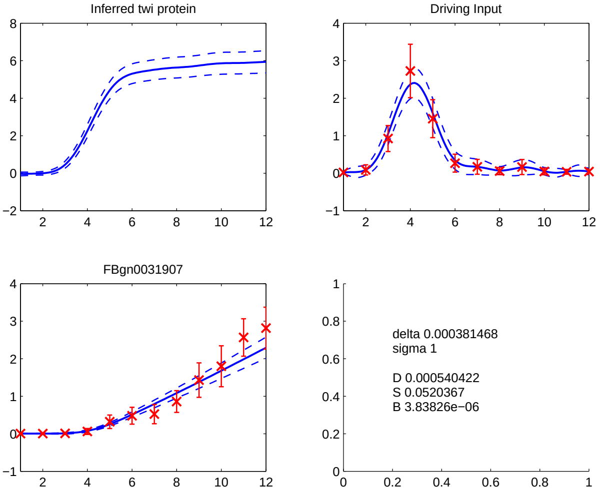


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

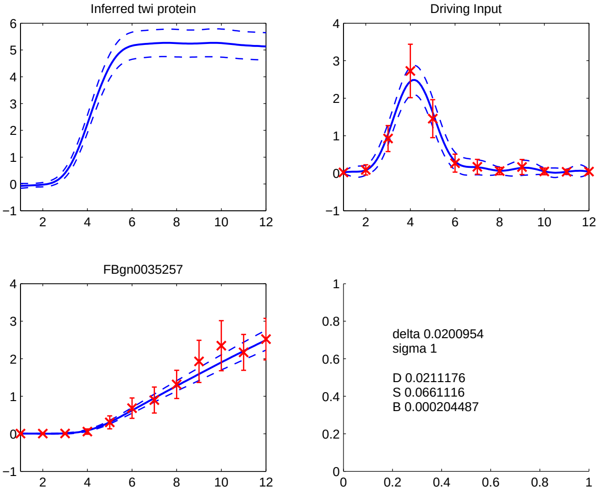


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

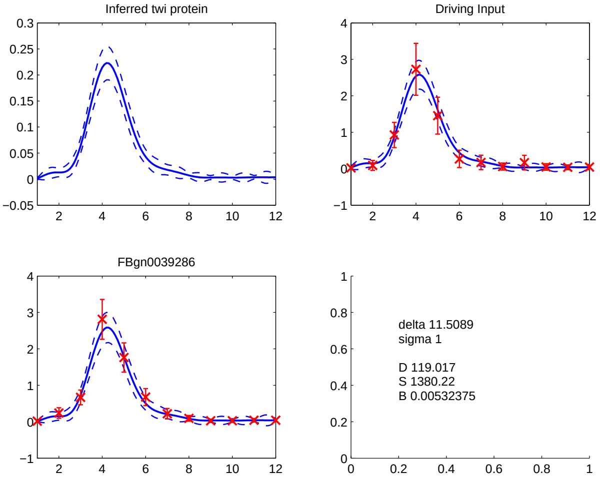


Figure: Model for flybase gene identity FBgn0039286.

Results of Ranking

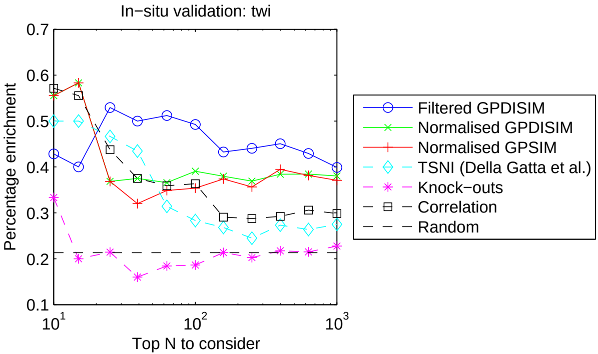


Figure: Percentage enrichment for top N targets for relevant terms in *Drosophila* in situs.

Results of Ranking

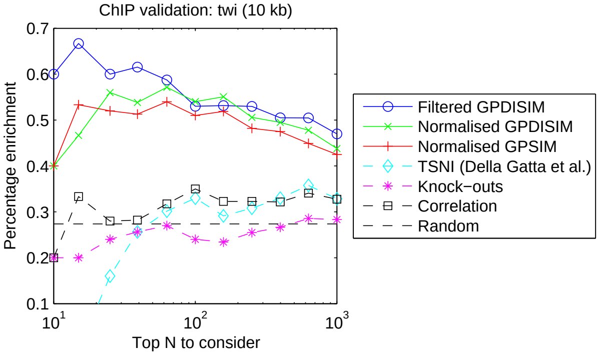


Figure: Percentage enrichment for top N targets for ChIP-chip confirmed targets.

Summary

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

Outline

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- These results are small simple systems.
- Other aspects:
  - Non-linear responses in differential equations (Michalis Titsias's work — turn to sampling, Pei Gao — use Laplace approximation).
  - Scaling up to larger systems (Mauricio's Talk).
  - Applications to other types of system, *e.g.* spatial systems etc. (using PDEs (Álvarez et al., 2009))
  - Stochastic differential equations (financial time series example).

Acknowledgements

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

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"

References I

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

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. [PDF].

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.

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

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. [PDF].

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

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

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