Dimensionality Reduction

Neil D. Lawrence

GPRS
25th–27th February 2015

Outline

Outline

Motivation for Non-Linear Dimensionality Reduction

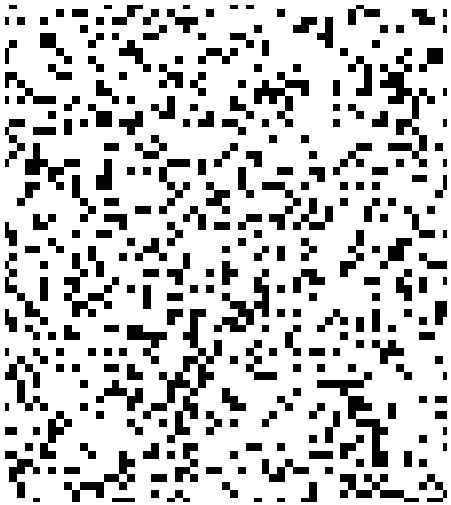
USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns

Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

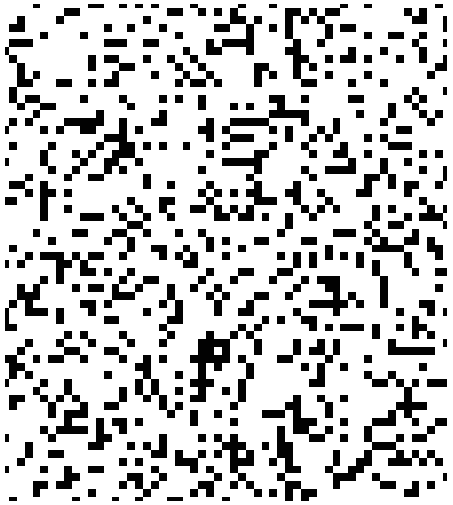
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

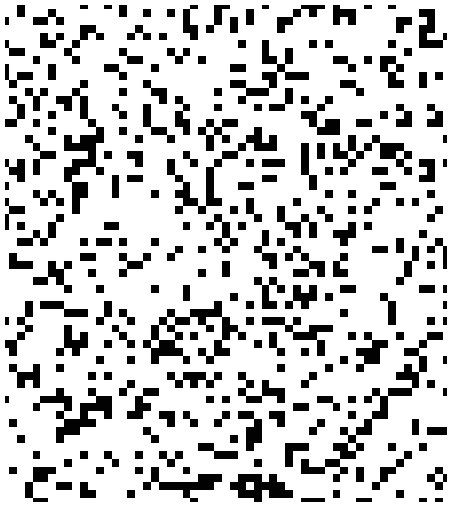
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

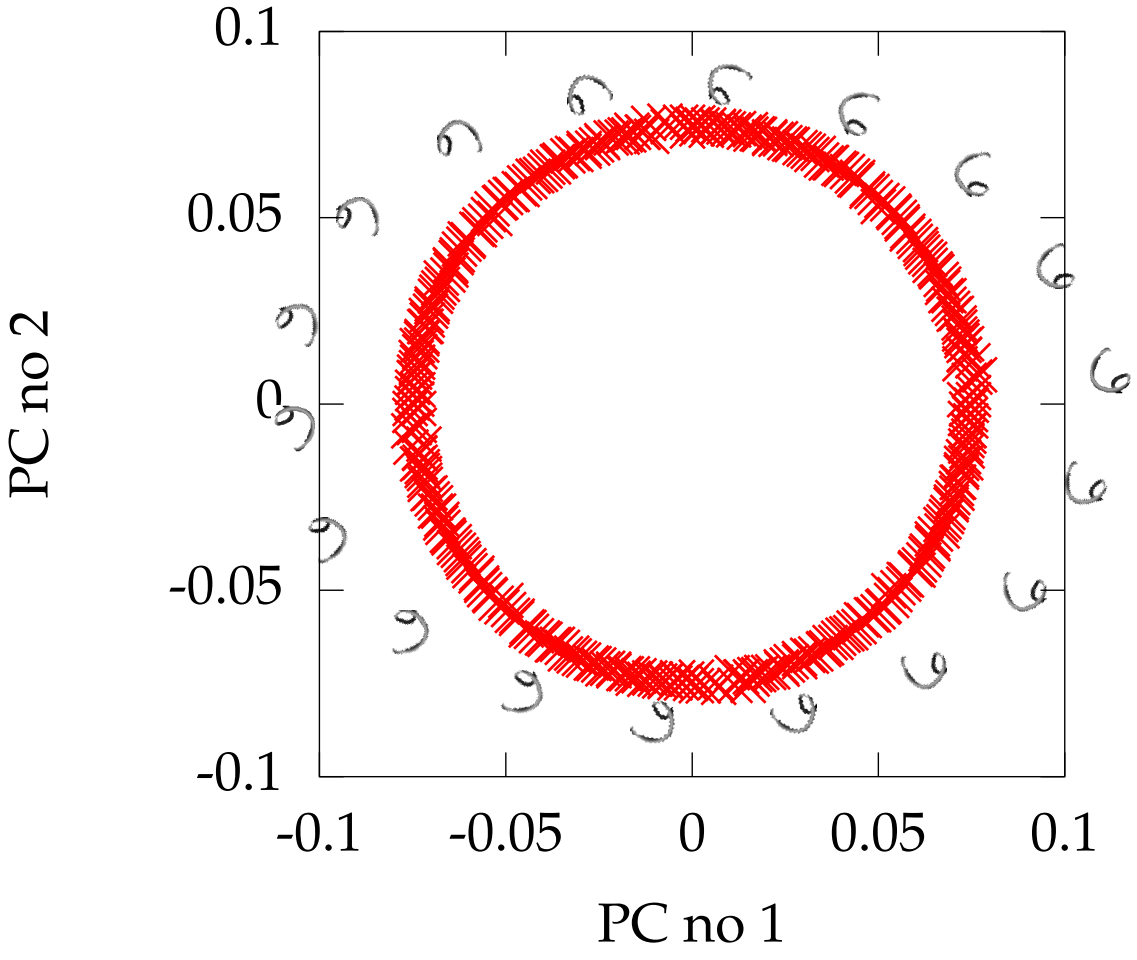
Rotate a 'Prototype'

MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

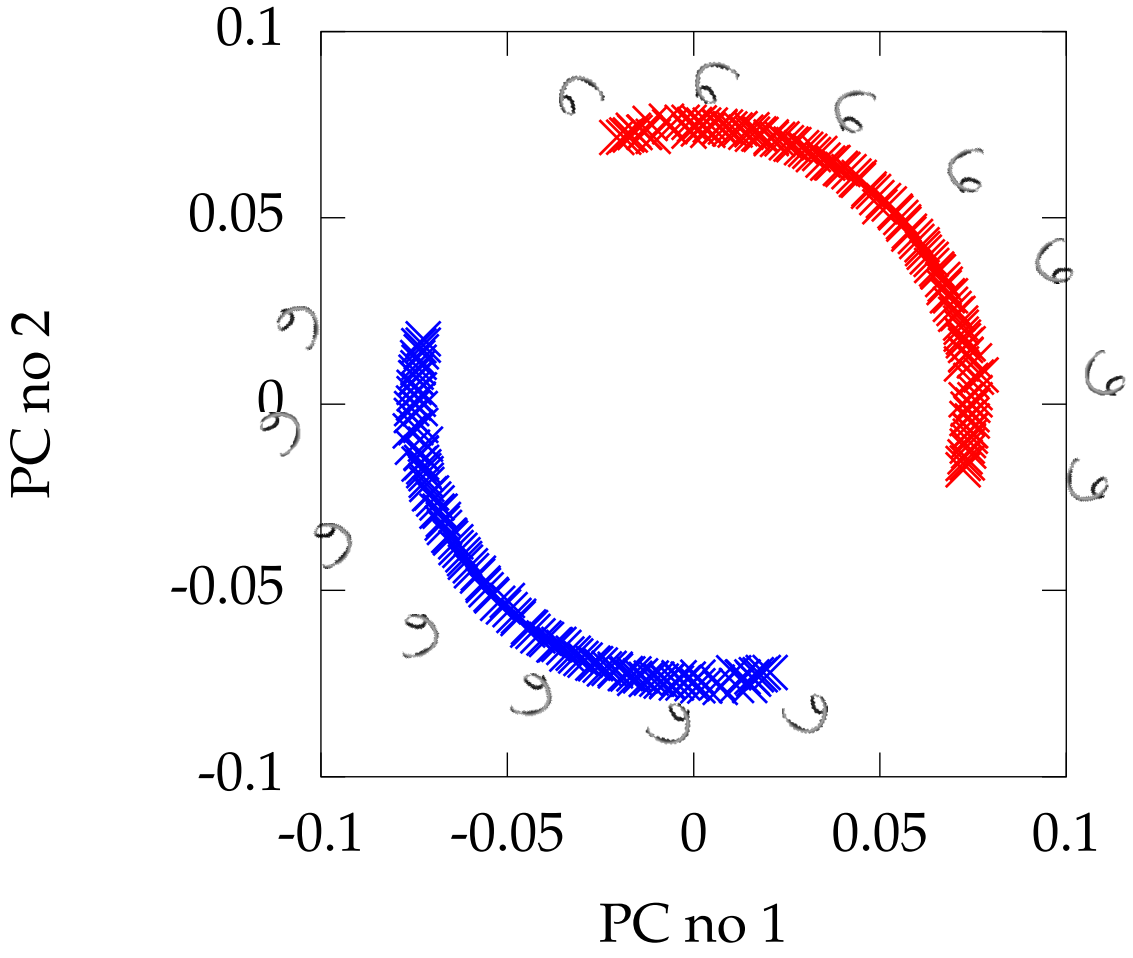
MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```



MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```



Low Dimensional Manifolds

Pure Rotation is too Simple

- In practice the data may undergo several distortions.
  - *e.g.* digits undergo 'thinning', translation and rotation.
- For data with 'structure':
  - we expect fewer distortions than dimensions;
  - we therefore expect the data to live on a lower dimensional manifold.
- Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Existing Methods

Spectral Approaches

- Classical Multidimensional Scaling (MDS) (Mardia et al., 1979).
  - Uses eigenvectors of similarity matrix.
    - Isomap (Tenenbaum et al., 2000) is MDS with a particular proximity measure.
  - Kernel PCA (Schölkopf et al., 1998)
    - Provides a representation and a mapping — dimensional expansion.
    - Mapping is implied throught he use of a kernel function as a similarity matrix.
  - Locally Linear Embedding (Roweis and Saul, 2000).
    - Looks to preserve locally linear relationships in a low dimensional space.

Existing Methods II

Iterative Methods

- Multidimensional Scaling (MDS)
  - Iterative optimisation of a stress function (Kruskal, 1964).
  - Sammon Mappings (Sammon, 1969).
    - Strictly speaking not a mapping — similar to iterative MDS.
- NeuroScale (Lowe and Tipping, 1997)
  - Augmentation of iterative MDS methods with a mapping.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.
- Generative Topographic Mapping (GTM) (Bishop et al., 1998)
  - Uses a grid based sample and an RBF network.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.
- Generative Topographic Mapping (GTM) (Bishop et al., 1998)
  - Uses a grid based sample and an RBF network.

Difficulty for Probabilistic Approaches

- Propagate a probability distribution through a non-linear mapping.

The New Model

A Probabilistic Non-linear PCA

- PCA has a probabilistic interpretation (Tipping and Bishop, 1999; Roweis, 1998).
- It is difficult to 'non-linearise'.

Dual Probabilistic PCA

- We present a new probabilistic interpretation of PCA (Lawrence, 2005).
- This interpretation can be made non-linear.
- The result is non-linear probabilistic PCA.

Notation

q— dimension of latent/embedded space
p— dimension of data space
n— number of data points

centred data, $\mathbf{Y} = [\mathbf{y}_{1,:}, \ldots, \mathbf{y}_{n,:}]^\top = \left[\mathbf{y}_{:,1}, \ldots, \mathbf{y}_{:,p}\right] \in \Re^{n \times p}$
latent variables, $\mathbf{X} = [\mathbf{x}_{1,:}, \ldots, \mathbf{x}_{n,:}]^\top = \left[\mathbf{x}_{:,1}, \ldots, \mathbf{x}_{:,q}\right] \in \Re^{n \times q}$
mapping matrix, $\mathbf{W} \in \Re^{p \times q}$

$\mathbf{a}_{i,:}$ is a vector from the ith row of a given matrix $\mathbf{A}$
$\mathbf{a}_{:,j}$ is a vector from the jth row of a given matrix $\mathbf{A}$

Reading Notation

X and Y are *design matrices*

- Covariance given by $n^{-1}\mathbf{Y}^\top\mathbf{Y}$.
- Inner product matrix given by $\mathbf{Y}\mathbf{Y}^\top$.

Linear Dimensionality Reduction

Linear Latent Variable Model

- Represent data, $\mathbf{Y}$, with a lower dimensional set of latent variables $\mathbf{X}$.
- Assume a linear relationship of the form

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:},$$

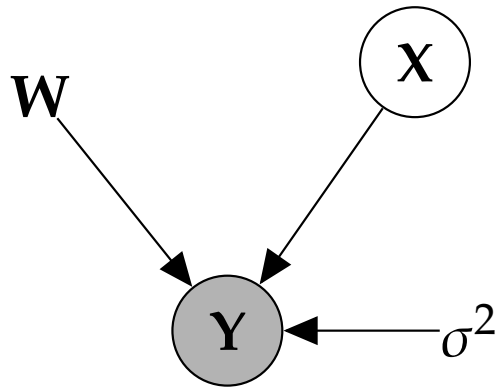
where

$$\boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right).$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

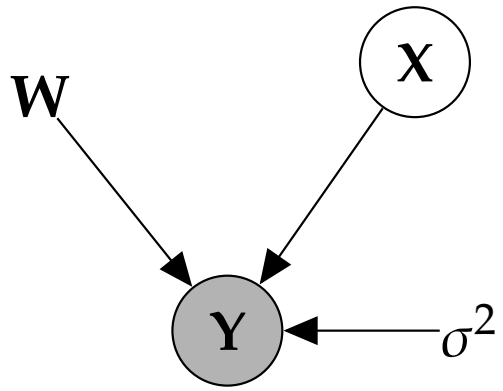


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:

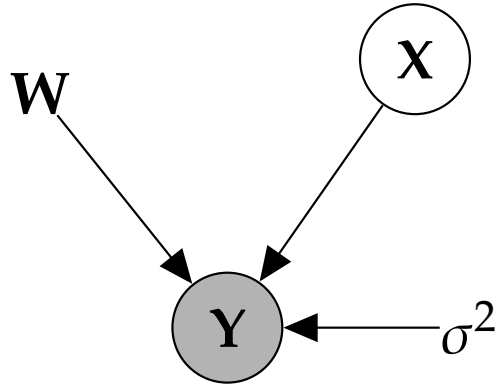


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.



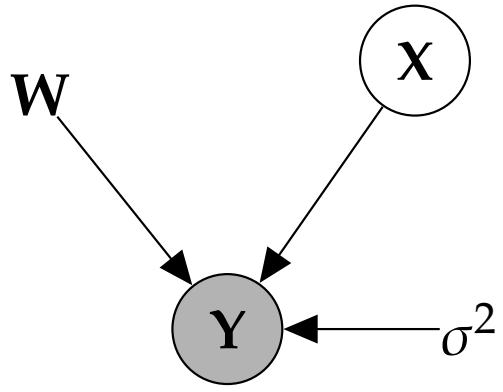
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top),$$

Computation of the Marginal Likelihood

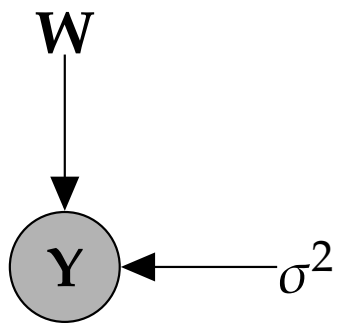
$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top),$$

$$\mathbf{W}\mathbf{x}_{i,:} + \epsilon_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)



$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

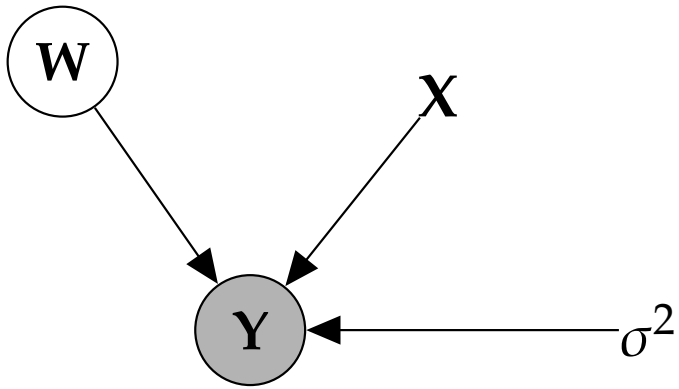
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

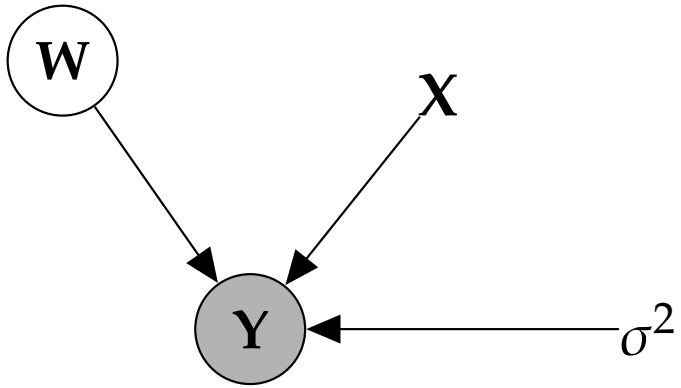


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:

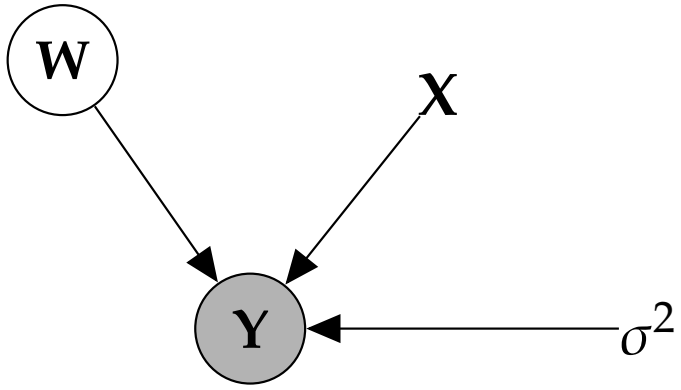


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, **W**.



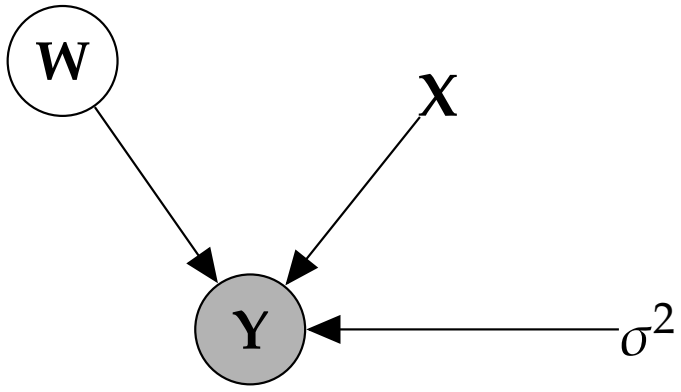
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, $\mathbf{W}$.
  - Integrate out *parameters*.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$$

$$\mathbf{X}\mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top),$$

Computation of the Marginal Likelihood

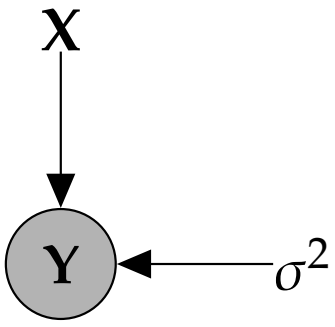
$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{X}\mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top),$$

$$\mathbf{X}\mathbf{w}_{:,j} + \epsilon_{:,j} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004, 2005)



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model IV

Dual PPCA Max. Likelihood Soln (Lawrence, 2004, 2005)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model IV

Dual PPCA Max. Likelihood Soln (Lawrence, 2004, 2005)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

$$\mathbf{Y}^\top \mathbf{Y}\mathbf{U}_q = \mathbf{U}_q \mathbf{\Lambda}_q \qquad \mathbf{W} = \mathbf{U}_q \mathbf{L}\mathbf{R}^\top$$

- Solution for Dual Probabilistic PCA (solves for the latent positions)

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{U}'_q = \mathbf{U}'_q \mathbf{\Lambda}_q \qquad \mathbf{X} = \mathbf{U}'_q \mathbf{L}\mathbf{R}^\top$$

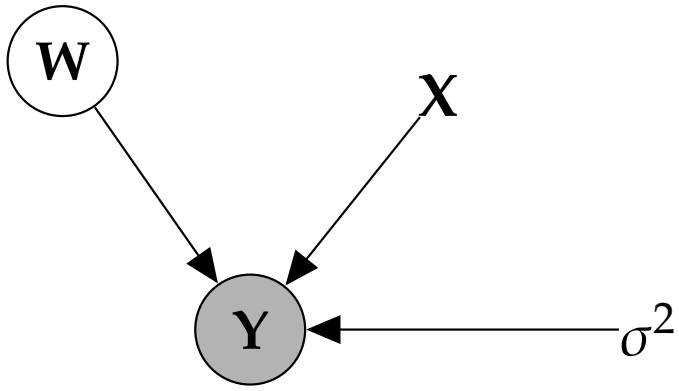
- Equivalence is from

$$\mathbf{U}_q = \mathbf{Y}^\top \mathbf{U}'_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameteters*, $\mathbf{W}$.
  - Integrate out *parameters*.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

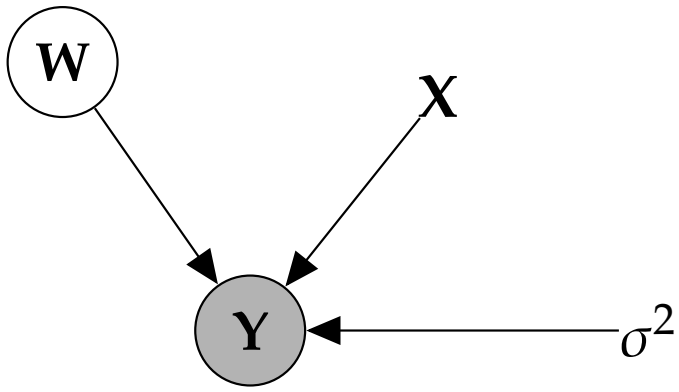
$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...

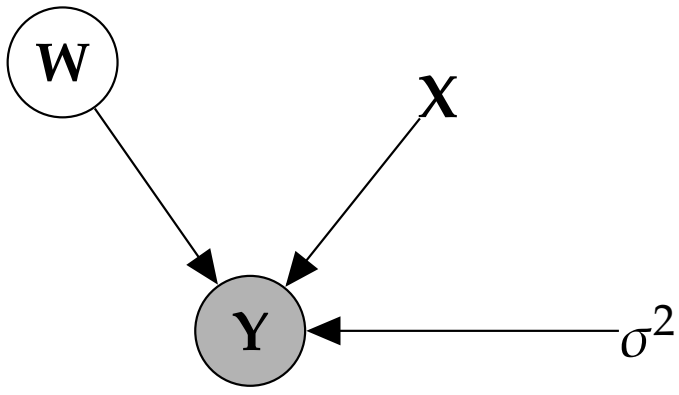


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.



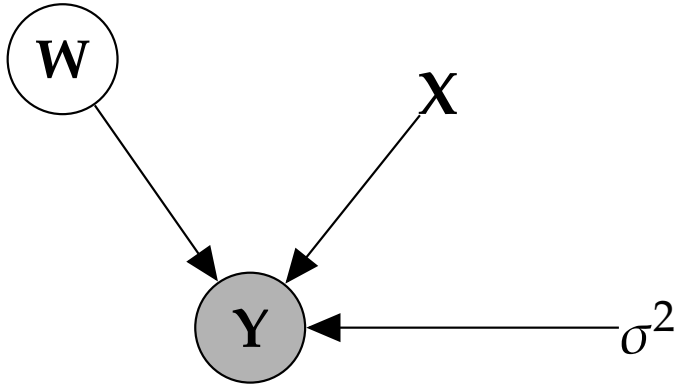
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

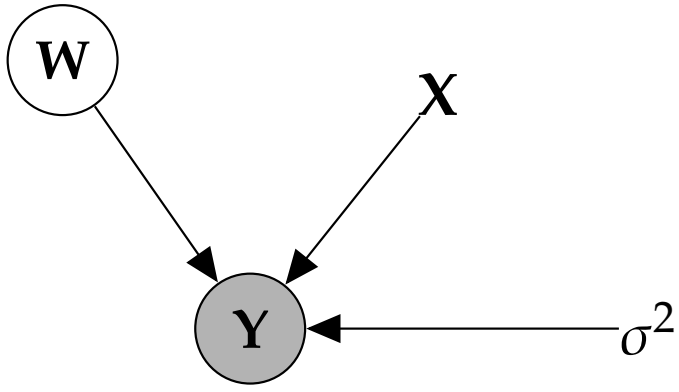
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

This is a product of Gaussian processes with linear kernels.

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$\mathbf{K}$ =?

Replace linear kernel with non-linear kernel for non-linear model.

Non-linear Latent Variable Models

Exponentiated Quadratic (EQ) Covariance

- The EQ covariance has the form $k_{i,j} = k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right)$, where

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2}{2\ell^2}\right).$$

- No longer possible to optimise wrt $\mathbf{X}$ via an eigenvalue problem.
- Instead find gradients with respect to $\mathbf{X}$, α, ℓ and σ^2 and optimise using conjugate gradients.

Applications

Style Based Inverse Kinematics

- Facilitating animation through modeling human motion (Grochow et al., 2004)

Tracking

- Tracking using human motion models (Urtasun et al., 2005, 2006)

Assisted Animation

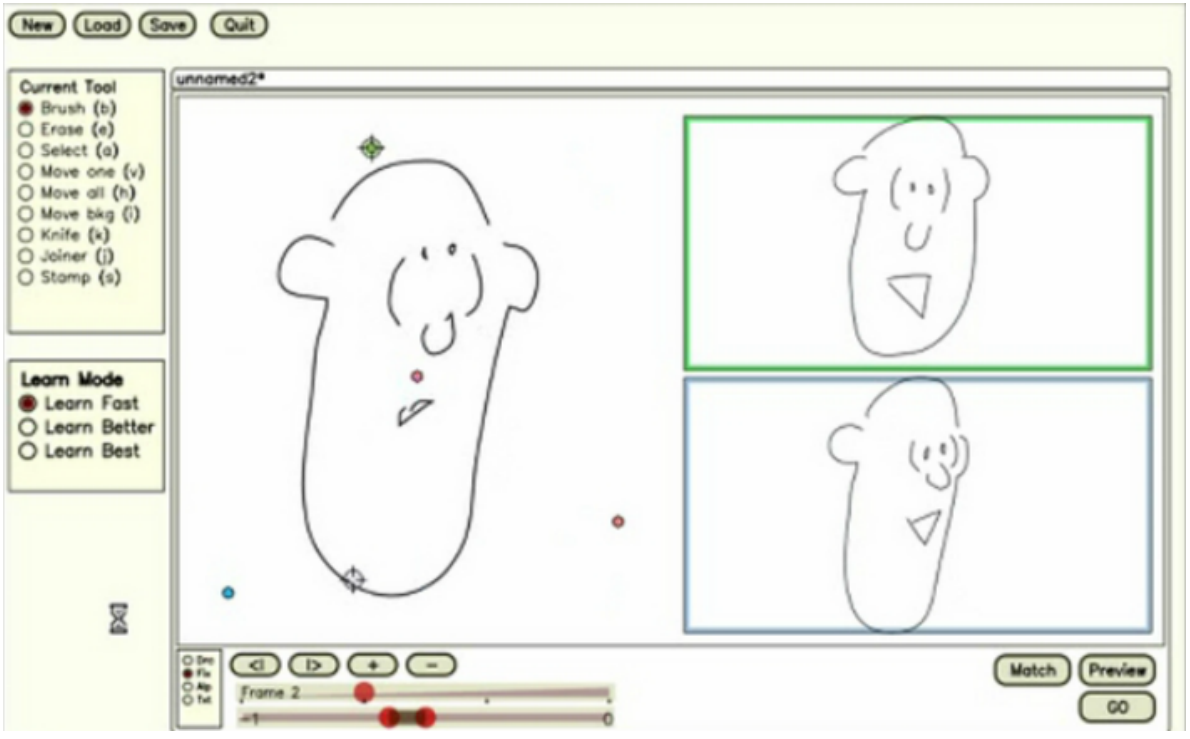
- Generalizing drawings for animation (Baxter and Anjyo, 2006)

Shape Models

- Inferring shape (e.g. pose from silhouette). (Ek et al., 2008b,a; Priacuriu and Reid, 2011a,b)

Example: Latent Doodle Space

(Baxter and Anjyo, 2006)



http://vimeo.com/3235882

Example: Latent Doodle Space

(Baxter and Anjyo, 2006)

Generalization with much less Data than Dimensions

- Powerful uncertainly handling of GPs leads to surprising properties.
- Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.

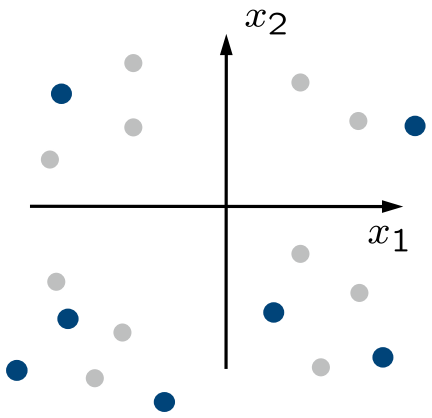
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



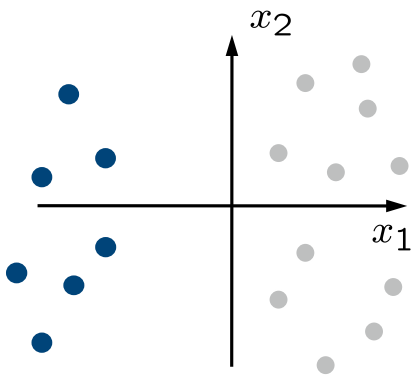
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\mathrm{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{n_i}{n}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^\top$$

where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the mean of all the training points of all classes.

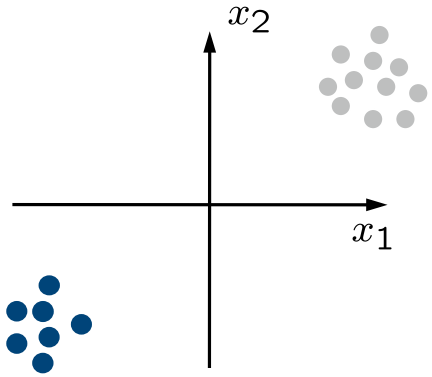
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{n_i}{n}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^\top$$

$$\mathbf{S}_b = \sum_{i=1}^{L} \frac{n_i}{n}\left[\frac{1}{n_i}\sum_{k=1}^{n_i}(\mathbf{x}_k^{(i)} - \mathbf{M}_i)(\mathbf{x}_k^{(i)} - \mathbf{M}_i)^\top\right]$$

where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the

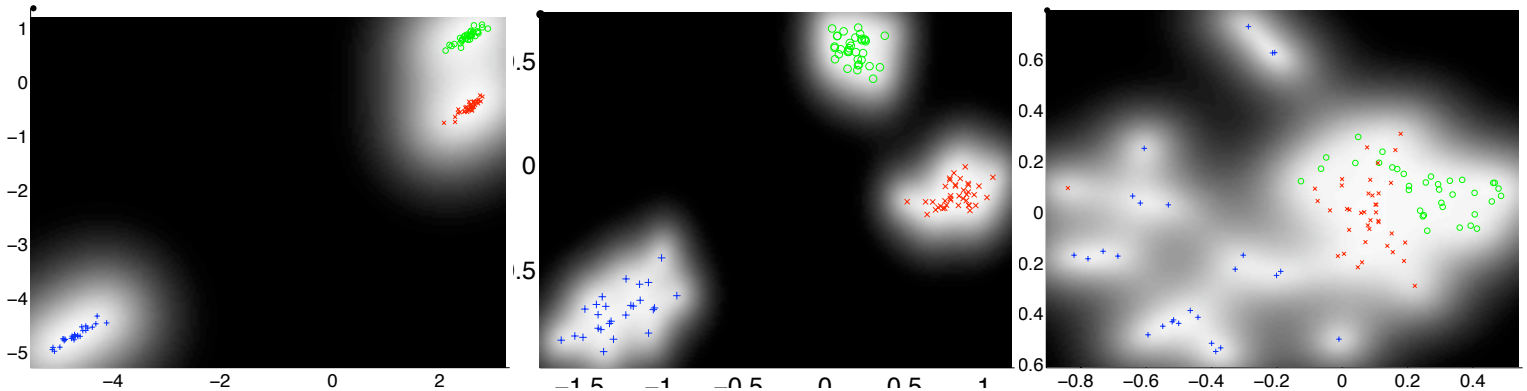
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



GaussianFace

(Lu and Tang, 2014)

- First system to surpass human performance on cropped Learning Faces in Wild Data.
  `http://tinyurl.com/nkt9a38`
- Lots of feature engineering, followed by a Discriminative GP-LVM.

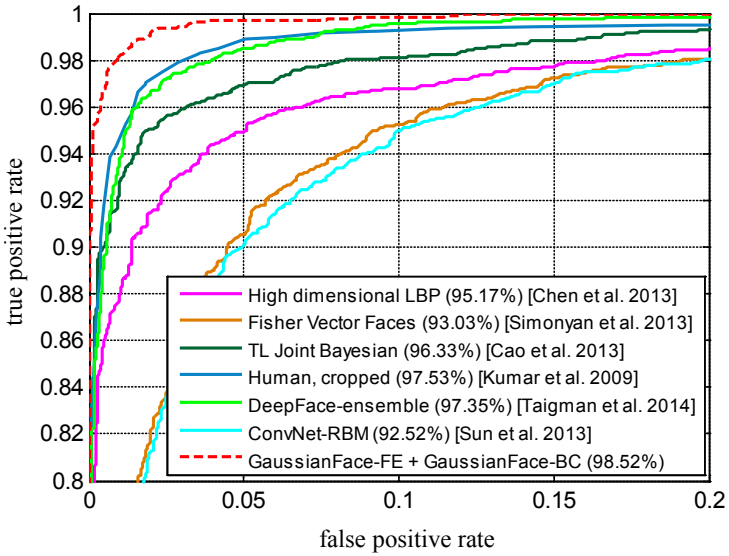


Figure 4: The ROC curve on LFW. Our method achieves the best performance, beating human-level performance.

Figure 5: The two rows present examples of matched and mismatched pairs respectively from LFW that were incorrectly classified by the GaussianFace model.

Conclusion and Future Work

Continuous Character Control

(Levine et al., 2012)

- Graph diffusion prior for enforcing connectivity between motions.

$$\log p(\mathbf{X}) = w_c \sum_{i,j} \log K^d_{ij}$$

with the graph diffusion kernel $\mathbf{K}^d$ obtain from

$$K^d_{ij} = \exp(\beta \mathbf{H}) \qquad \text{with} \qquad \mathbf{H} = -\mathbf{T}^{-1/2}\mathbf{L}\mathbf{T}^{-1/2}$$

the graph Laplacian, and $\mathbf{T}$ is a diagonal matrix with $T_{ii} = \sum_j w(\mathbf{x}_i, \mathbf{x}_j)$,

$$L_{ij} = \begin{cases} \sum_k w(\mathbf{x}_i, \mathbf{x}_k) & \text{if } i = j \\ -w(\mathbf{x}_i, \mathbf{x}_j) & \text{otherwise.} \end{cases}$$

and $w(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^{-p}$ measures similarity.

Character Control: Results

Other Topics

- Local distance preservation ▸ Details
- Dynamical models ▸ Details
- Hierarchical models ▸ Details
- Bayesian GP-LVM ▸ Details

Back Constraints I

Local Distance Preservation **(Lawrence and Quiñonero Candela, 2006)**

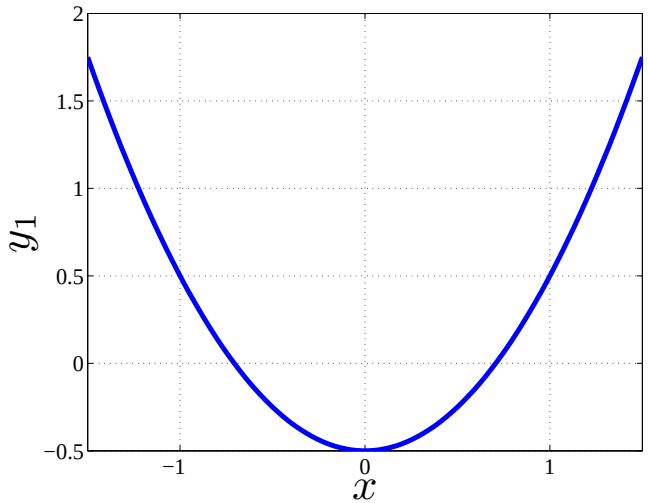
- Most dimensional reduction techniques preserve local distances.
- The GP-LVM does not.
- GP-LVM maps smoothly from latent to data space.
  - Points close in latent space are close in data space.
  - This does not imply points close in data space are close in latent space.
- Kernel PCA maps smoothly from data to latent space.
  - Points close in data space are close in latent space.
  - This does not imply points close in latent space are close in data space.

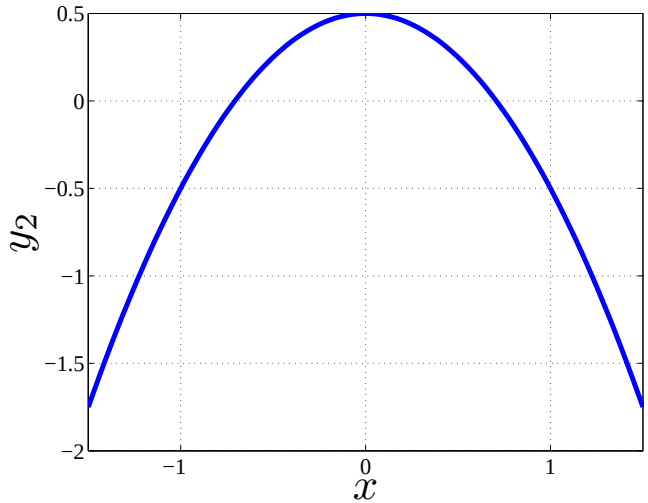
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



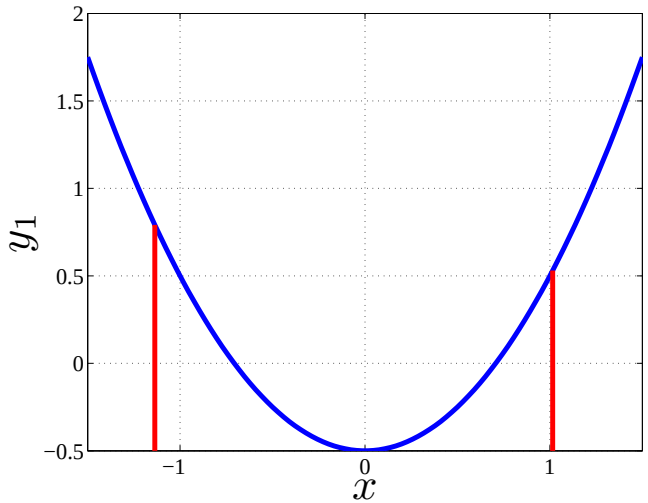


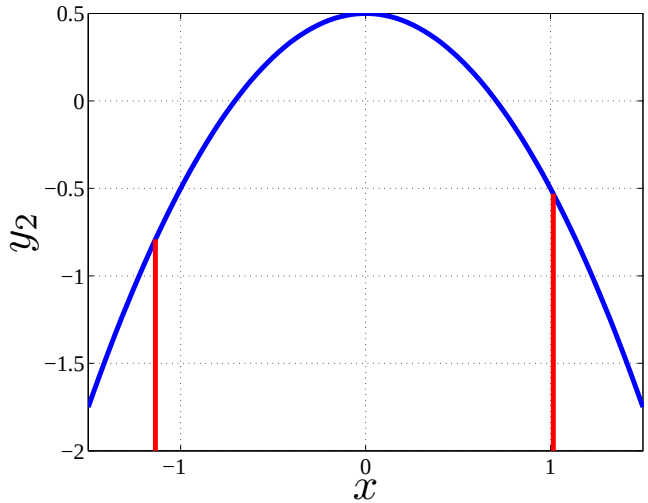
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



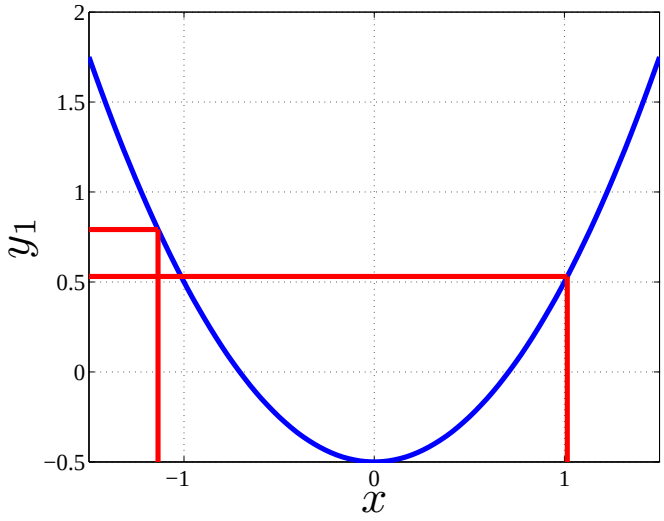


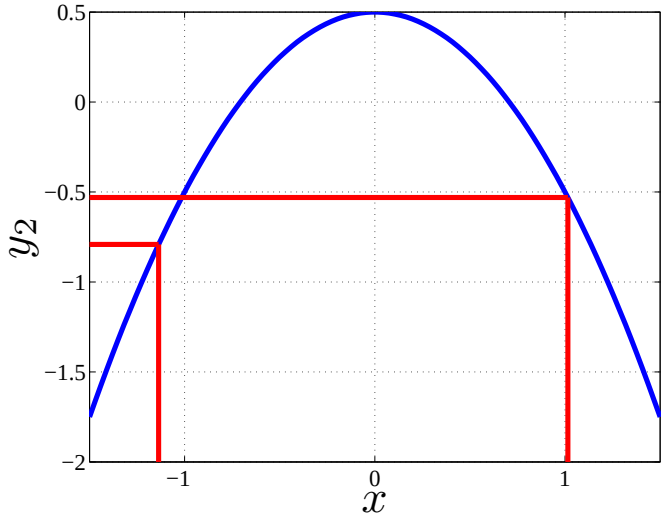
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



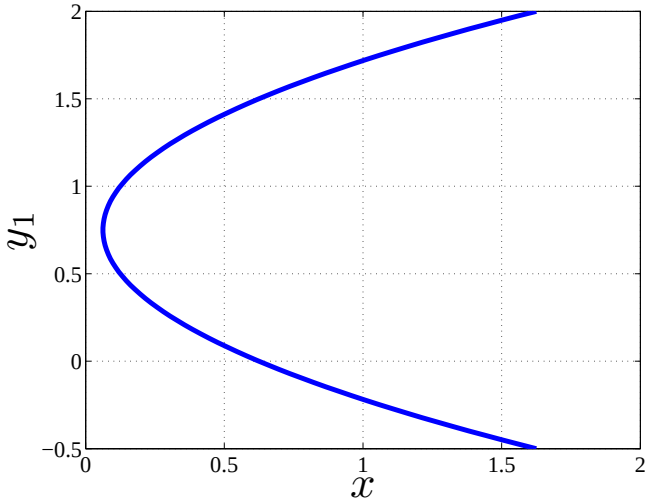


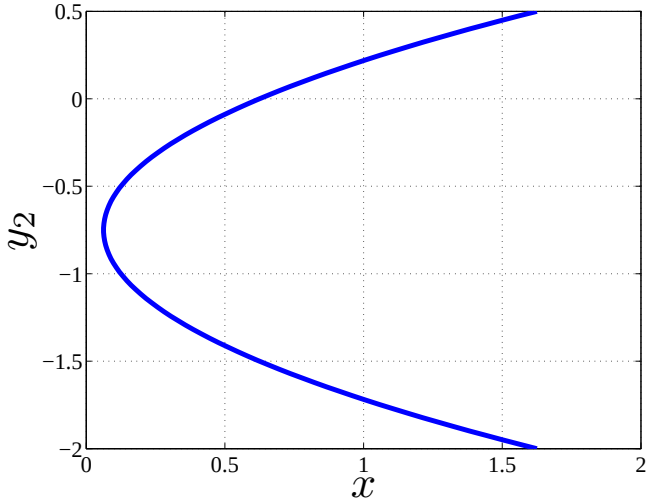
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



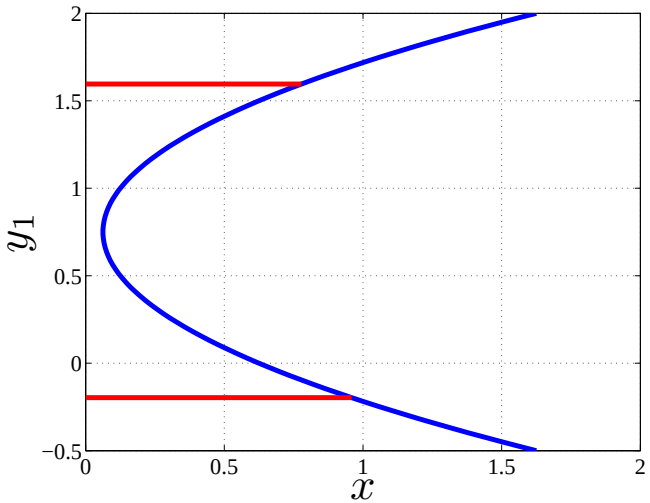


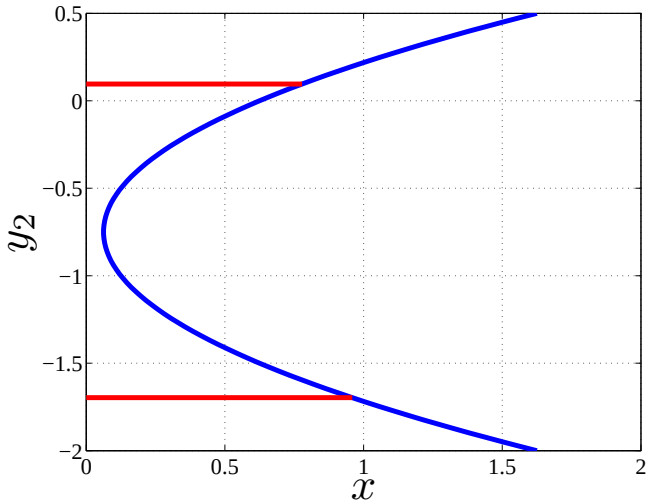
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



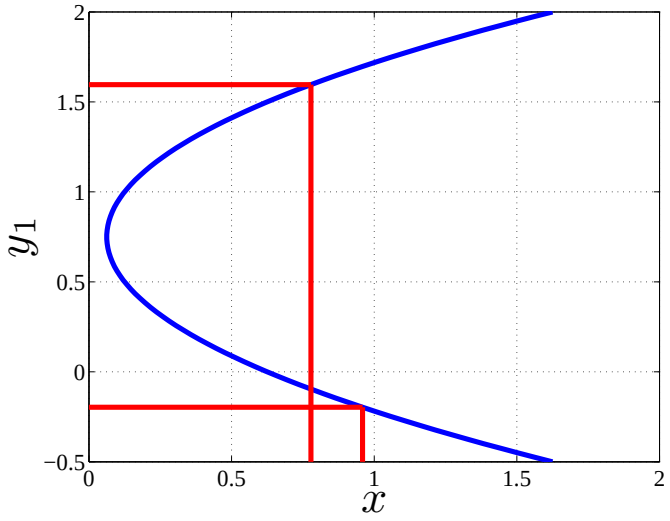


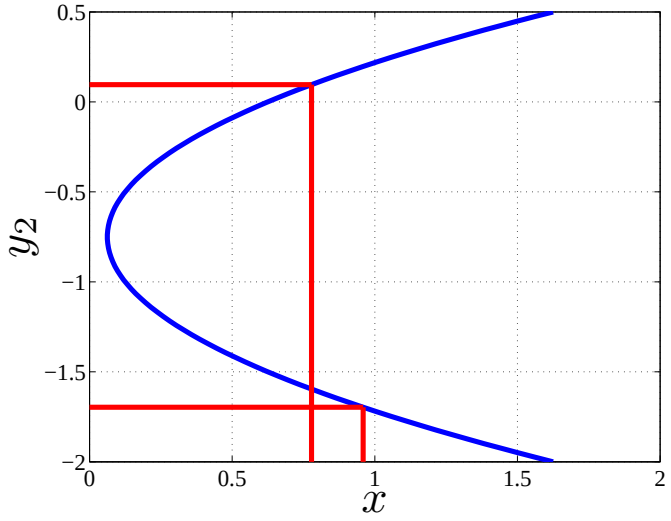
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$





Multi-Dimensional Scaling with a Mapping

- Lowe and Tipping (1997) made latent positions a function of the data.

$$x_{i,j} = f_j(\mathbf{y}_{i,:}; \mathbf{v})$$

- Function was either multi-layer perceptron or a radial basis function network.
- Their motivation was different from ours:
  - They wanted to add the advantages of a true mapping to multi-dimensional scaling.

Back Constraints in the GP-LVM

Back Constraints

- We can use the same idea to force the GP-LVM to respect local distances.(Lawrence and Quiñonero Candela, 2006)
  - By constraining each $\mathbf{x}_i$ to be a 'smooth' mapping from $\mathbf{y}_i$ local distances can be respected.
- This works because in the GP-LVM we maximise wrt latent variables, we don't integrate out.
- Can use any 'smooth' function:
  1. Neural network.
  2. RBF Network.
  3. Kernel based mapping.

Optimising BC-GPLVM

Computing Gradients

- GP-LVM normally proceeds by optimising

$$L(\mathbf{X}) = \log p(\mathbf{Y}|\mathbf{X})$$

  with respect to $\mathbf{X}$ using $\frac{\mathrm{d}L}{\mathrm{d}\mathbf{X}}$.
- The back constraints are of the form

$$x_{i,j} = f_j(\mathbf{y}_{i,:}; \mathbf{v})$$

  where $\mathbf{v}$ are parameters.
- We can compute $\frac{\mathrm{d}L}{\mathrm{d}\mathbf{v}}$ via chain rule and optimise parameters of mapping.

Motion Capture Results

`demStick1` **and** `demStick3`

Figure : The latent space for the motion capture data with (*right*) and without (*left*) back constraints.

Motion Capture Results

`demStick1` **and** `demStick3`

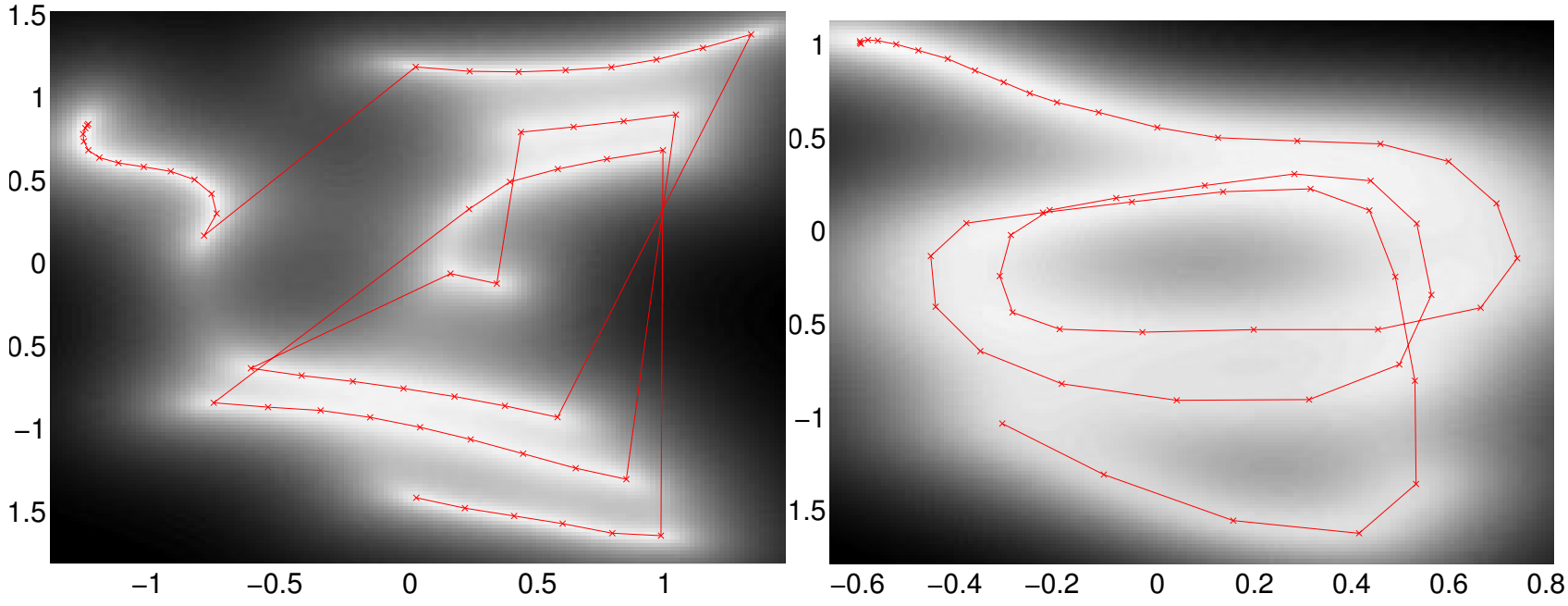
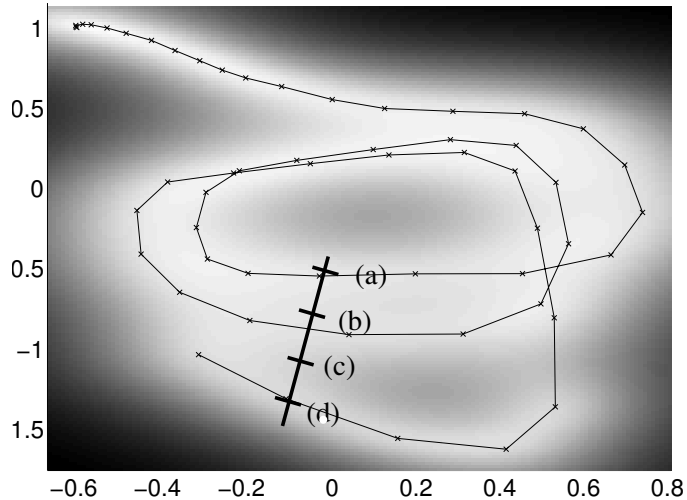


Figure : The latent space for the motion capture data with (*right*) and without (*left*) back constraints.

Stick Man Results

`demStickResults`



(a) (b) (c) (d)

Projection into data space from four points in the latent space. The inclination of the runner changes becoming more upright.

Adding Dynamics

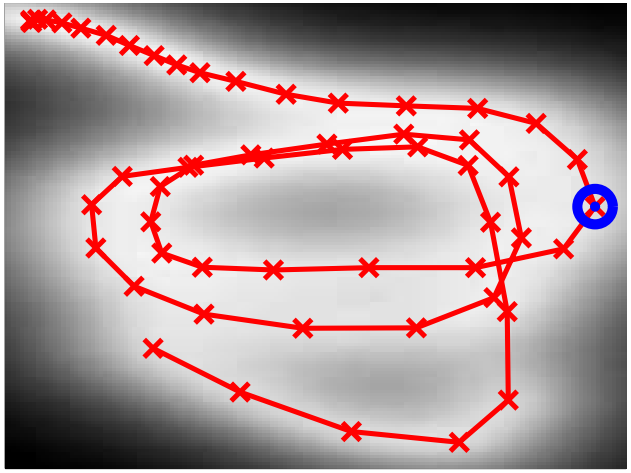
MAP Solutions for Dynamics Models

- Data often has a temporal ordering.
- Markov-based dynamics are often used.
- For the GP-LVM
  - Marginalising such dynamics is intractable.
  - But: MAP solutions are trivial to implement.
- Many choices: Kalman filter, Markov chains *etc*..
- Wang et al. (2006) suggest using a Gaussian Process.

Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.

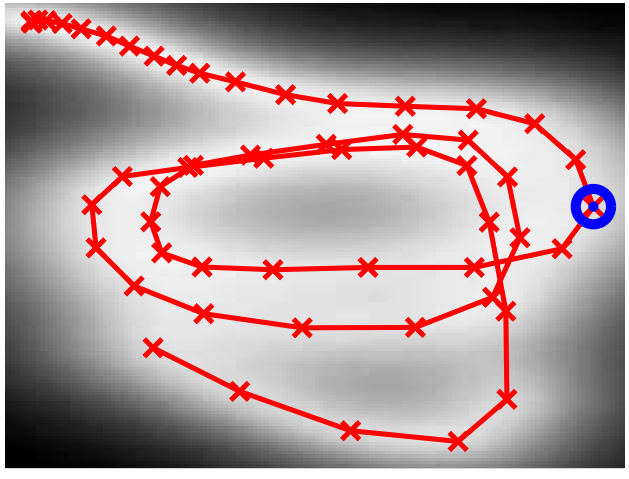


t

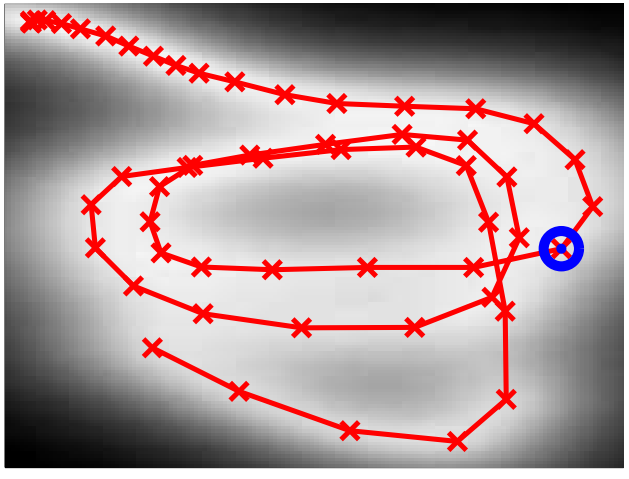
Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



t

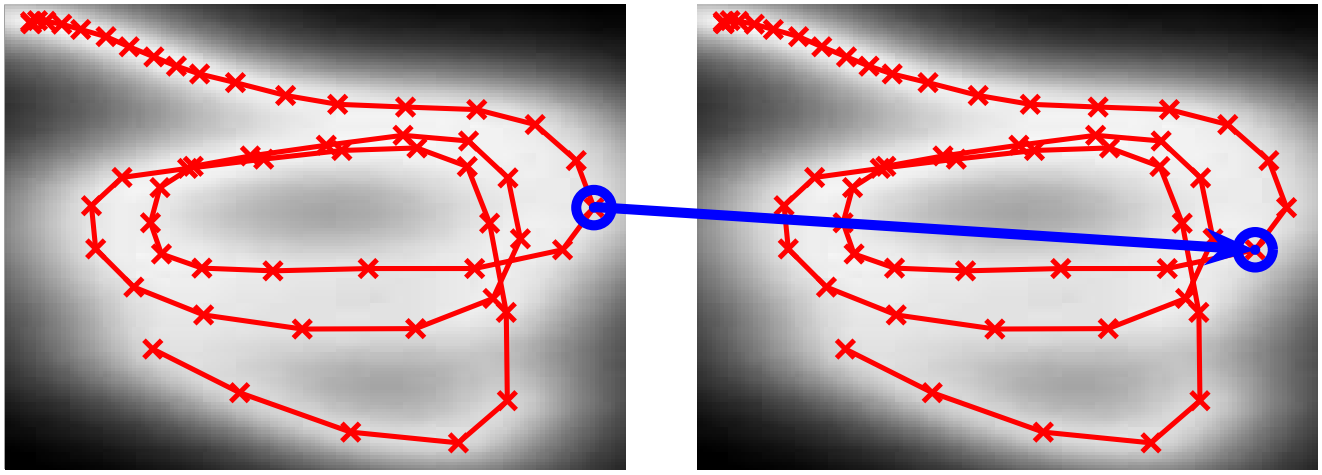


$t+1$

Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



t $\quad$ $t+1$

Motion Capture Results

`demStick1` **and** `demStick2`

Figure : The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Motion Capture Results

`demStick1` **and** `demStick2`

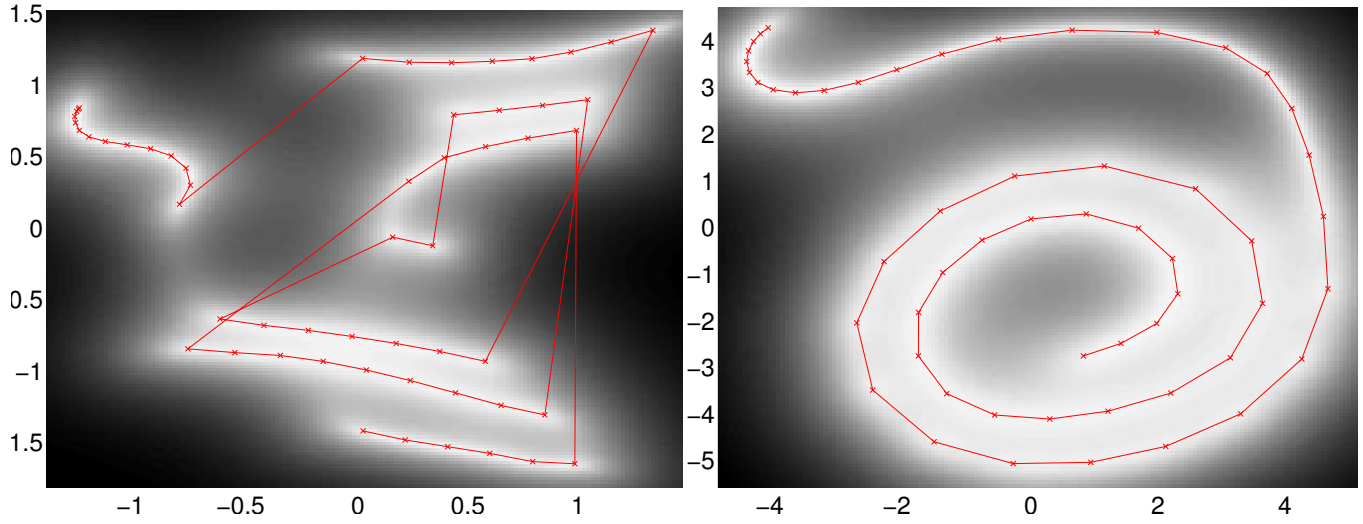
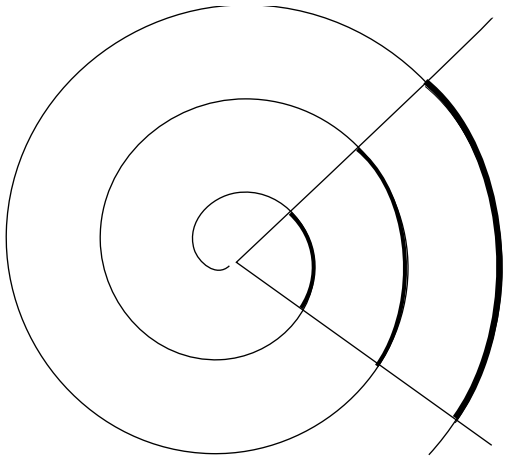


Figure : The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Regressive Dynamics

Inner Groove Distortion

- Autoregressive unimodal dynamics, $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$.
- Forces spiral visualisation.
- Poorer model due to inner groove distortion.



Regressive Dynamics

Direct use of Time Variable

- Instead of auto-regressive dynamics, consider regressive dynamics.
- Take $\mathbf{t}$ as an input, use a prior $p(\mathbf{X}|\mathbf{t})$.
- User a Gaussian process prior for $p(\mathbf{X}|\mathbf{t})$.
- Also allows us to consider variable sample rate data.

Motion Capture Results

`demStick1`, `demStick2` **and** `demStick5`

Figure : The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Motion Capture Results

`demStick1`, `demStick2` **and** `demStick5`

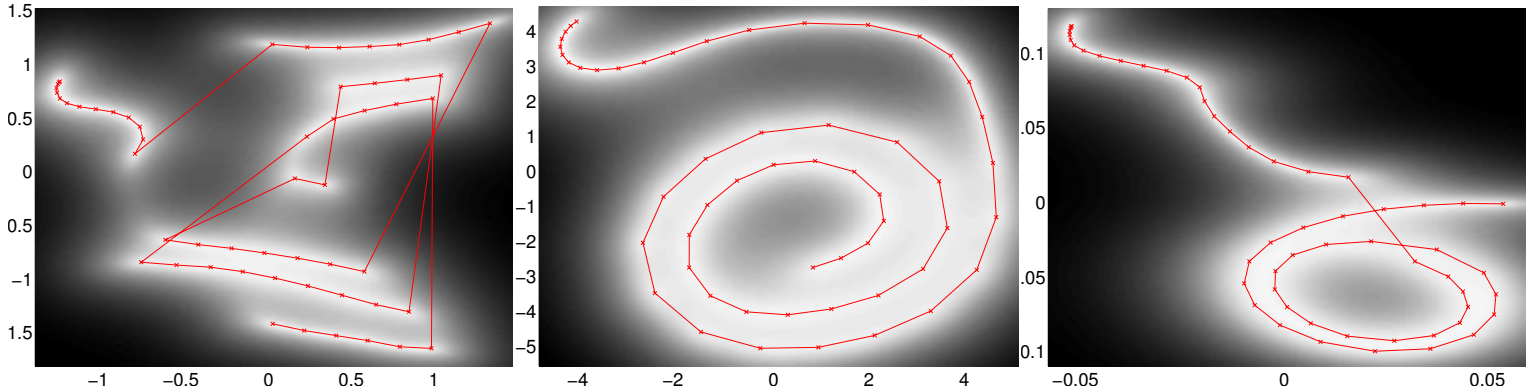


Figure : The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Return

Hierarchical GP-LVM

(Lawrence and Moore, 2007)

Stacking Gaussian Processes

- Regressive dynamics provides a simple hierarchy.
  - The input space of the GP is governed by another GP.
- By stacking GPs we can consider more complex hierarchies.
- Ideally we should marginalise latent spaces
  - In practice we seek MAP solutions.

Two Correlated Subjects

(Lawrence and Moore, 2007)

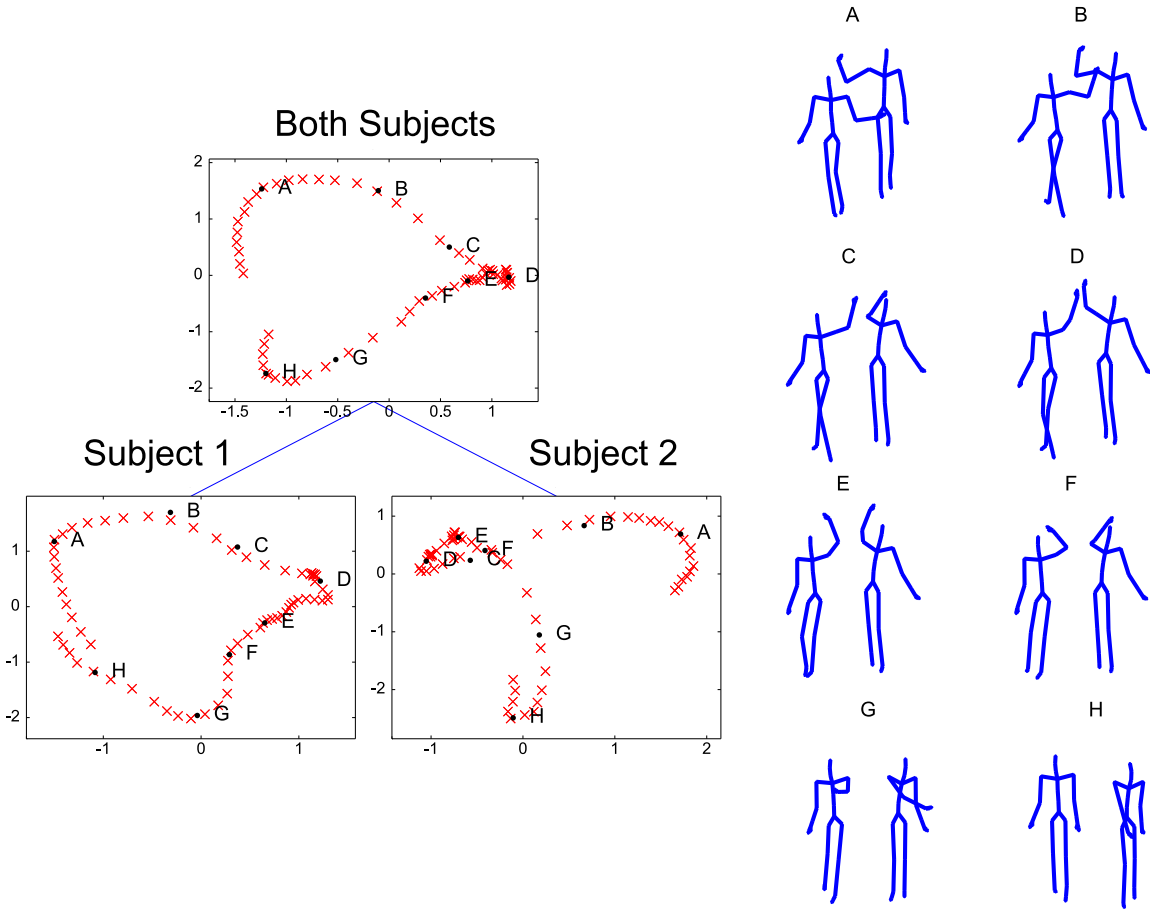


Figure : Hierarchical model of a 'high five'.

Within Subject Hierarchy

(Lawrence and Moore, 2007)

Decomposition of Body

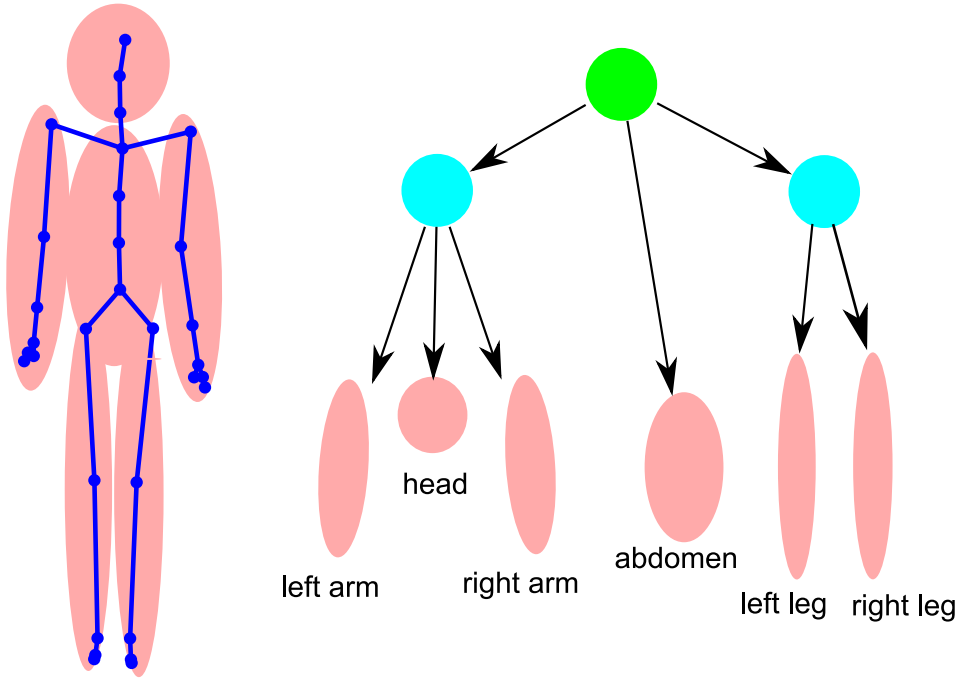


Figure : Decomposition of a subject.

Single Subject Run/Walk

(Lawrence and Moore, 2007)

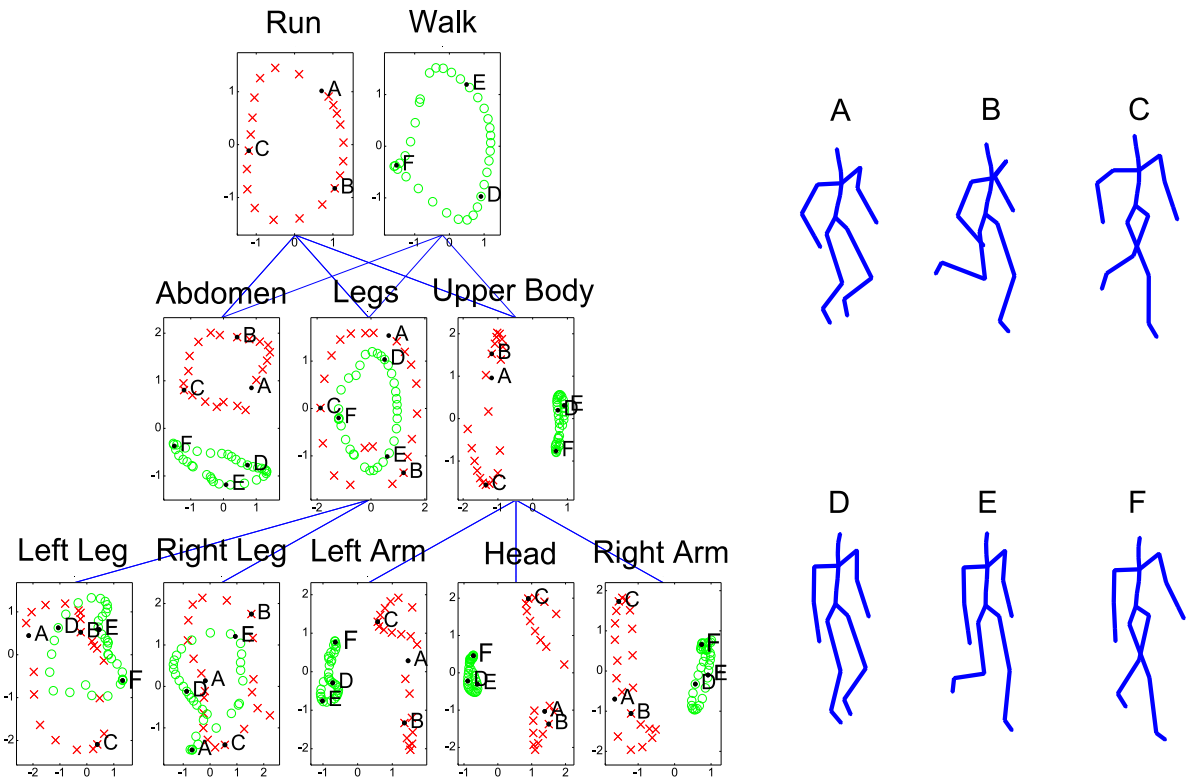


Figure : Hierarchical model of a walk and a run.

Return

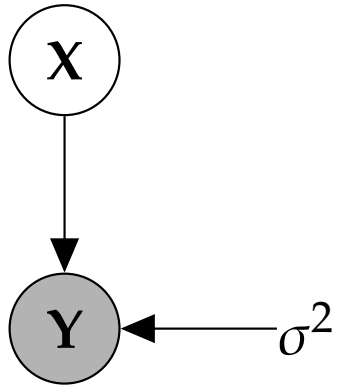
Selecting Data Dimensionality

- GP-LVM Provides probabilistic non-linear dimensionality reduction.
- How to select the dimensionality?
- Need to estimate marginal likelihood.
- In standard GP-LVM it increases with increasing q.

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.

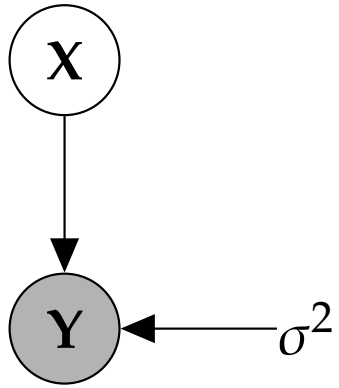


$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.

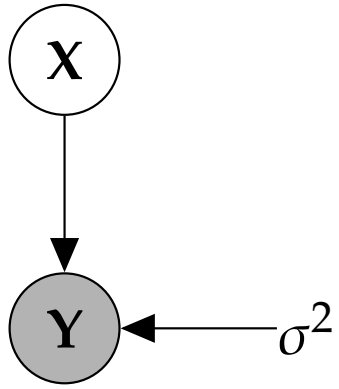


$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



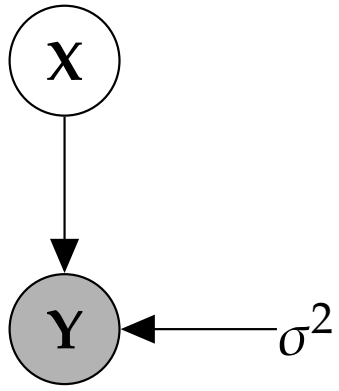
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.
  - Unfortunately integration is intractable.



$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

$$p(\mathbf{Y}|\boldsymbol{\alpha}) = ??$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \langle \log p(\mathbf{y}|\mathbf{X}) \rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \| p(\mathbf{X})\right)$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

- Extremely difficult to compute because $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is dependent on $\mathbf{X}$ and appears in the inverse.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle, \sigma^2 \mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}|\mathbf{X}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{u})\mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}$$

$$\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X})\,||\,p(\mathbf{X})\right)$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}$$

$$\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X})\,||\,p(\mathbf{X})\right)$$

- Which is analytically tractable for Gaussian $q(\mathbf{X})$ and some covariance functions.

Required Expectations

- Need expectations under $q(\mathbf{X})$ of:

$$\log c_i = \frac{1}{2\sigma^2}\left[k_{i,i} - \mathbf{k}_{i,\mathbf{u}}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{k}_{i,\mathbf{u}}\right]$$

and

$$\log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{Y})}, \sigma^2\mathbf{I}\right) = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}\right)^2$$

- This requires the expectations

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\right\rangle_{q(\mathbf{X})}$$

and

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right\rangle_{q(\mathbf{X})}$$

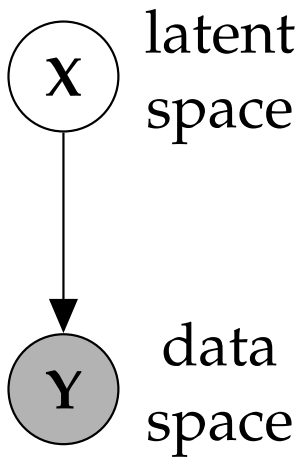
which can be computed analytically for some covariance functions.

Priors for Latent Space

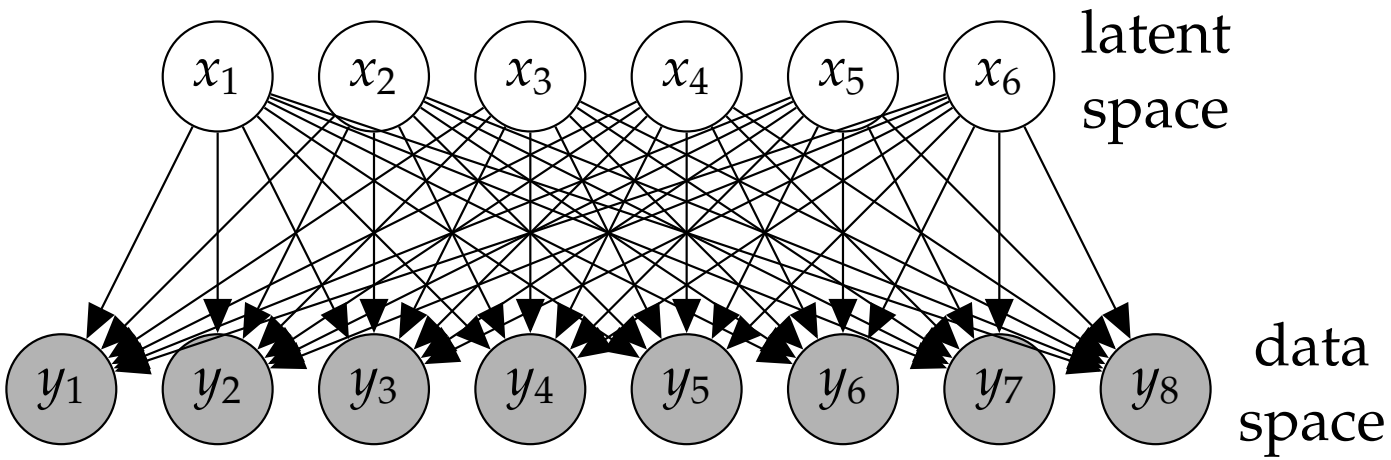
Titsias and Lawrence (2010)

- Variational marginalization of $\mathbf{X}$ allows us to learn parameters of $p(\mathbf{X})$.
- Standard GP-LVM where $\mathbf{X}$ learnt by MAP, this is not possible (see e.g. Wang et al., 2008).
- First example: learn the dimensionality of latent space.

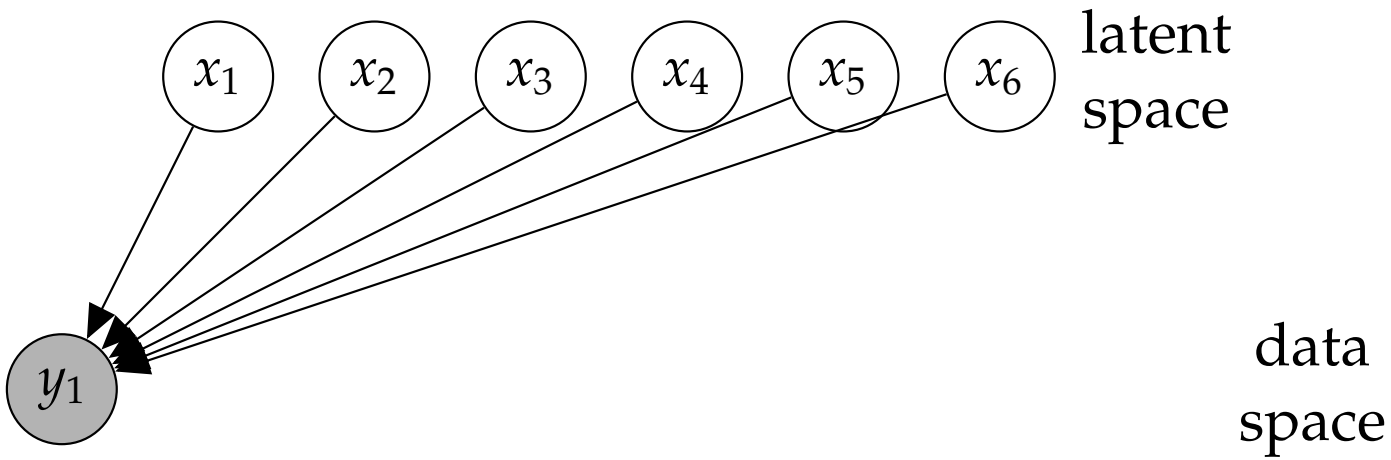
Graphical Representations of GP-LVM



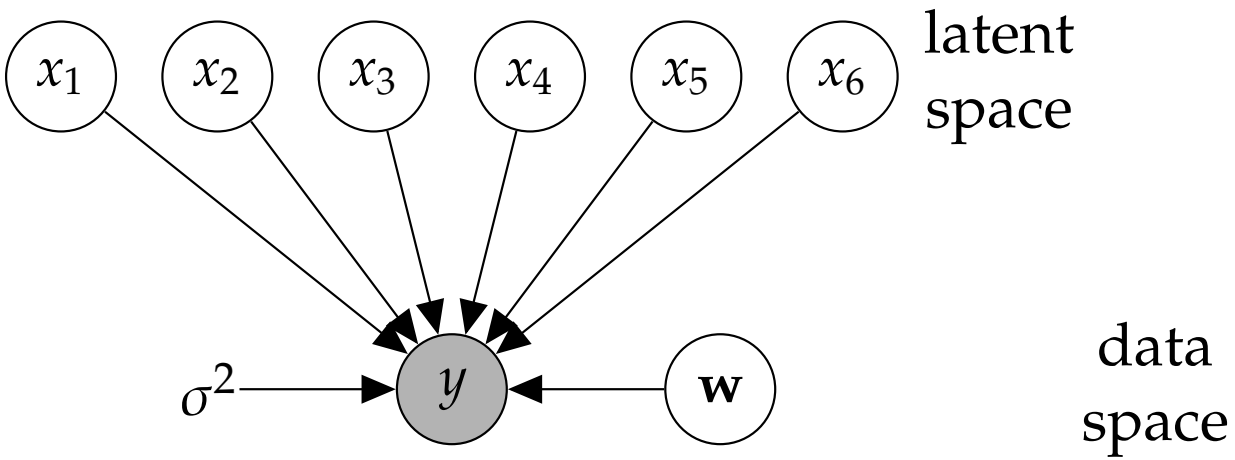
Graphical Representations of GP-LVM



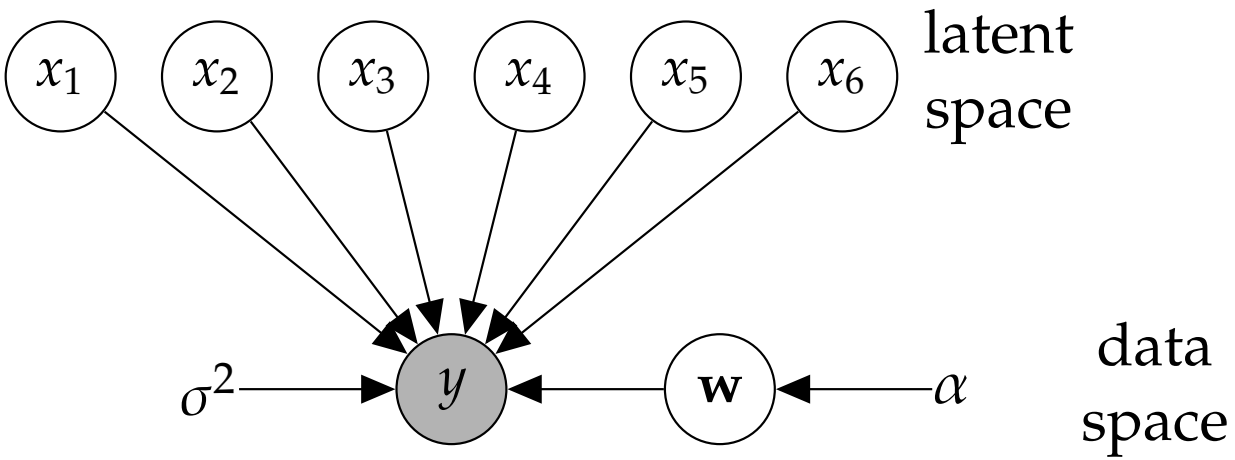
Graphical Representations of GP-LVM



Graphical Representations of GP-LVM



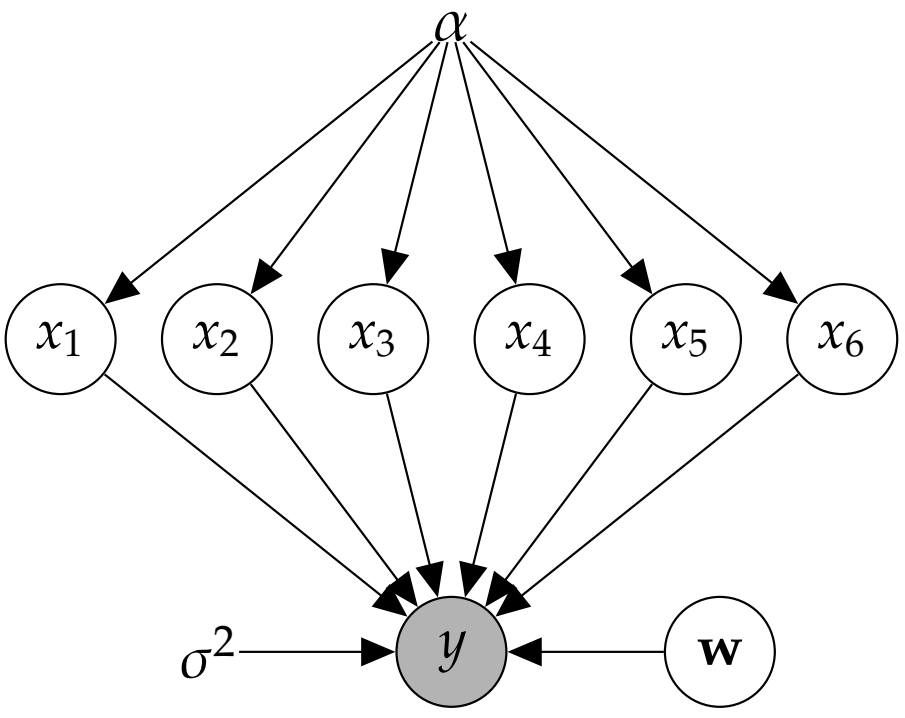
Graphical Representations of GP-LVM



$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I}) \quad \mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$$

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

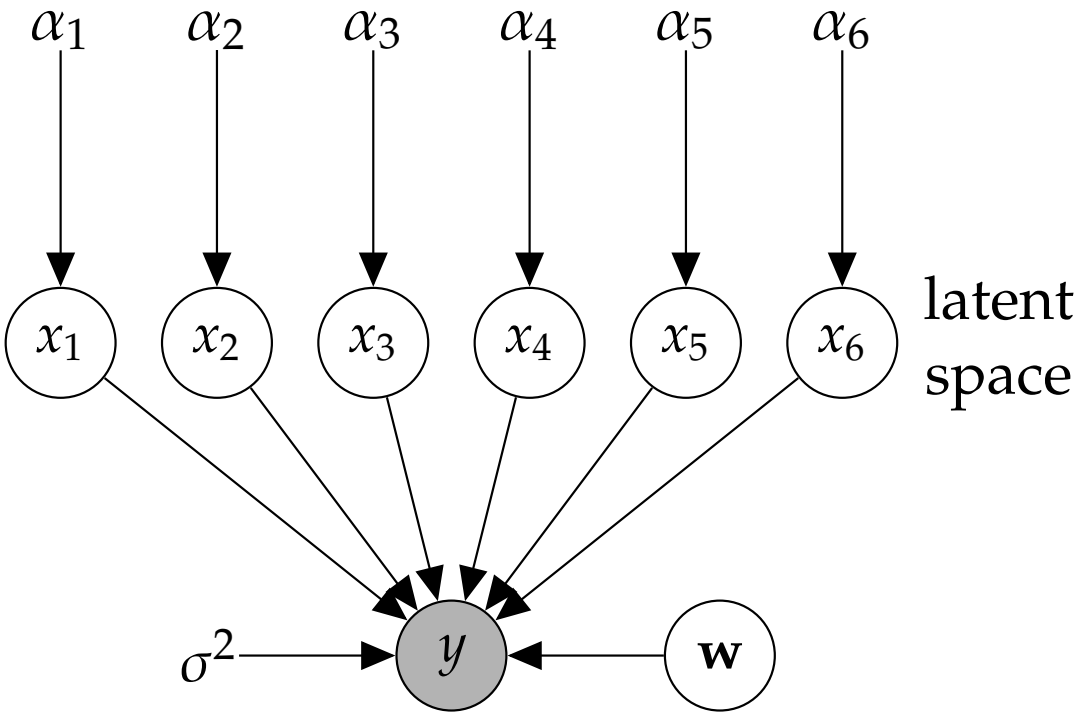
Graphical Representations of GP-LVM



$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathbf{x} \sim \mathcal{N}(0, \alpha\mathbf{I})$$

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

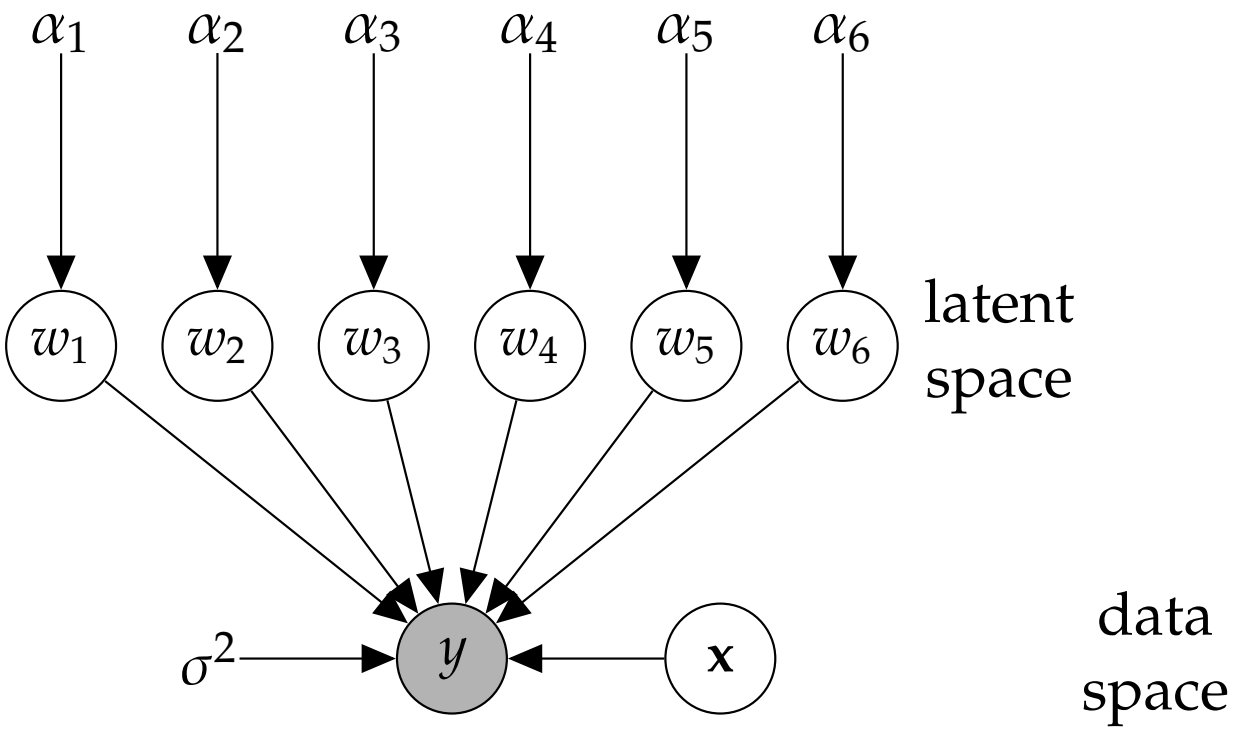
Graphical Representations of GP-LVM



$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad x_i \sim \mathcal{N}(0, \alpha_i)$$

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

Graphical Representations of GP-LVM



$$w_i \sim \mathcal{N}(0, \alpha_i) \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

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

Non-linear $f(\mathbf{x})$

- In linear case equivalence because $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$

$$p(w_i) \sim \mathcal{N}(\mathbf{0}, \alpha_i)$$

- In non linear case, need to scale columns of $\mathbf{X}$ in prior for $f(\mathbf{x})$.
- This implies scaling columns of $\mathbf{X}$ in covariance function

$$k(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}) = \exp\left(-\frac{1}{2}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})^\top \mathbf{A}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})\right)$$

$\mathbf{A}$ is diagonal with elements α_i^2. Now keep prior spherical

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \mathbf{I}\right)$$

- Covariance functions of this type are known as ARD (see e.g. Neal, 1996; MacKay, 2003; Rasmussen and Williams, 2006).

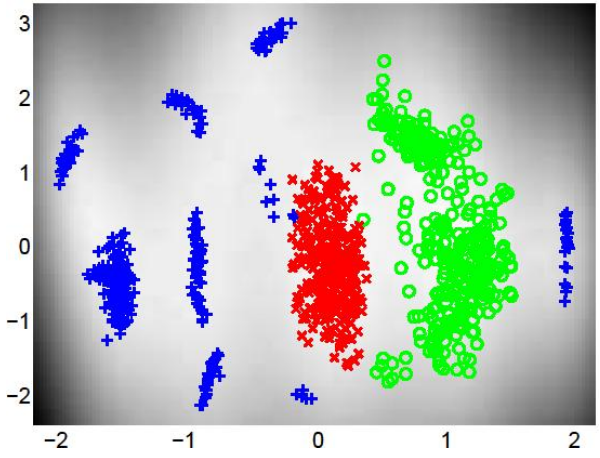
Automatic dimensionality detection

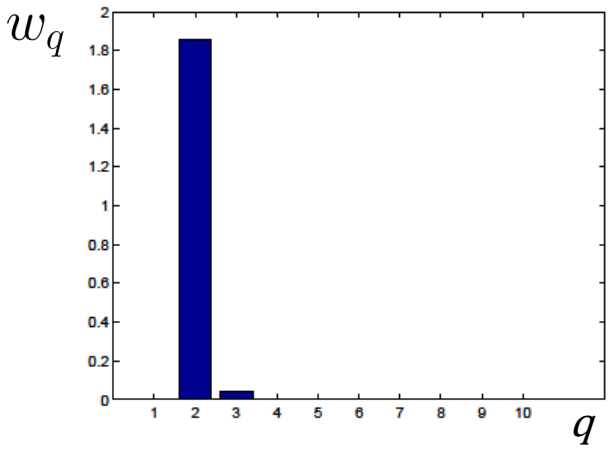
- Achieved by employing an *Automatic Relevance Determination (ARD)* covariance function for the prior on the GP mapping

- $f \sim GP(\mathbf{0}, k_f)$ with

$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{i,q} - x_{j,q})^2\right)$$

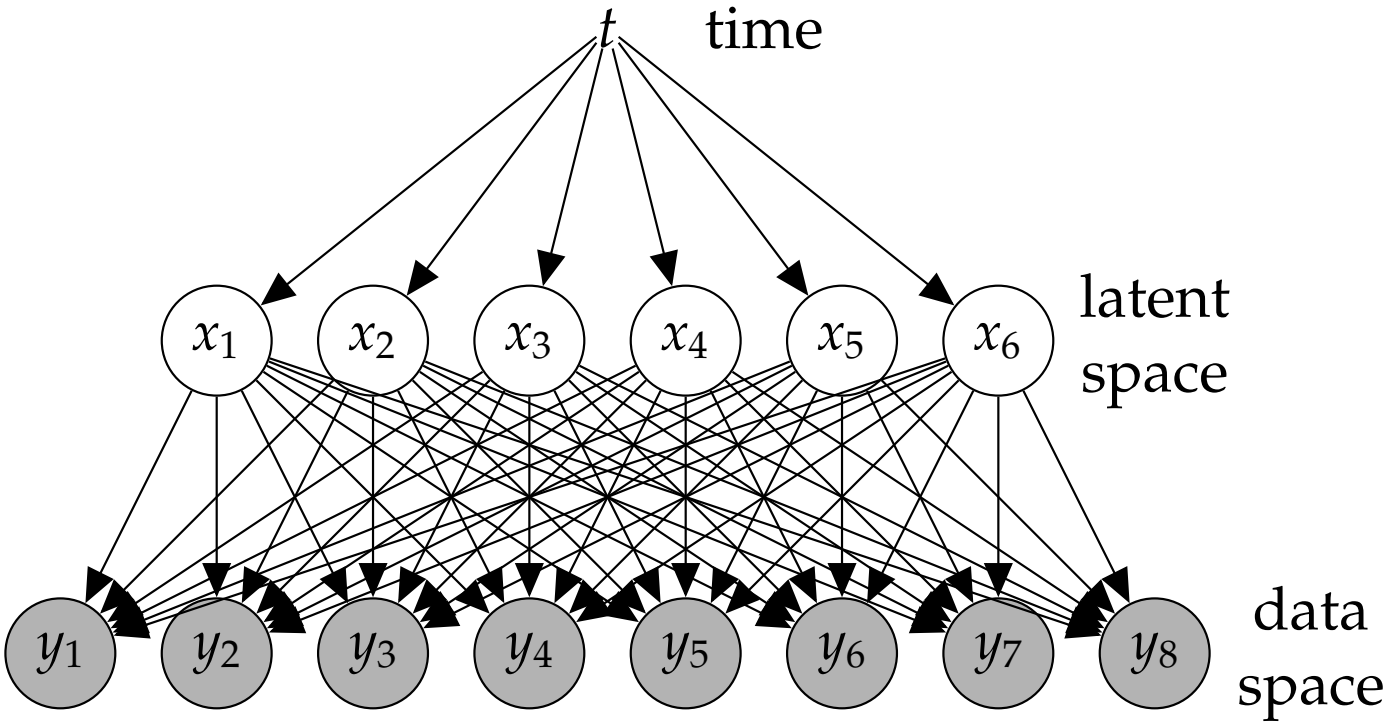
- Example





Gaussian Process Dynamical Systems

(Damianou et al., 2011)



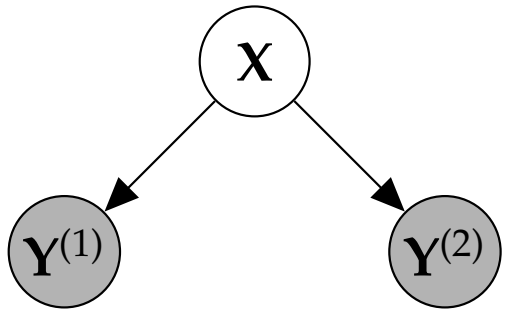
Gaussian Process over Latent Space

- Assume a GP prior for $p(\mathbf{X})$.
- Input to the process is time, $p(\mathbf{X}|t)$.

Interpolation of HD Video

Modeling Multiple 'Views'

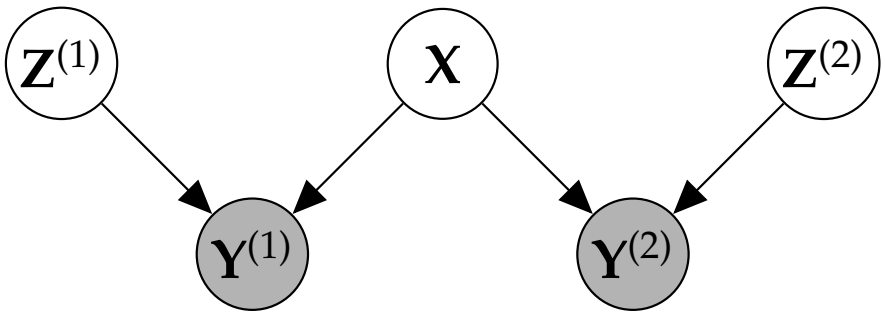
- Single space to model correlations between two different data sources, e.g., images & text, image & pose.
- Shared latent spaces: (Shon et al., 2006; Navaratnam et al., 2007; Ek et al., 2008b)



- Effective when the 'views' are correlated.
- But not all information is shared between both 'views'.
- PCA applied to concatenated data vs CCA applied to data.

Shared-Private Factorization

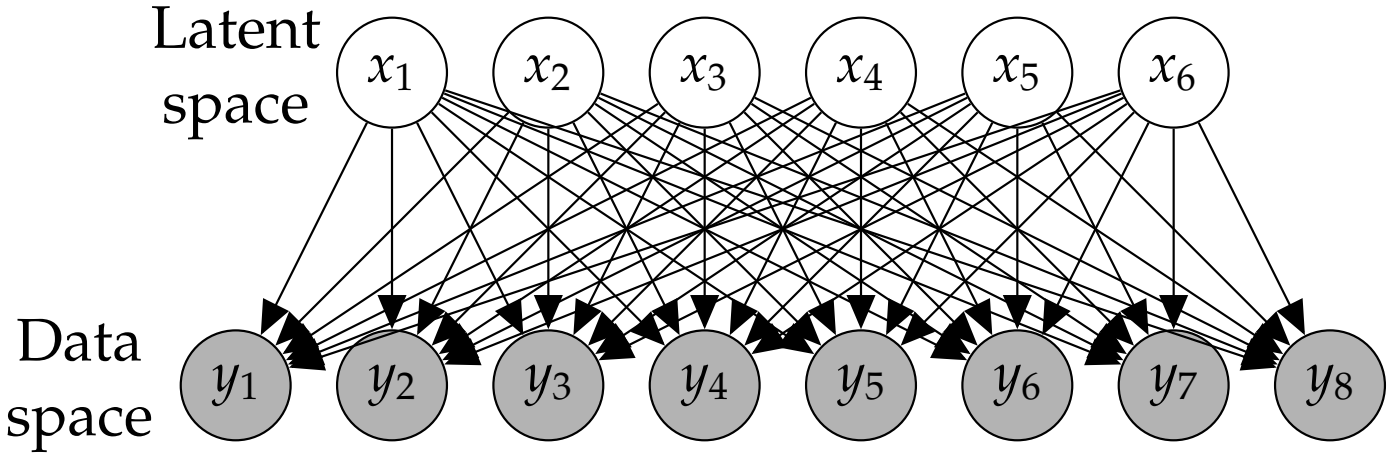
- In real scenarios, the 'views' are neither fully independent, nor fully correlated.
- Shared models
  - either allow information relevant to a single view to be mixed in the shared signal,
  - or are unable to model such private information.
- Solution: Model shared and private information (Virtanen et al., 2011; Ek et al., 2008a; Leen and Fyfe, 2006; Klami and Kaski, 2007, 2008; Tucker, 1958)



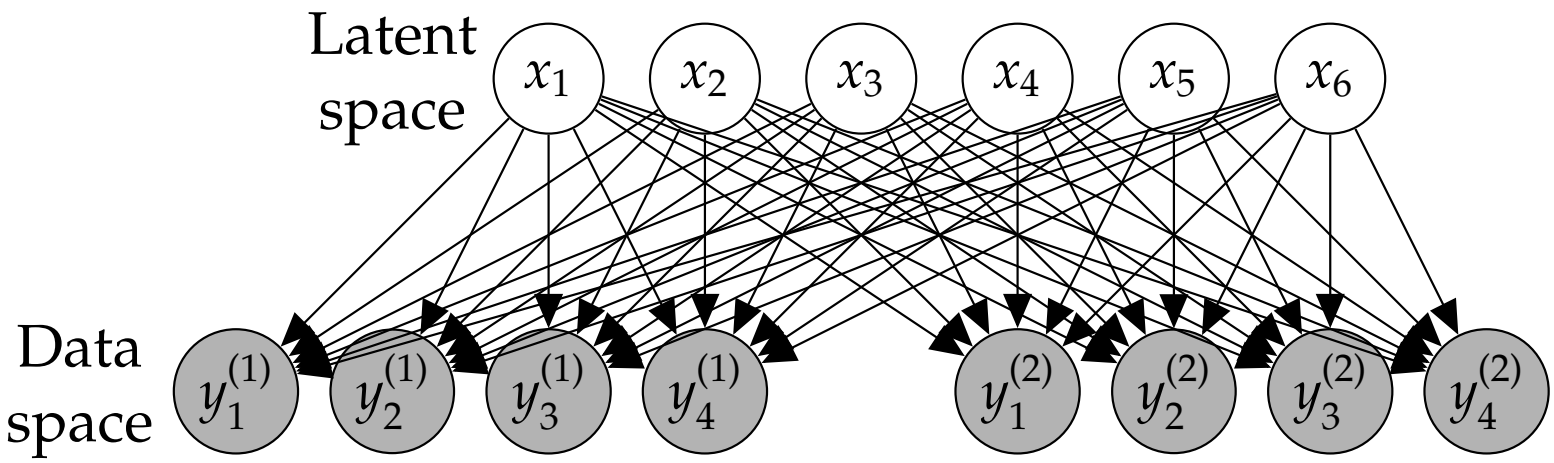
- Probabilistic CCA is case when dimensionality of $\mathbf{Z}$ matches $\mathbf{Y}^{(i)}$ (cf Inter Battery Factor Analysis (Tucker, 1958)).

Manifold Relevance Determination

Damianou et al. (2012)



Shared GP-LVM



Separate ARD parameters for mappings to $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$.

Example: Yale faces

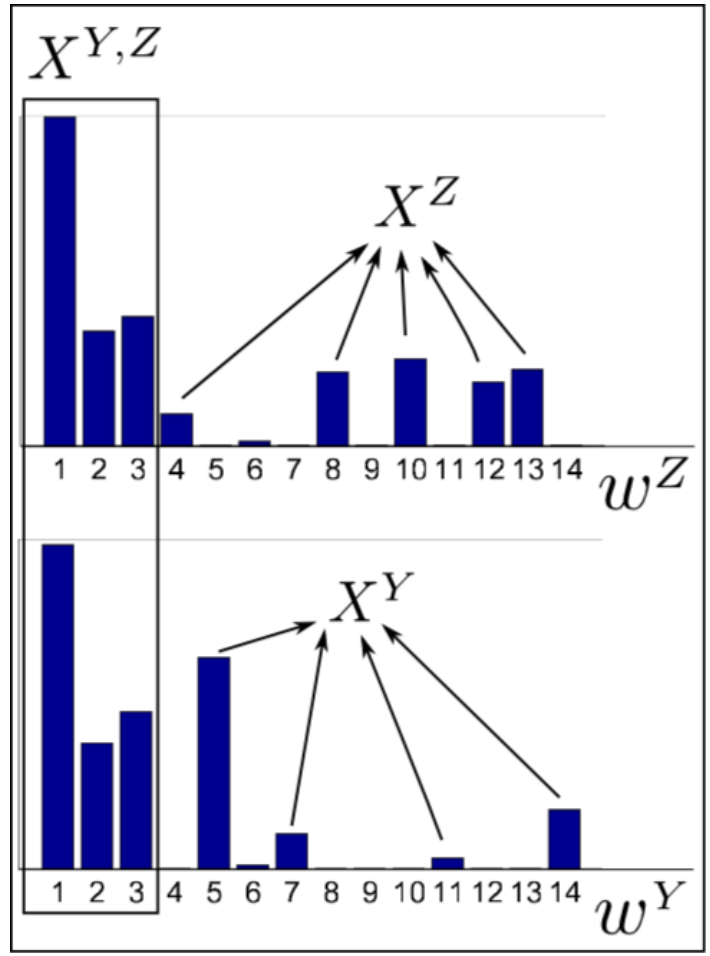
- Dataset Y: 3 persons under all illumination conditions
- Dataset Z: As above for 3 different persons
- Align datapoints $\mathbf{x}_n$ and $\mathbf{z}_n$ only based on the lighting direction

Results

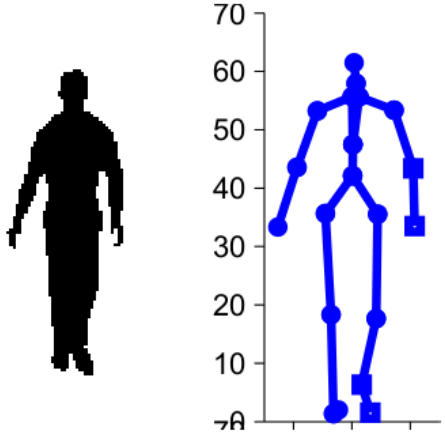
- Latent space X initialised with 14 dimensions

- Weights define a segmentation of X

- Video / demo…



Potential applications..?



Manifold Relevance Determination

Latent Force Models

Neil D. Lawrence

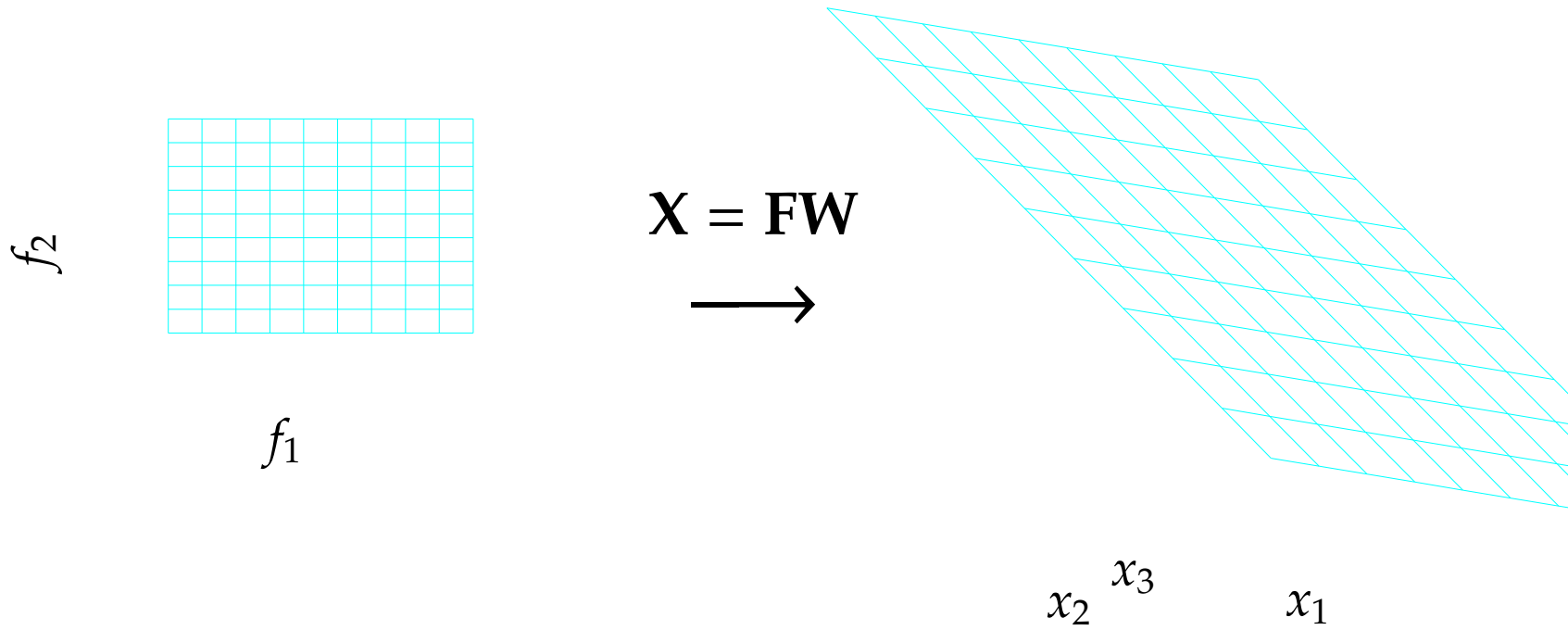
GPRS
25th–27th February 2015

Outline

Outline

Linear Dimensionality Reduction

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



Dimensionality Reduction

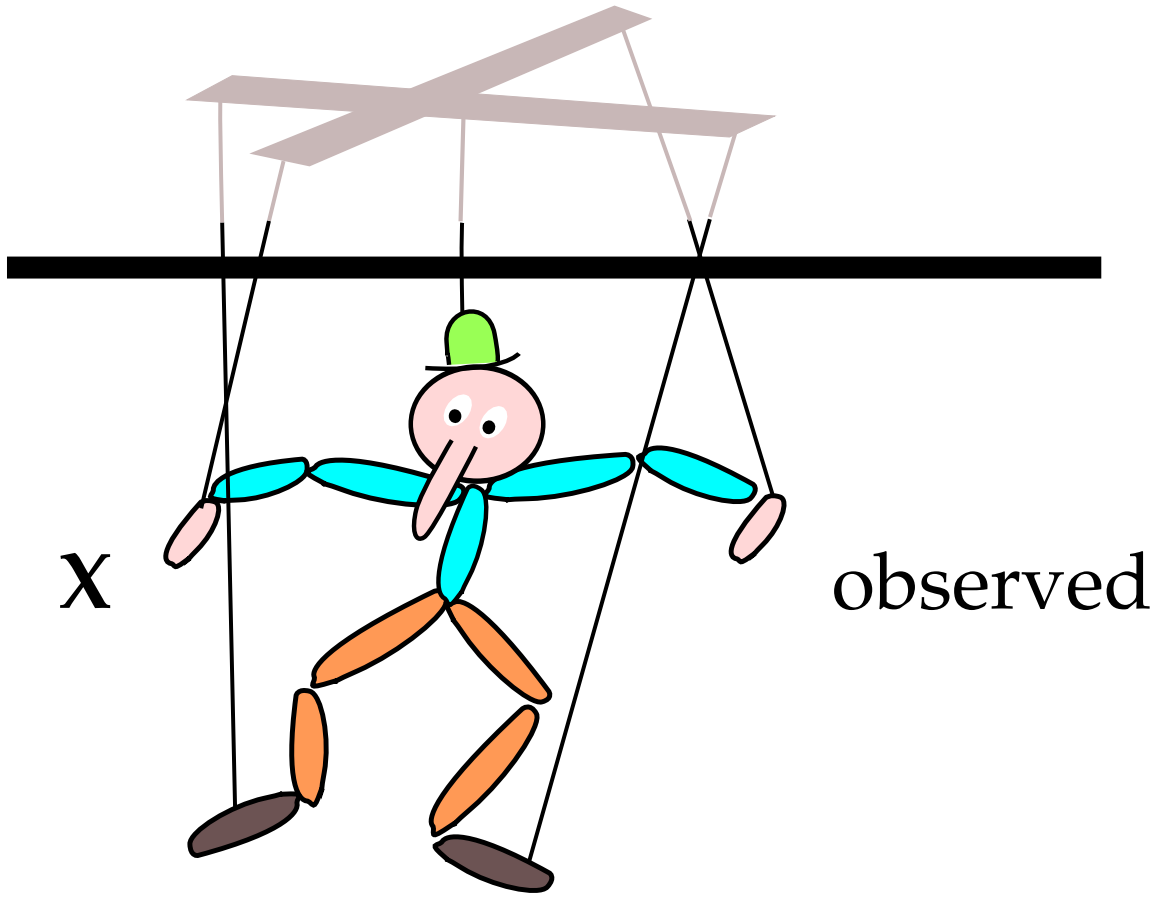
- Linear relationship between the data, $\mathbf{X}$, and a reduced dimensional representation, $\mathbf{F}$.

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

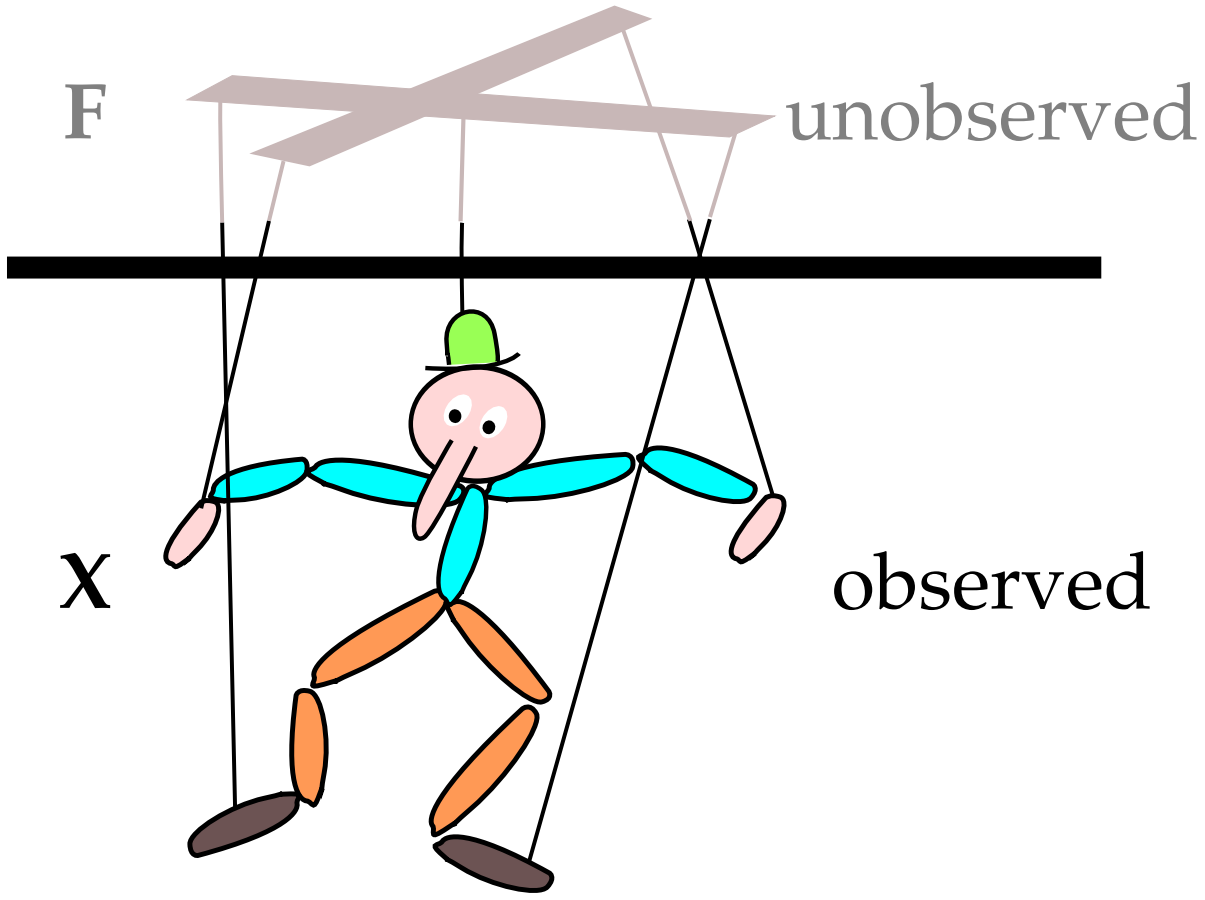
$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

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

Marionette Analogy



Marionette Analogy



$\mathbf{F}$ 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 Σ constrained to be diagonal, we have factor analysis.

Dimensionality Reduction: Temporal Data

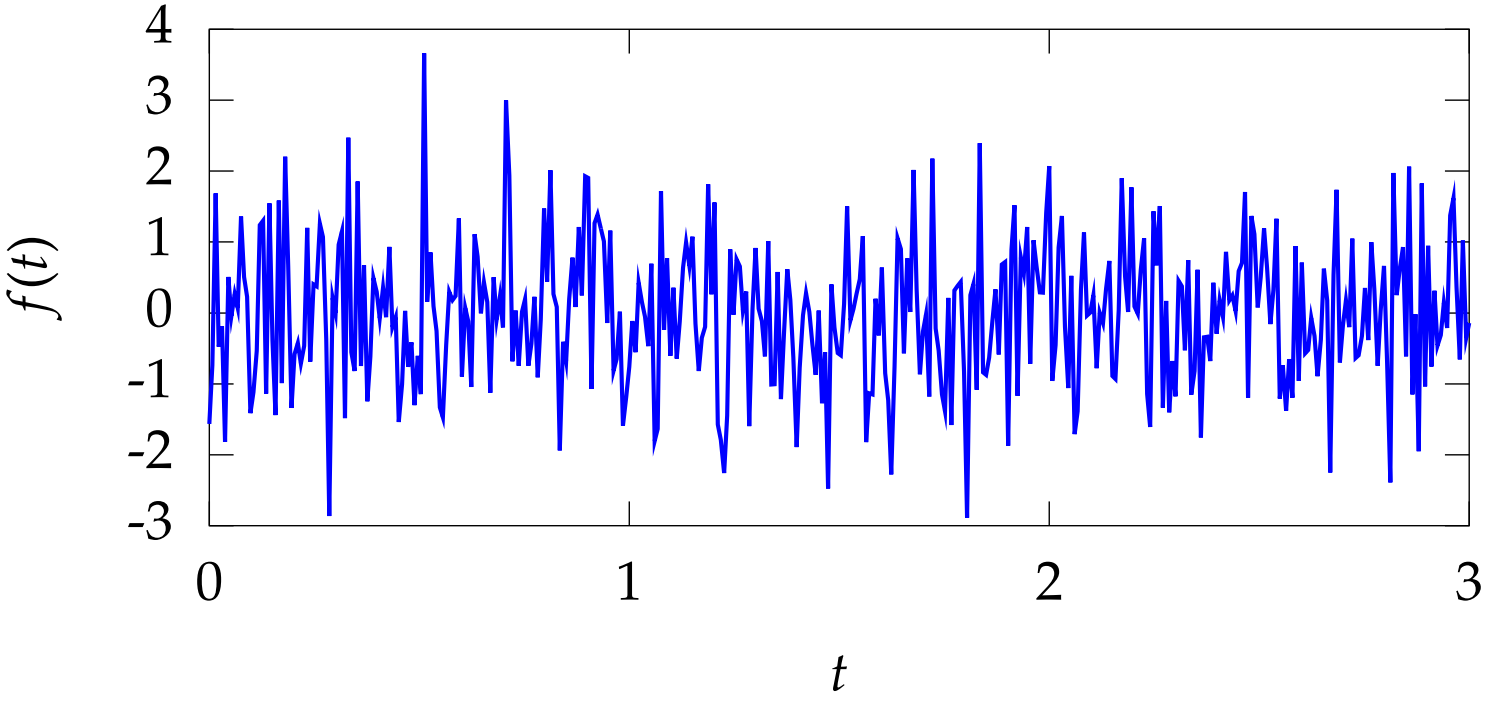


Figure : PCA: Pure sampling from a Gaussian does not retain temporal effects.

Dimensionality Reduction: Temporal Data

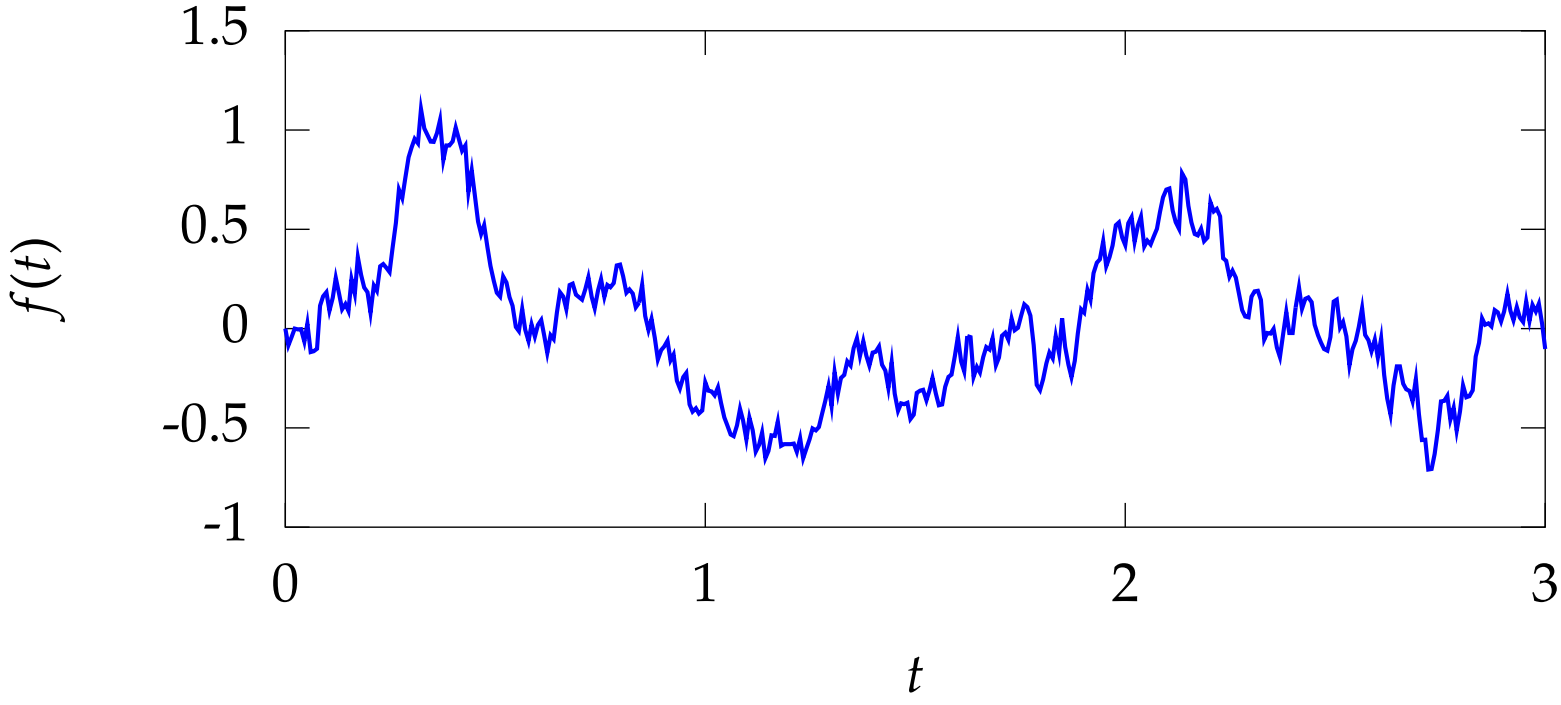


Figure : Kalman filter (Rauch-Tung-Striebel smoother) is Markov-Gaussian (non smooth).

Dimensionality Reduction: Temporal Data

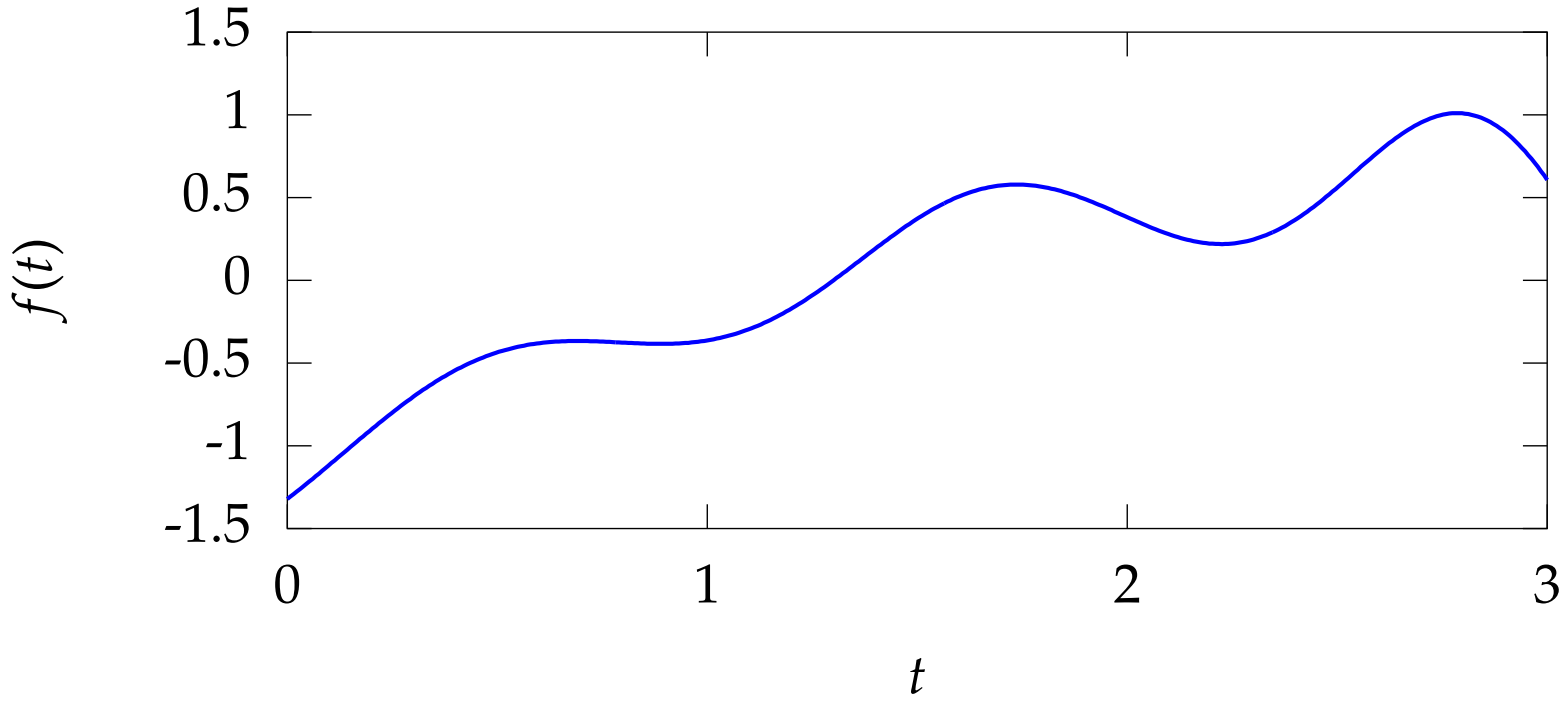


Figure : General Gaussian processes allow for priors over *smooth* functions.

Mechanical Analogy

Back to Mechanistic Models!

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

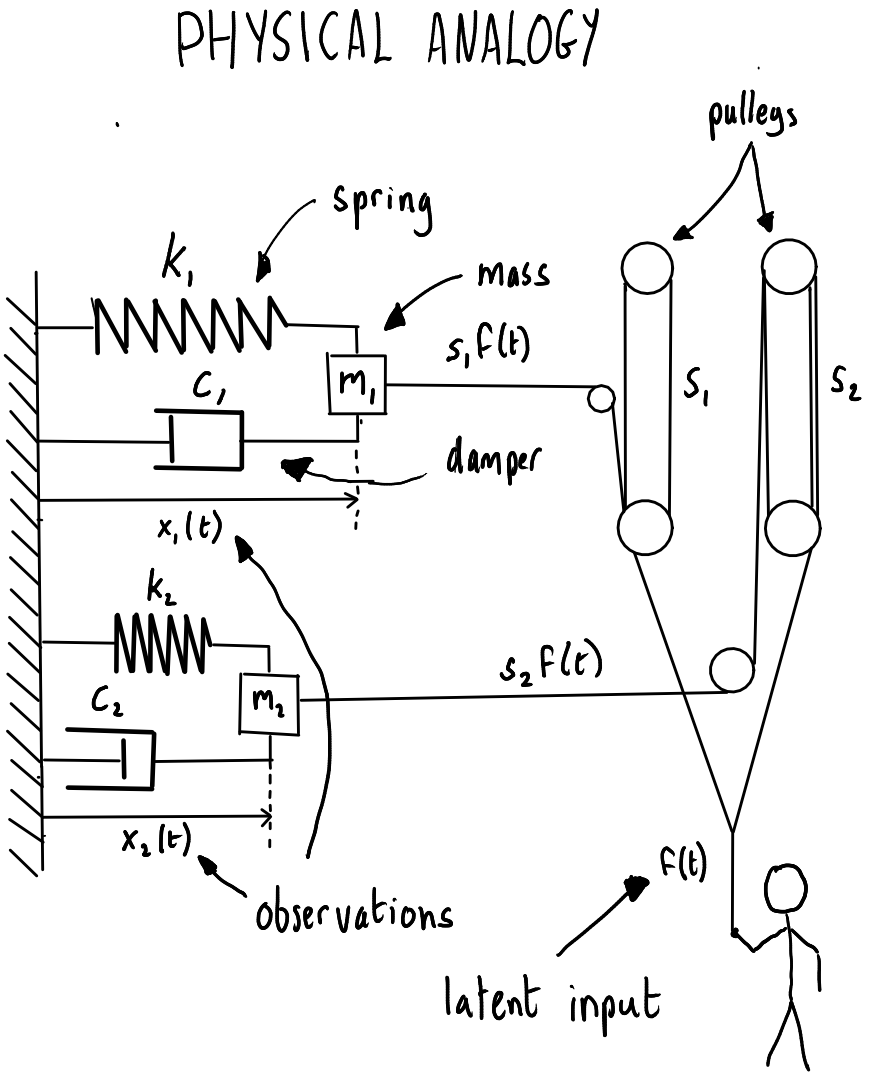
Extend Model

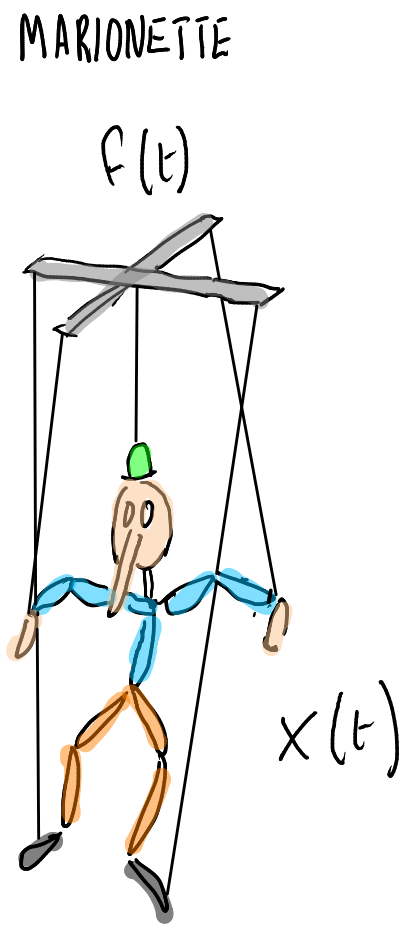
- Add a damper and give the system mass.

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

- 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





Gaussian Process priors and Latent Force Models

Driven Harmonic Oscillator

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

$$m_k \ddot{x}_k(t) + c_k \dot{x}_k(t) + d_k x_k(t) = b_k + \sum_{i=0}^{q} s_{ik} f_i(t), \tag{4}$$

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

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

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

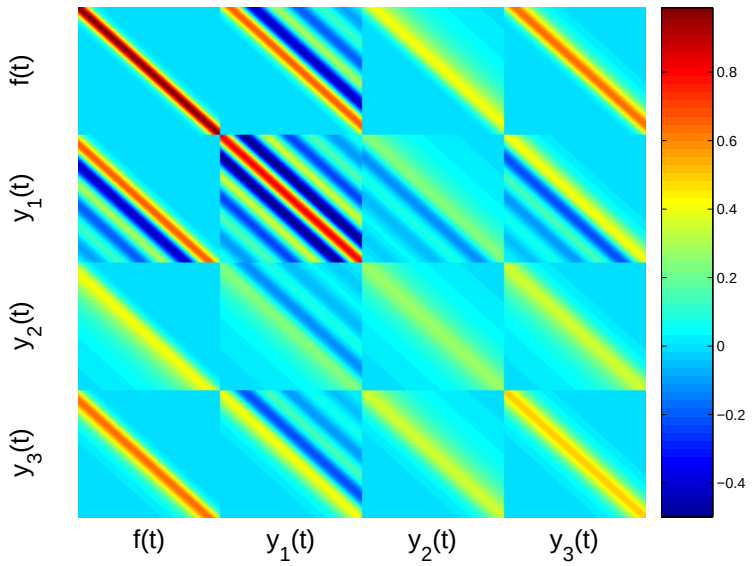
Covariance for ODE Model

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

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



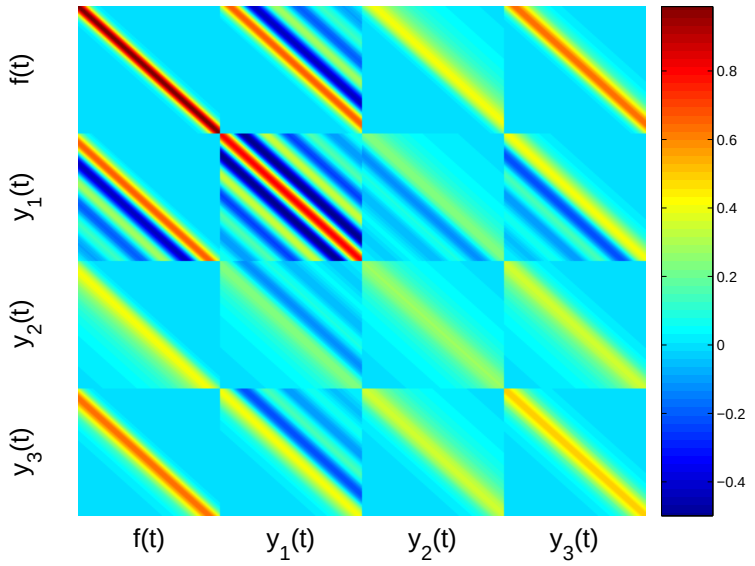
Covariance for ODE Model

- Analogy

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



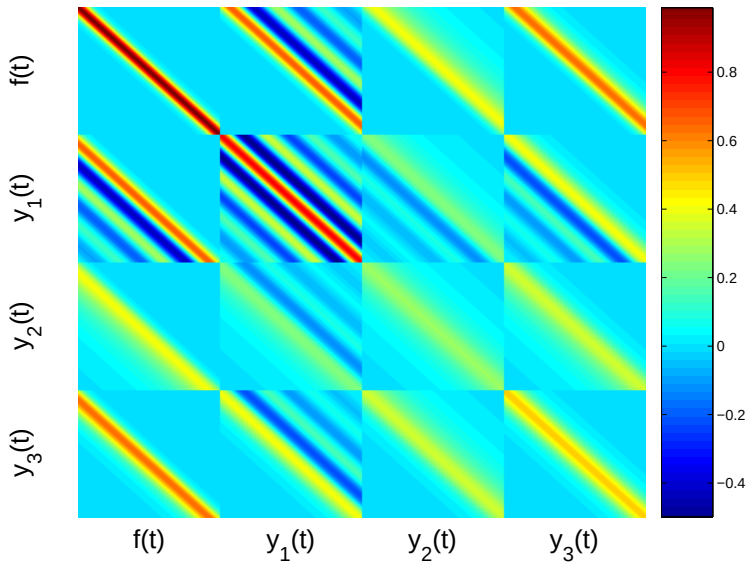
Covariance for ODE Model

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

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



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

- `lfmSample`

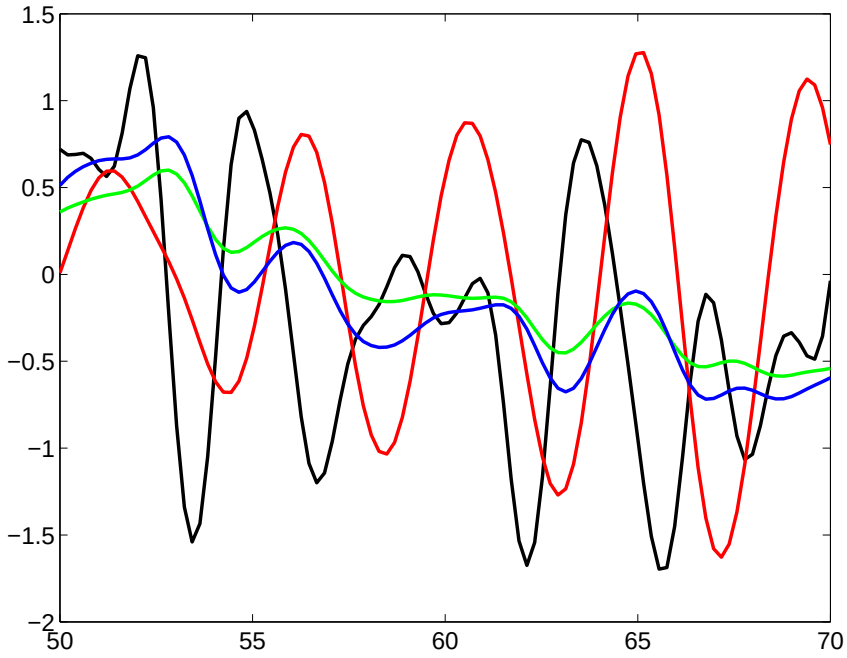


Figure : Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

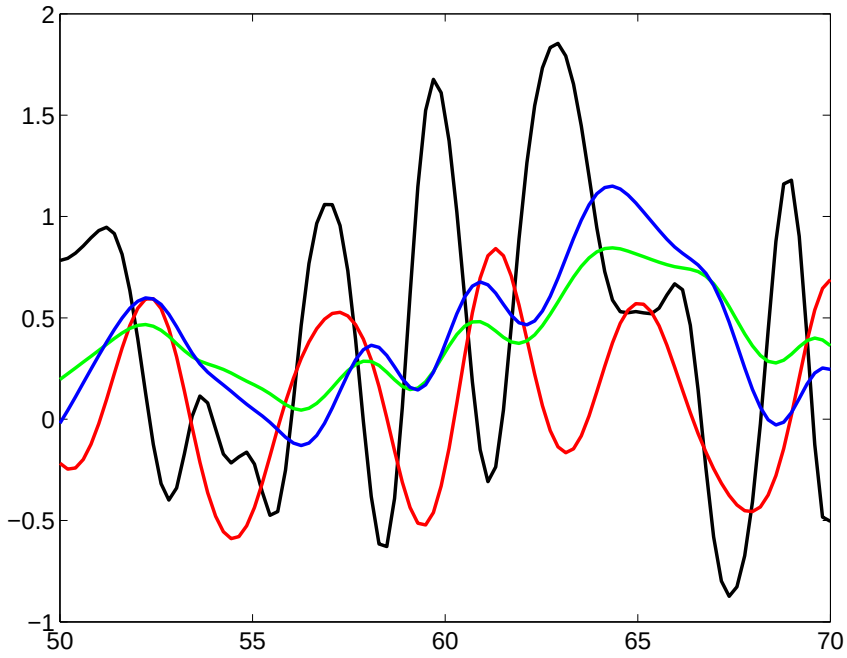


Figure : Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

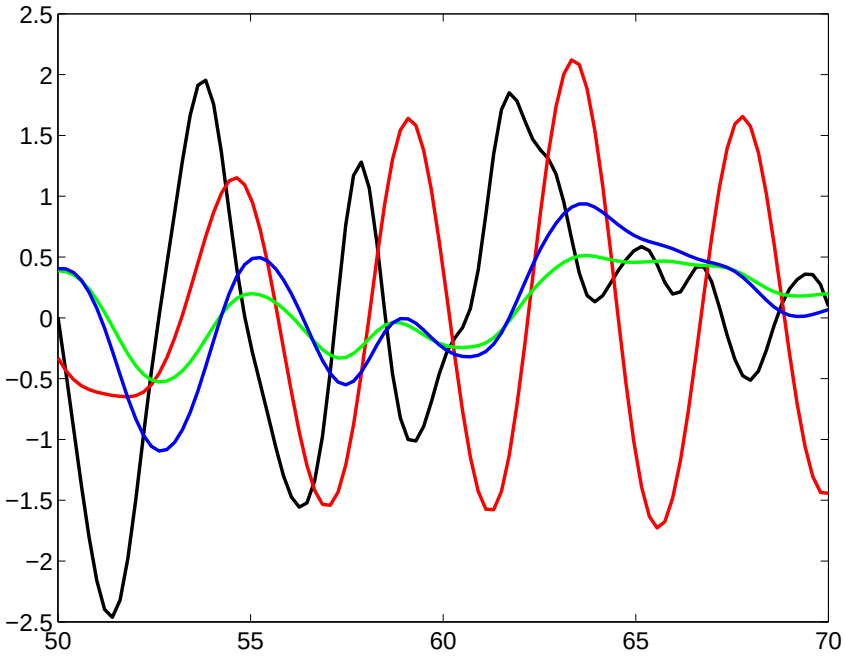


Figure : Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

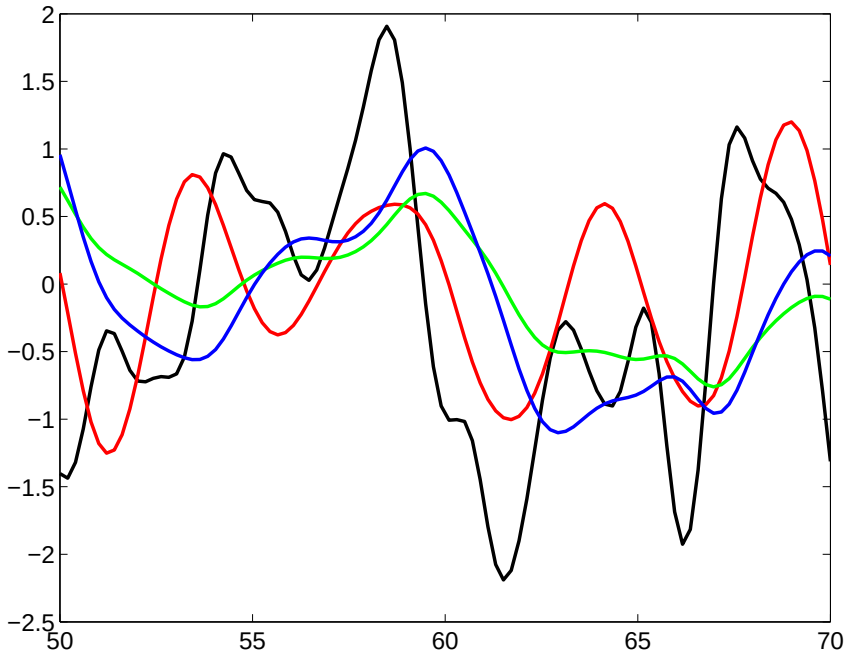


Figure : Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

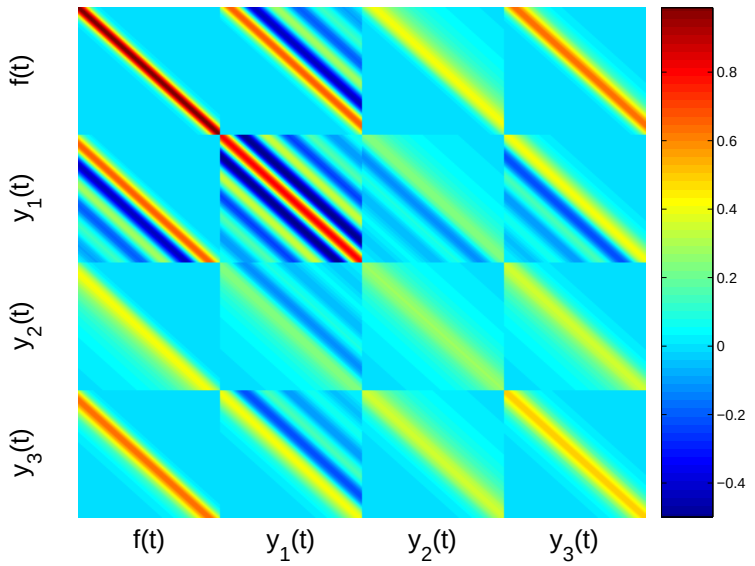
Covariance for ODE

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

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



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

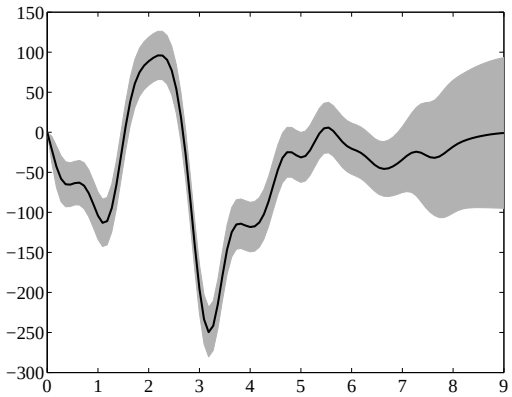
- Model left arm only.
- 3 balancing motions (18, 19, 20) from subject 49.
- 18 and 19 are similar, 20 contains more dramatic movements.
- Train on 18 and 19 and testing on 20
- Data was down-sampled by 32 (from 120 fps).
- Reconstruct motion of left arm for 20 given other movements.
- Compare with GP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

Mocap Results

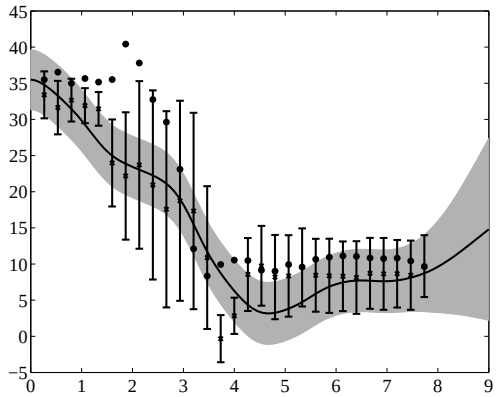
Table : Root mean squared (RMS) angle error for prediction of the left arm's configuration in the motion capture data. Prediction with the latent force model outperforms the prediction with regression for all apart from the radius's angle.

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

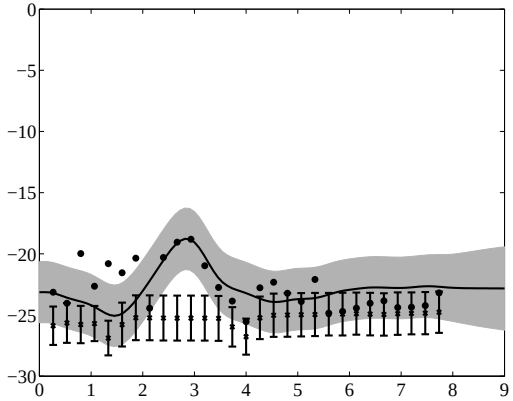
Mocap Results II



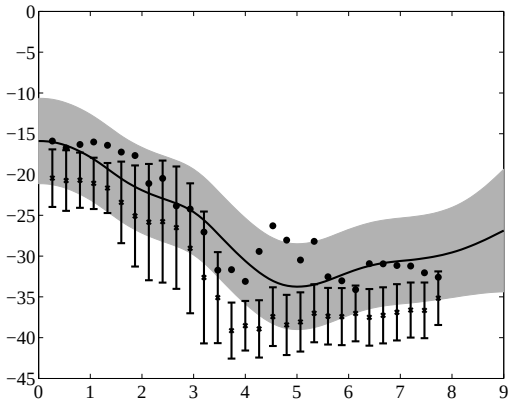
(a) Inferred Latent Force



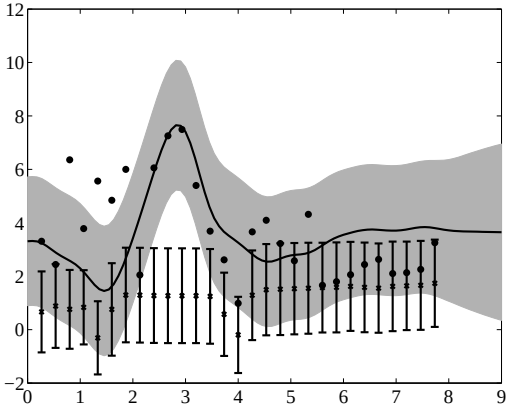
(b) Wrist



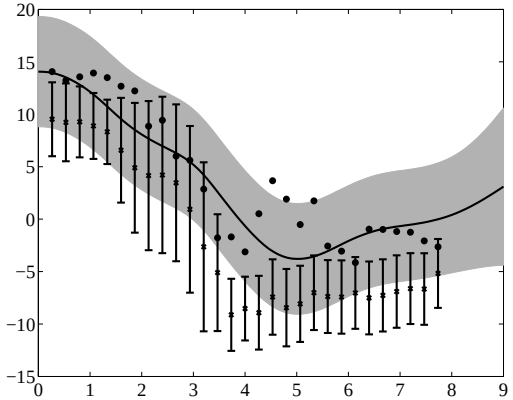
(c) Hand X Rotation



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

Motion Capture Results

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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?

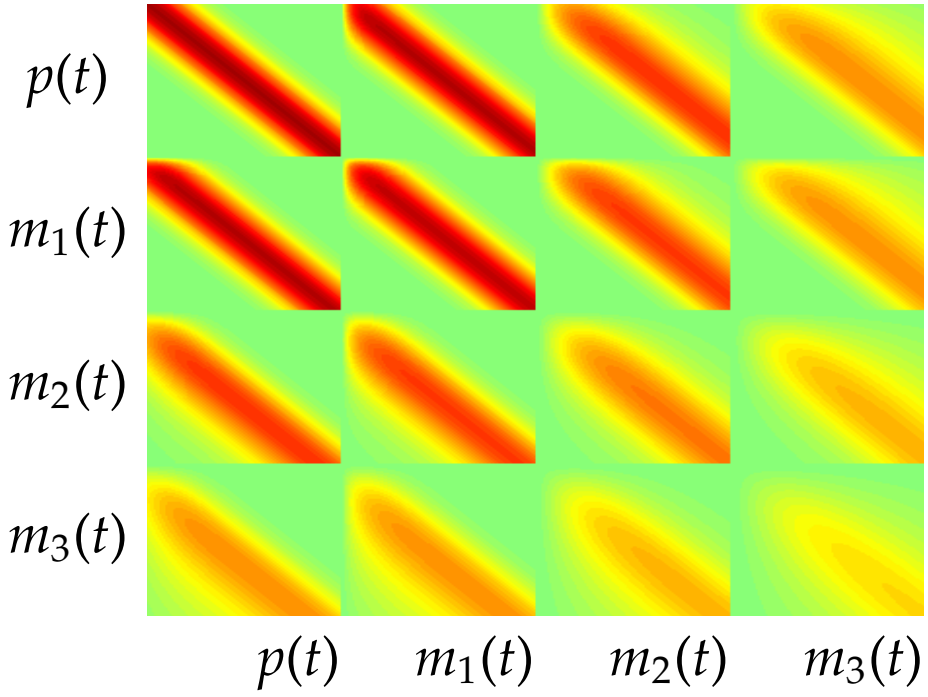
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



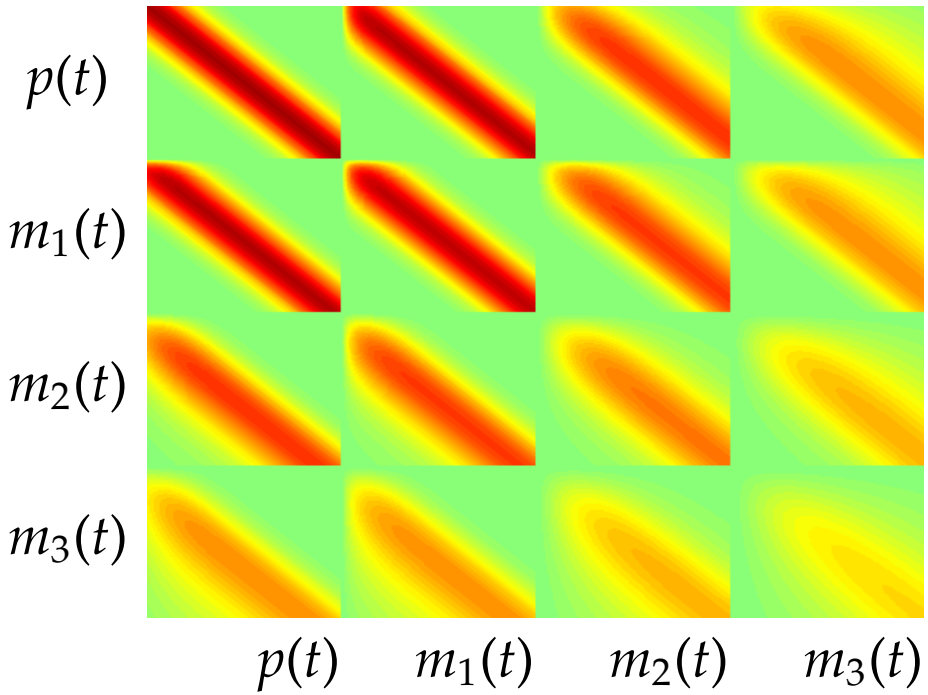
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



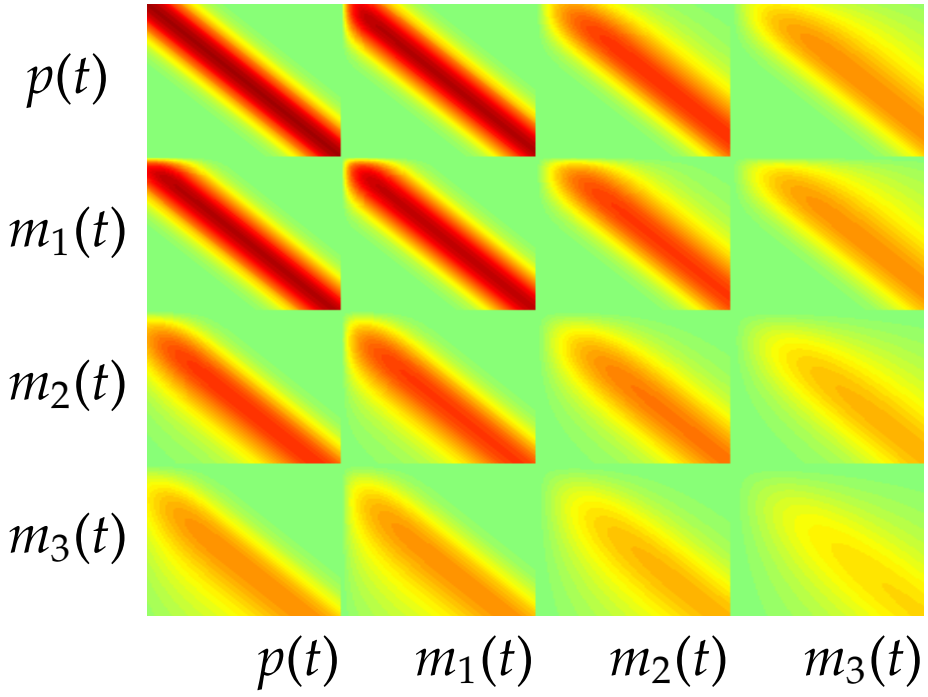
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



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

- `simSample`

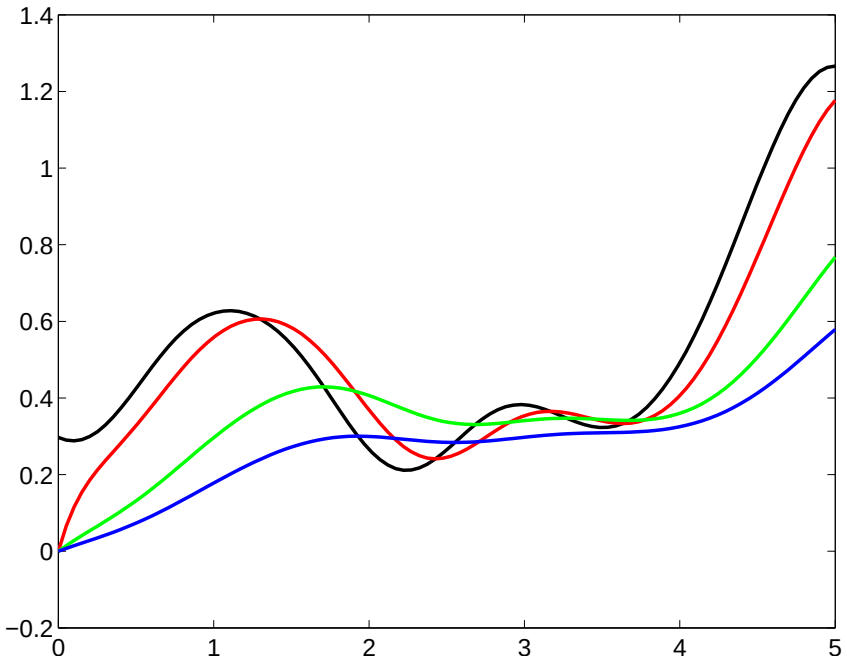


Figure : Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

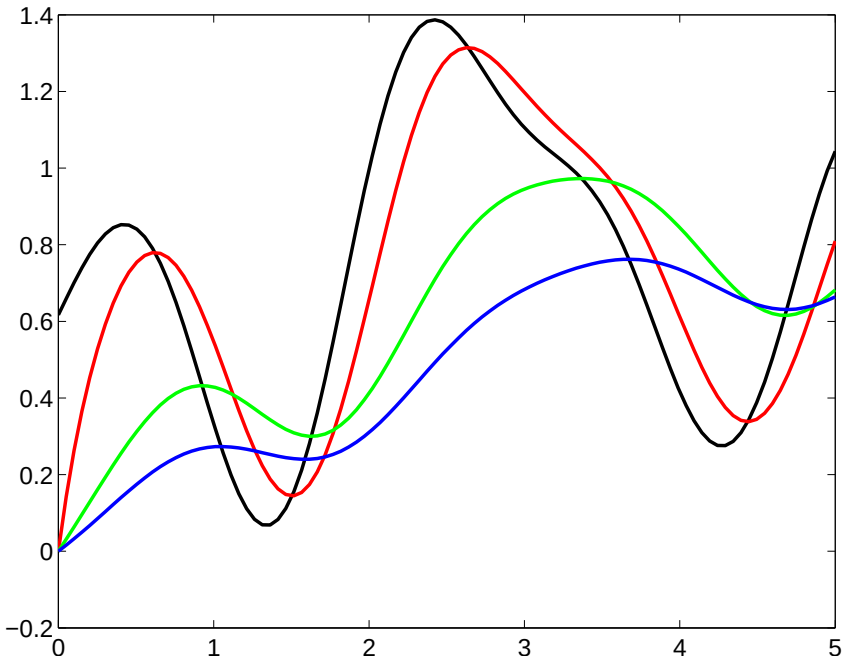


Figure : Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

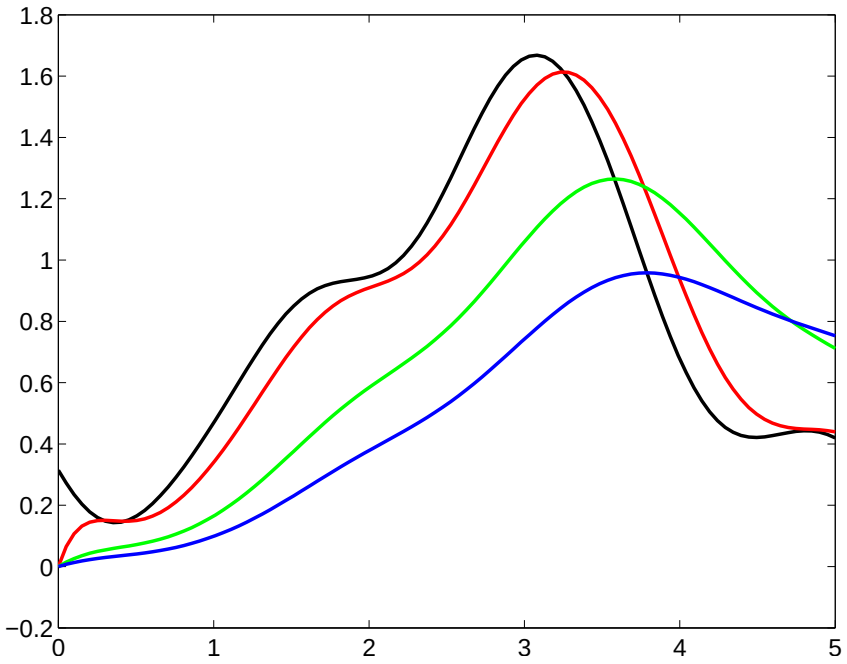


Figure : Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

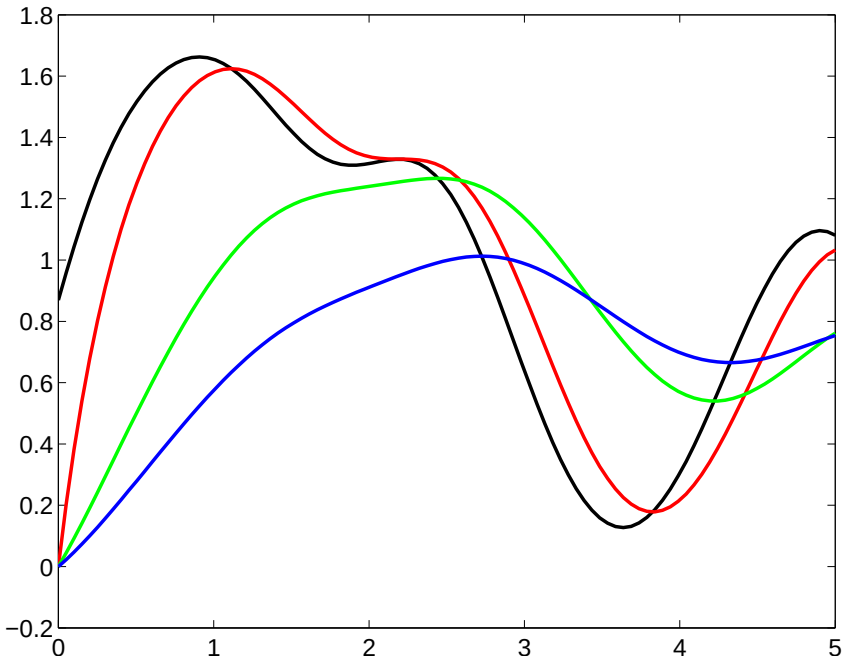
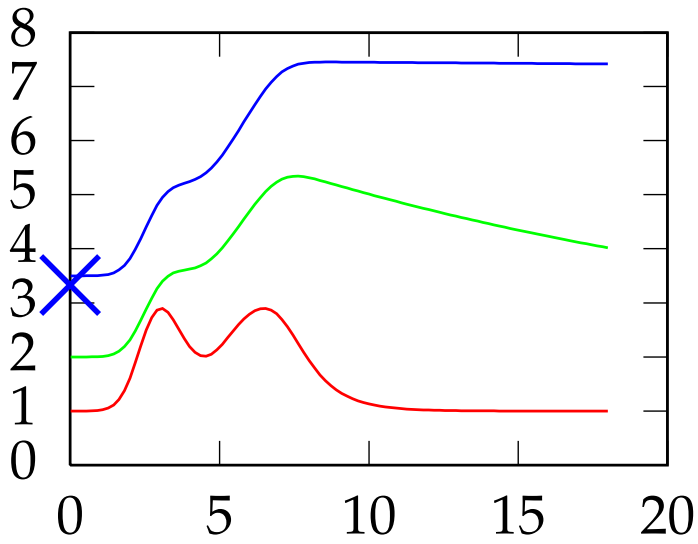
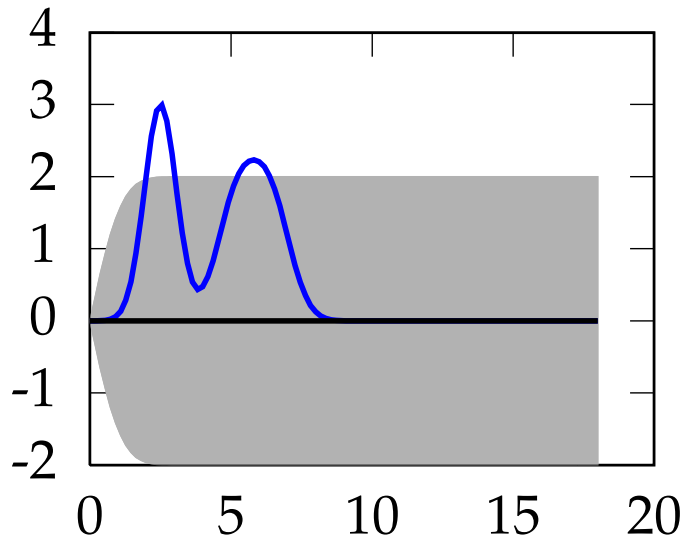
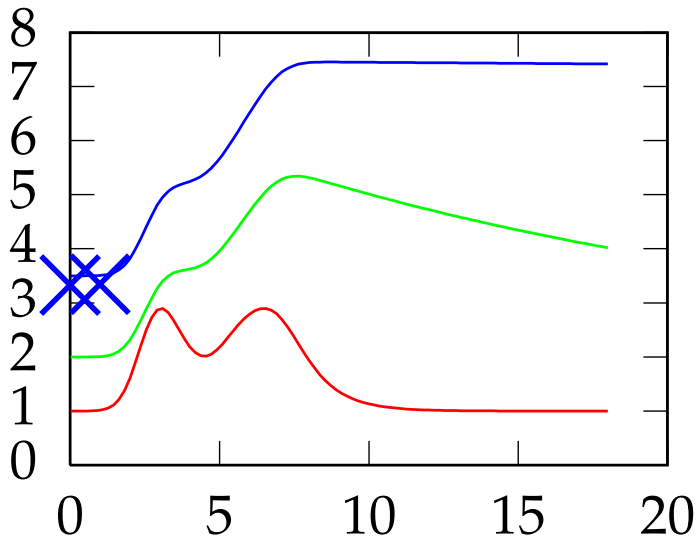
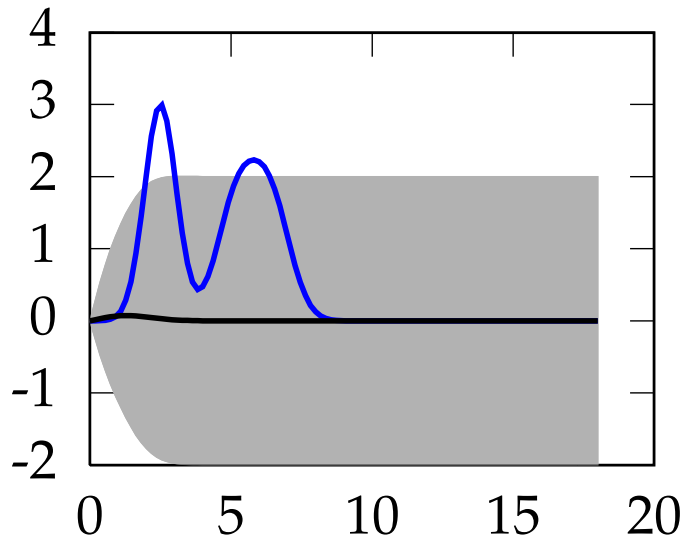


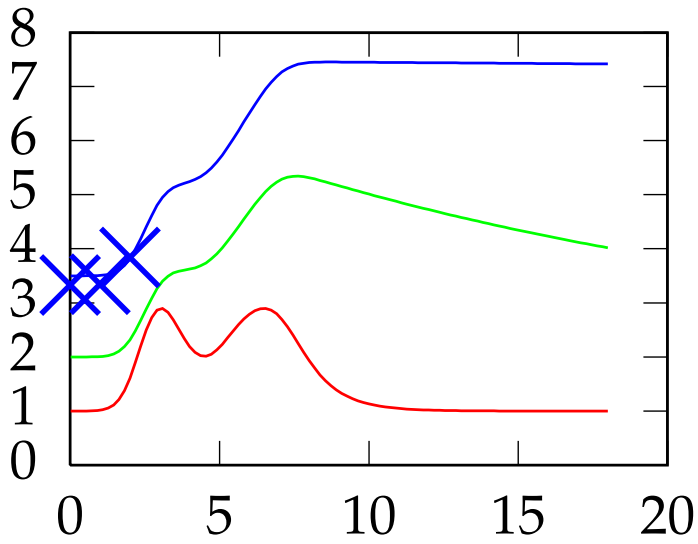
Figure : Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

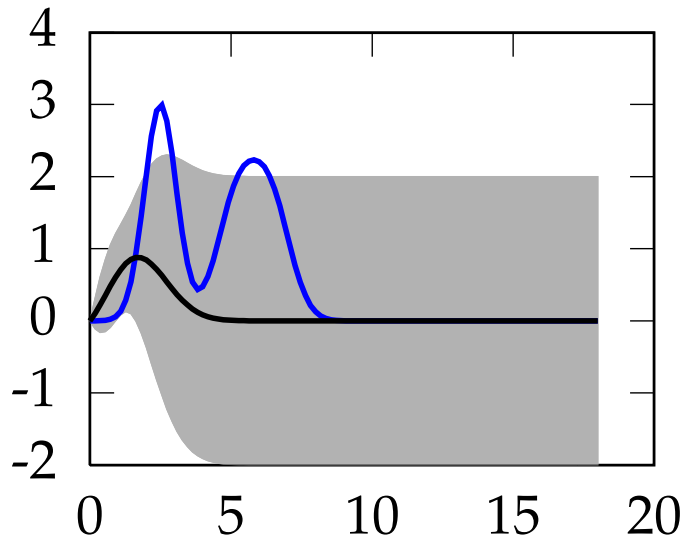

8
7
6
5
4
3
2
1
0
0
5
10
15
20

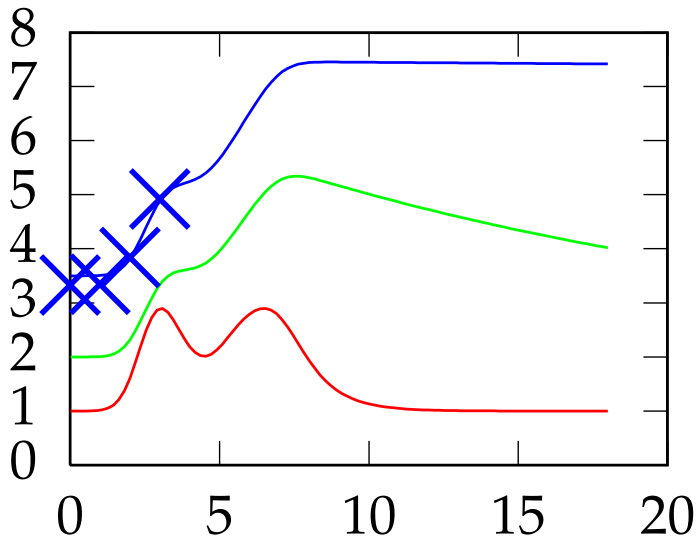

4
3
2
1
0
-1
-2
0
5
10
15
20

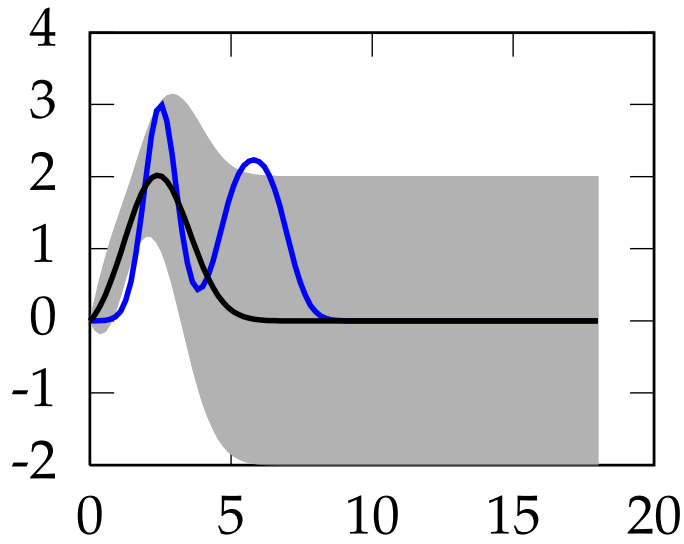

8
7
6
5
4
3
2
1
0
0
5
10
15
20

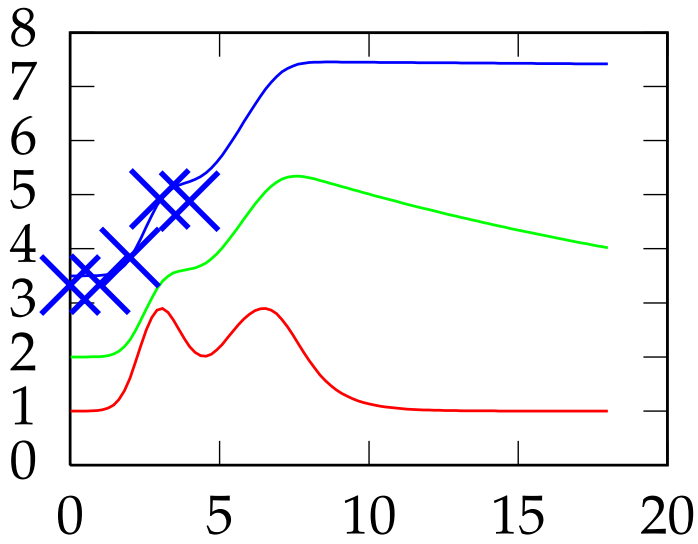

4
3
2
1
0
-1
-2
0
5
10
15
20

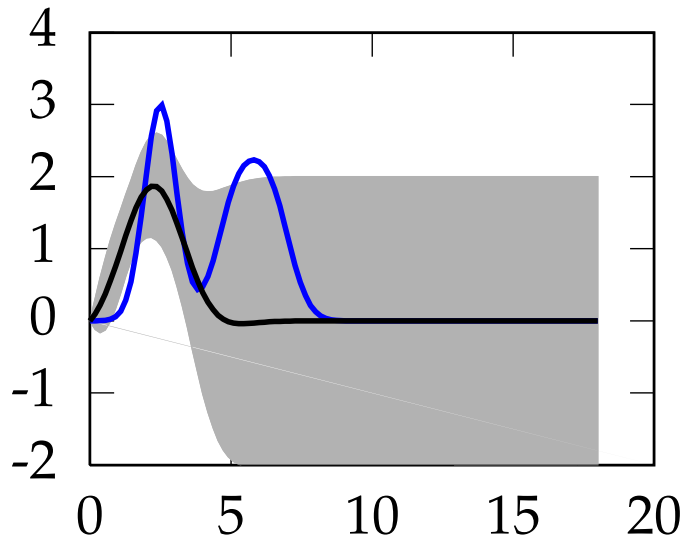

8
7
6
5
4
3
2
1
0
0
5
10
15
20

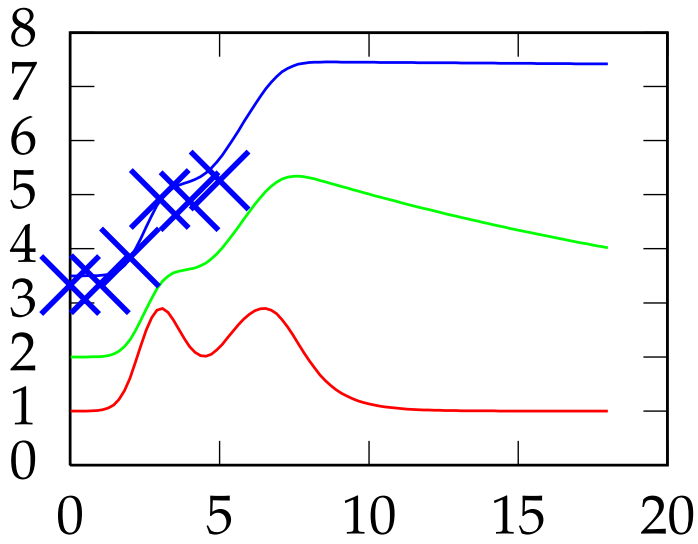

4
3
2
1
0
-1
-2
0
5
10
15
20

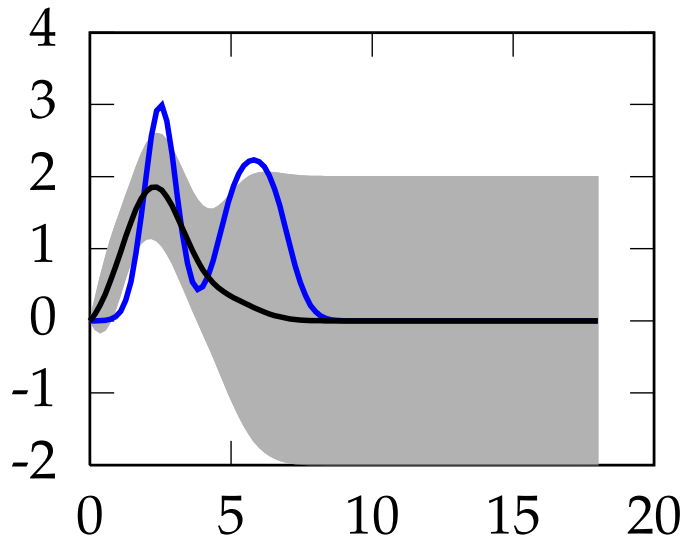

8
7
6
5
4
3
2
1
0
0
5
10
15
20

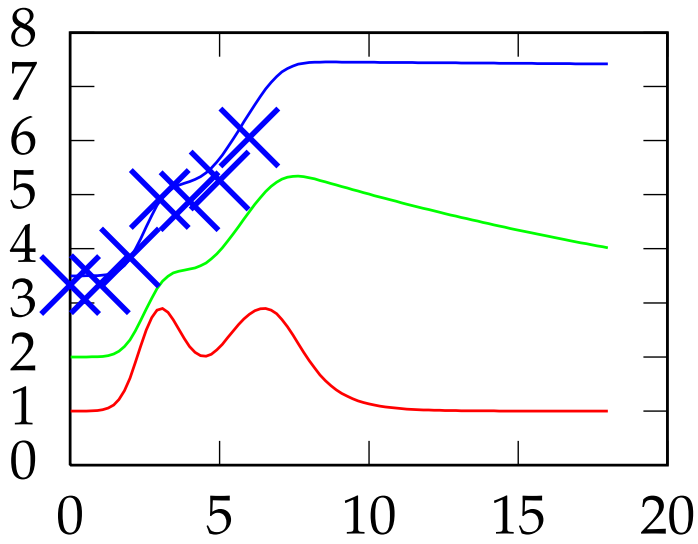

4
3
2
1
0
-1
-2
0
5
10
15
20

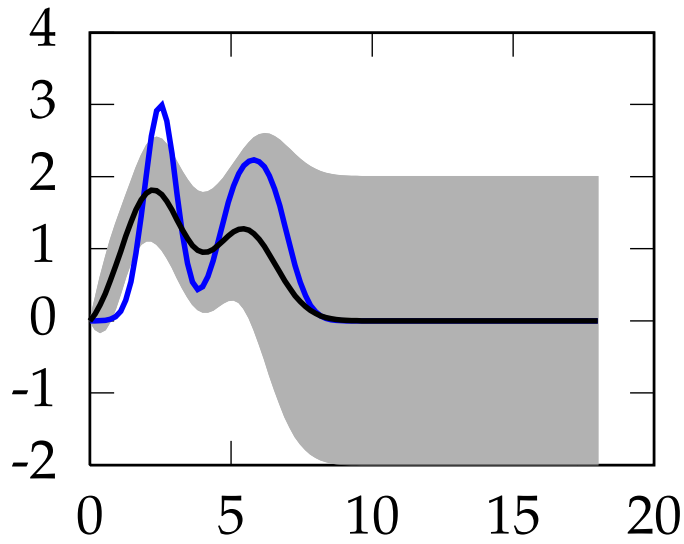

8
7
6
5
4
3
2
1
0
0
5
10
15
20

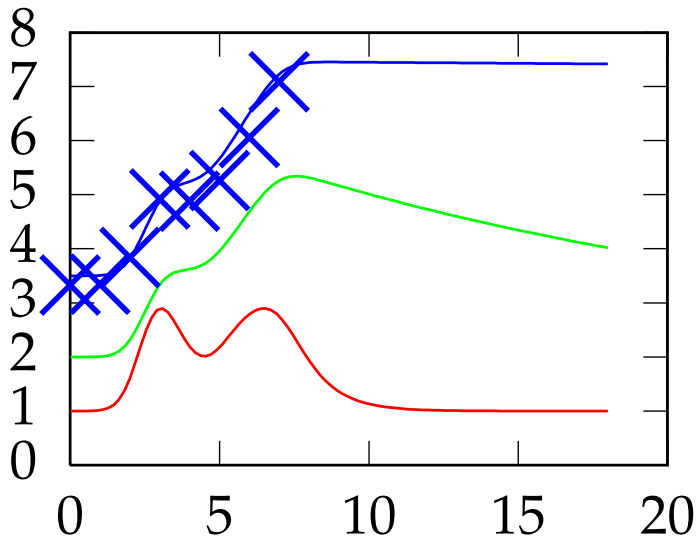

4
3
2
1
0
-1
-2
0
5
10
15
20

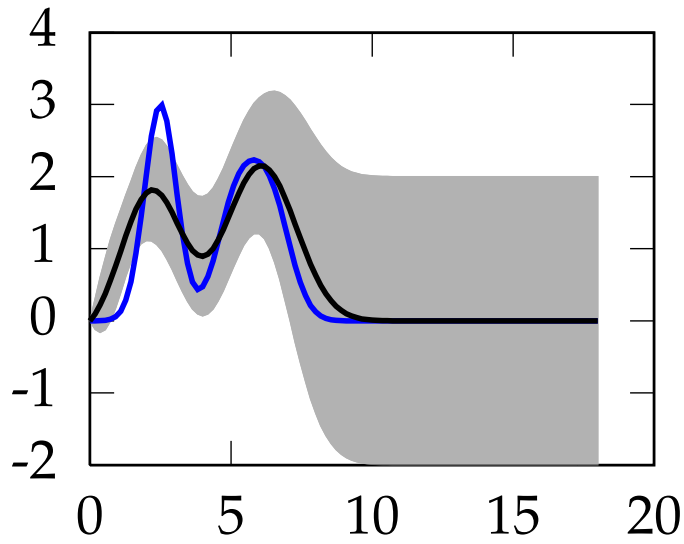

8
7
6
5
4
3
2
1
0
0
5
10
15
20

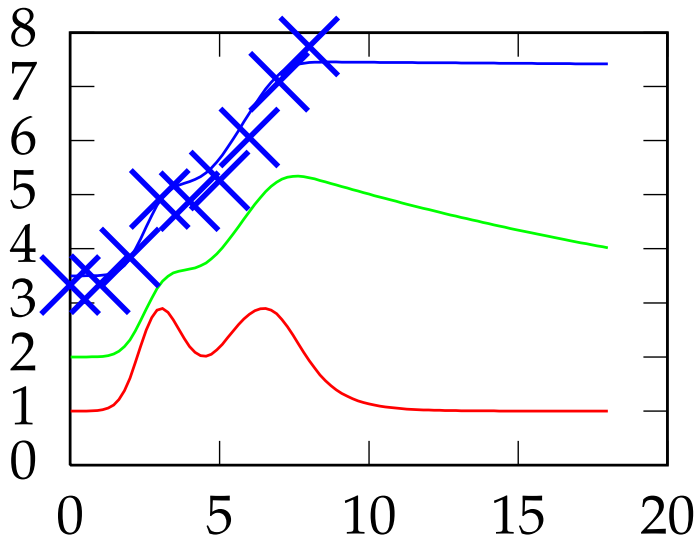

4
3
2
1
0
-1
-2
0
5
10
15
20

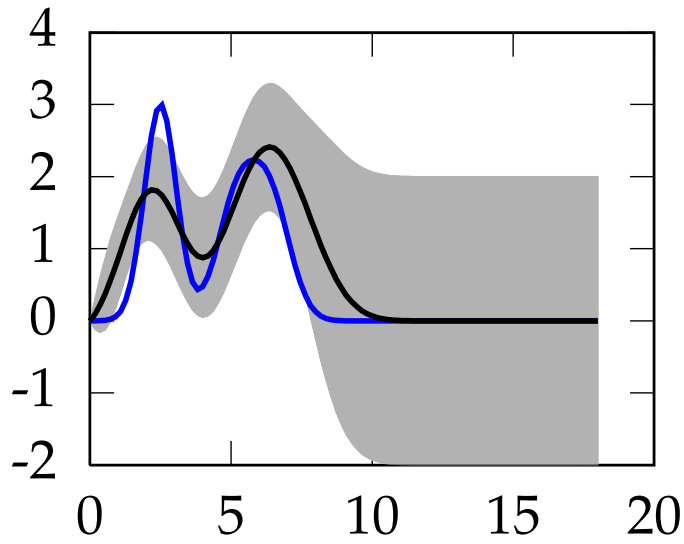

8
7
6
5
4
3
2
1
0
0
5
10
15
20

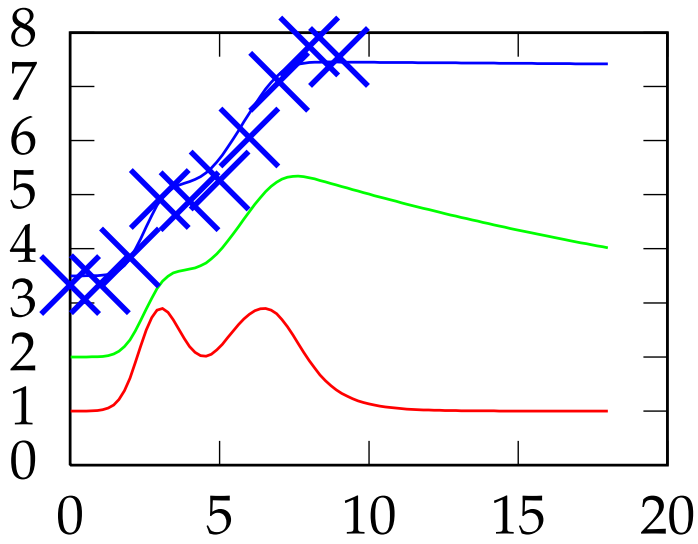

4
3
2
1
0
-1
-2
0
5
10
15
20

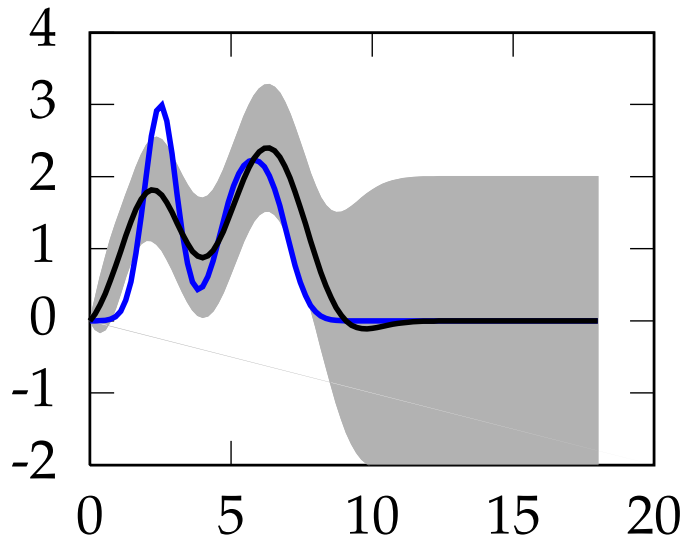

8
7
6
5
4
3
2
1
0
0
5
10
15
20

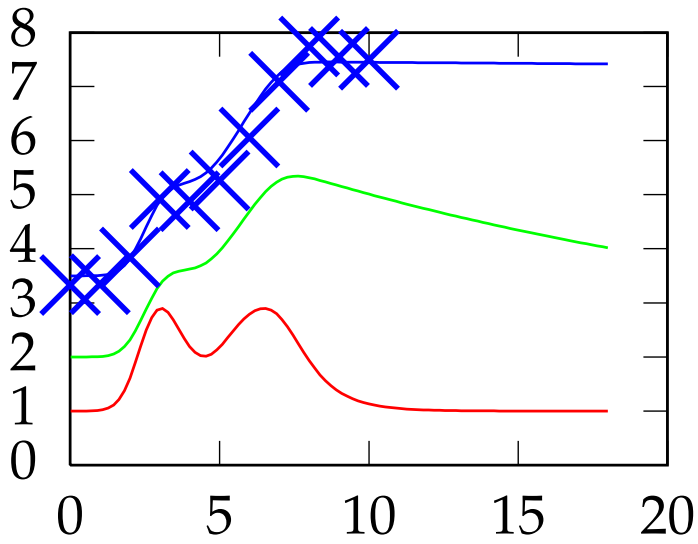

4
3
2
1
0
-1
-2
0
5
10
15
20

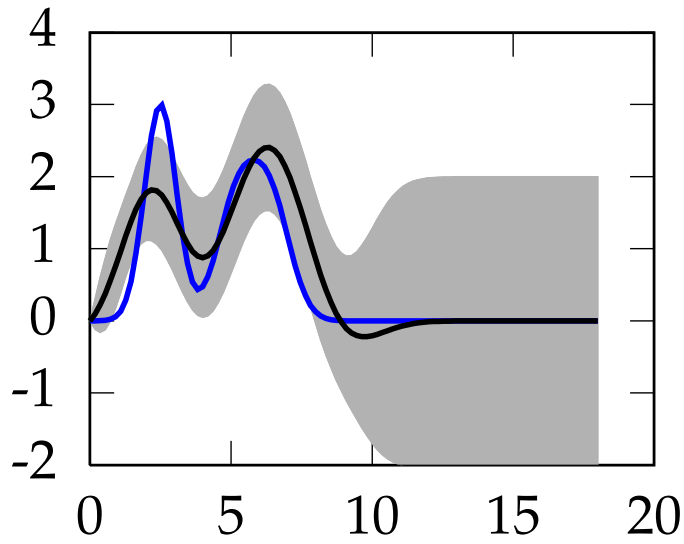

8
7
6
5
4
3
2
1
0
0
5
10
15
20

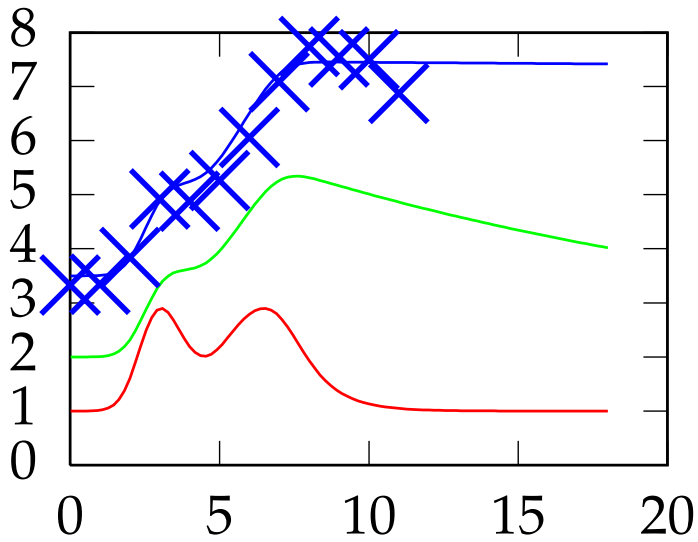

4
3
2
1
0
-1
-2
0
5
10
15
20

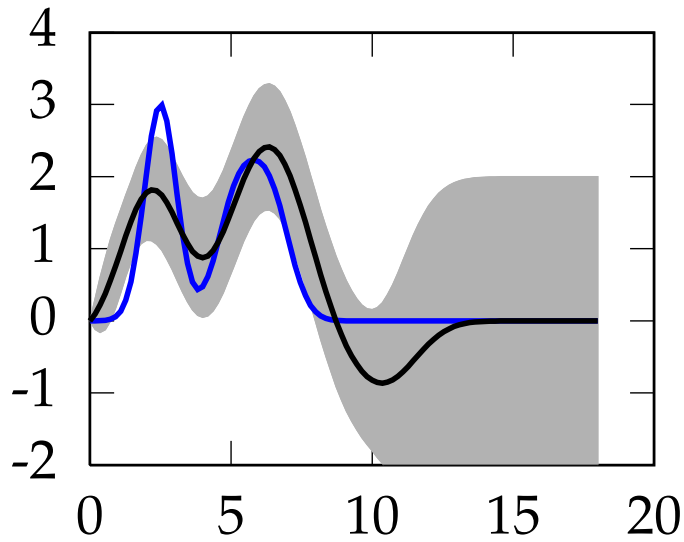

8
7
6
5
4
3
2
1
0
0
5
10
15
20

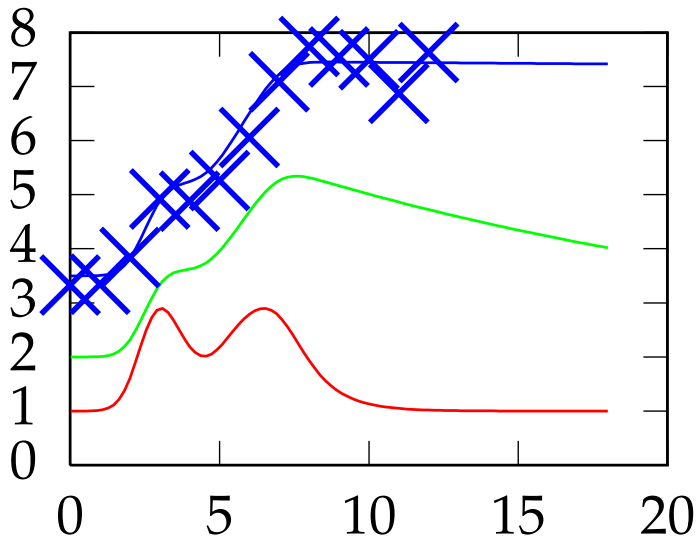

4
3
2
1
0
-1
-2
0
5
10
15
20

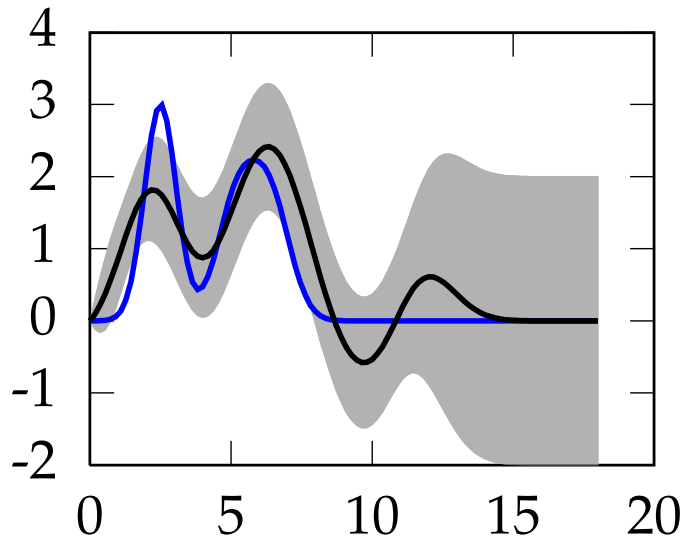

8
7
6
5
4
3
2
1
0
0
5
10
15
20

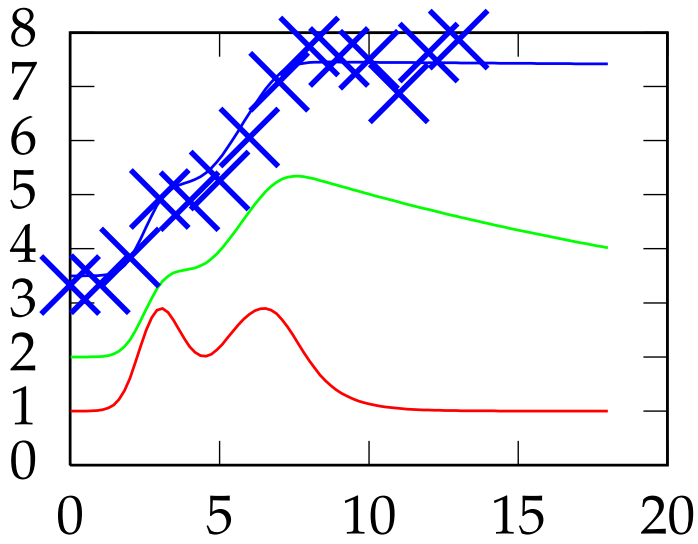

4
3
2
1
0
-1
-2
0
5
10
15
20

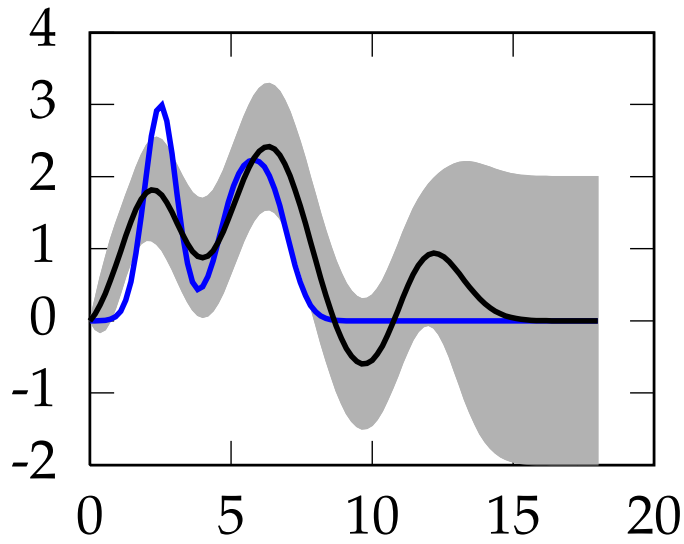

8
7
6
5
4
3
2
1
0
0
5
10
15
20

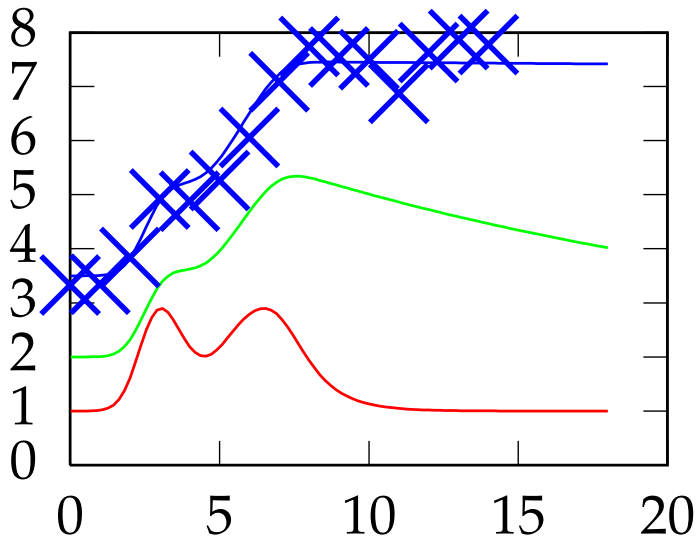

4
3
2
1
0
-1
-2
0
5
10
15
20

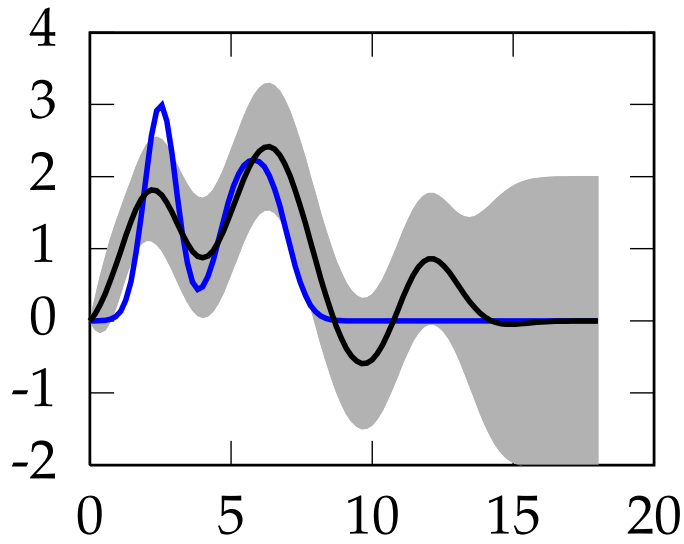

8
7
6
5
4
3
2
1
0
0
5
10
15
20

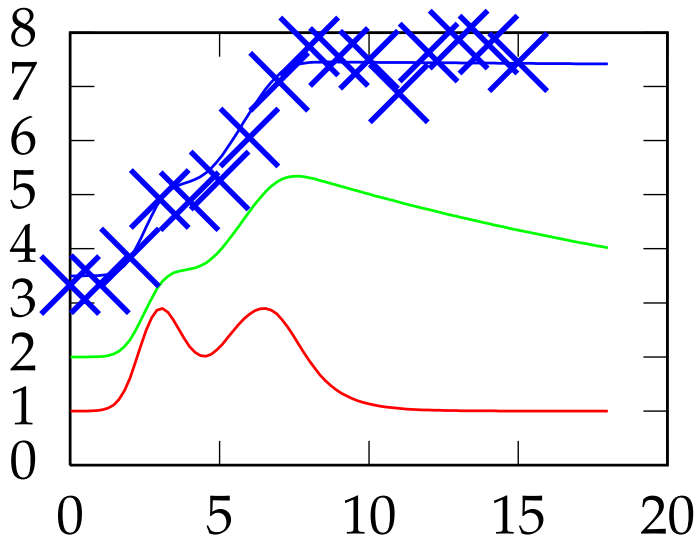

4
3
2
1
0
-1
-2
0
5
10
15
20

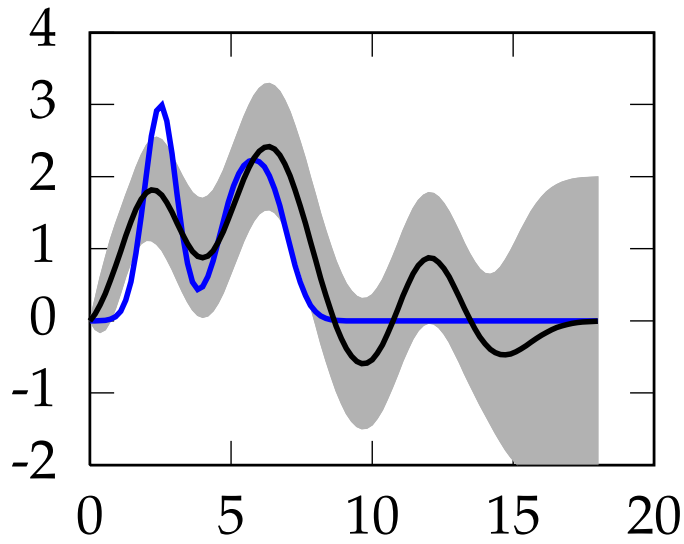

8
7
6
5
4
3
2
1
0
0
5
10
15
20

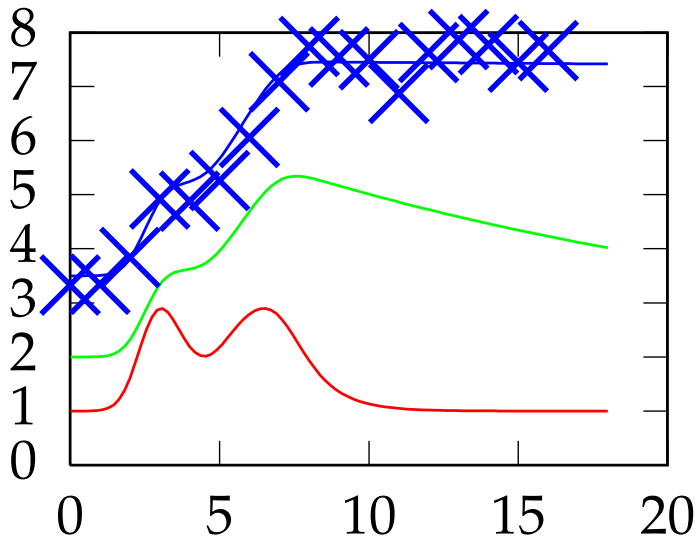

4
3
2
1
0
-1
-2
0
5
10
15
20

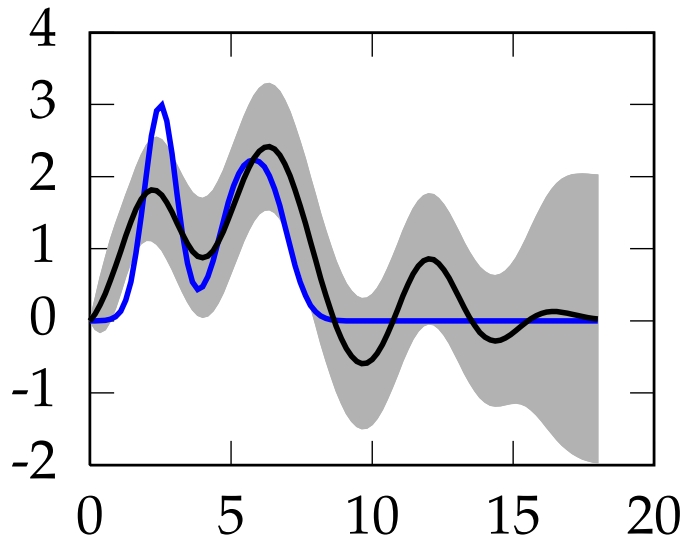

8
7
6
5
4
3
2
1
0
0
5
10
15
20

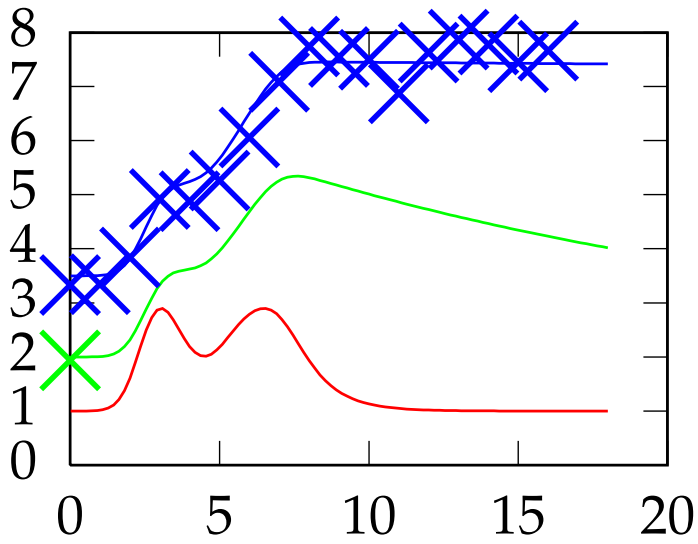

4
3
2
1
0
-1
-2
0
5
10
15
20

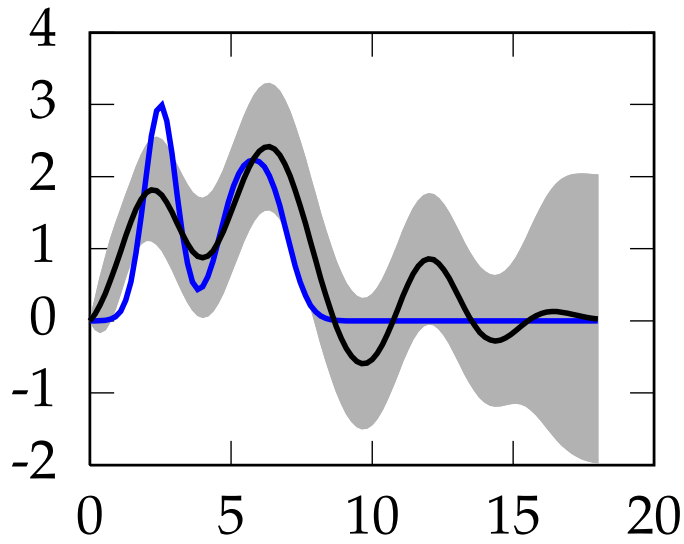

8
7
6
5
4
3
2
1
0
0
5
10
15
20

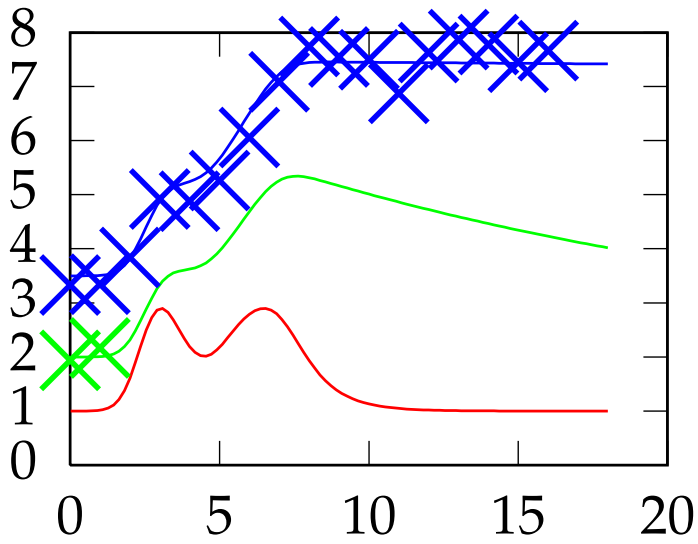

4
3
2
1
0
-1
-2
0
5
10
15
20

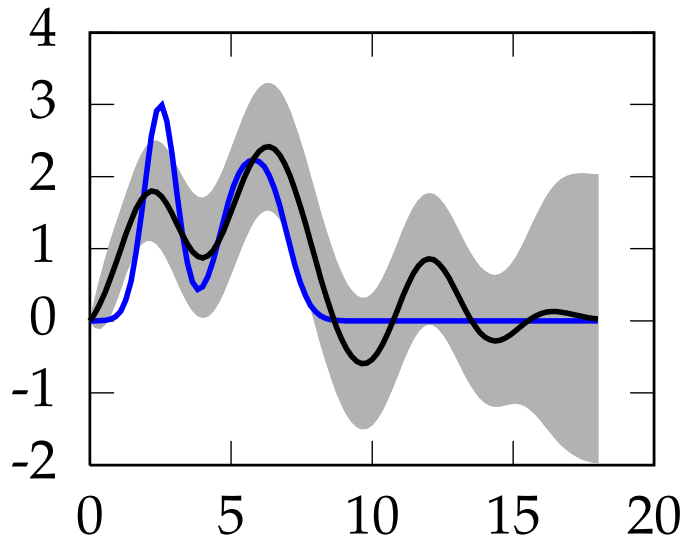

8
7
6
5
4
3
2
1
0
0
5
10
15
20

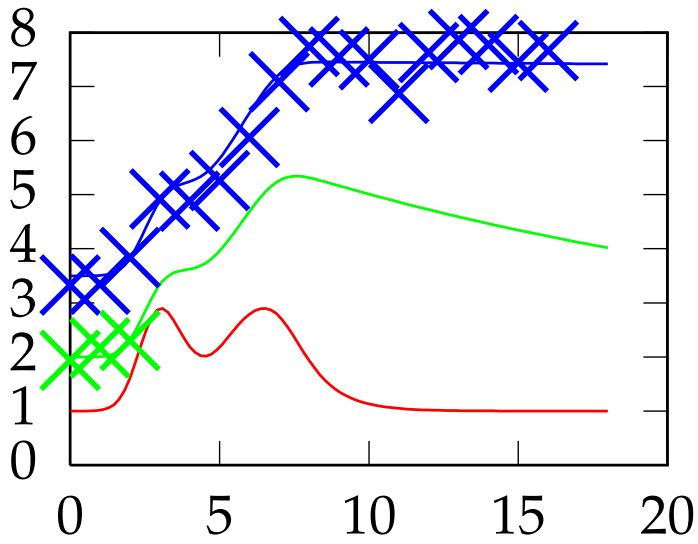

4
3
2
1
0
-1
-2
0
5
10
15
20

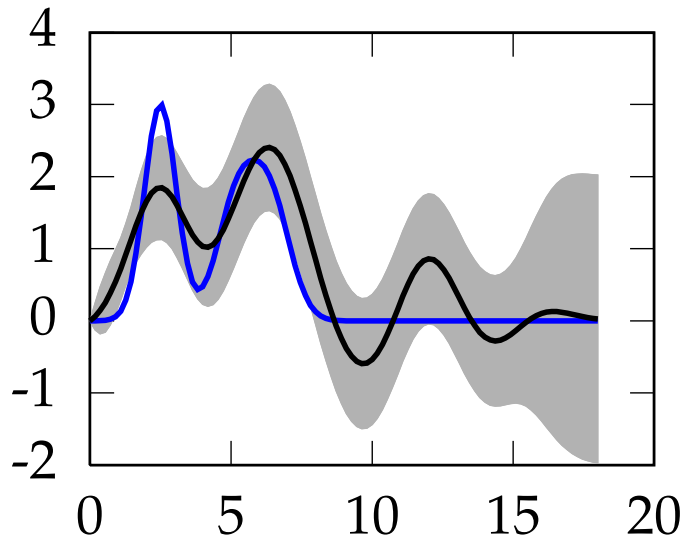

8
7
6
5
4
3
2
1
0
0
5
10
15
20

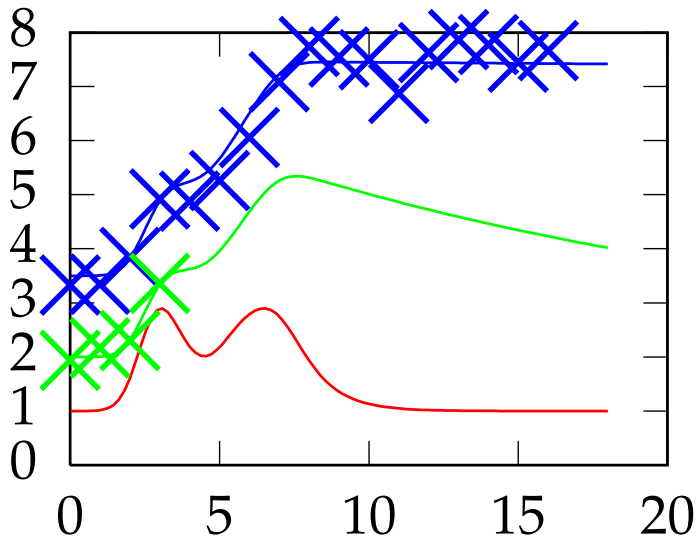

4
3
2
1
0
-1
-2
0
5
10
15
20

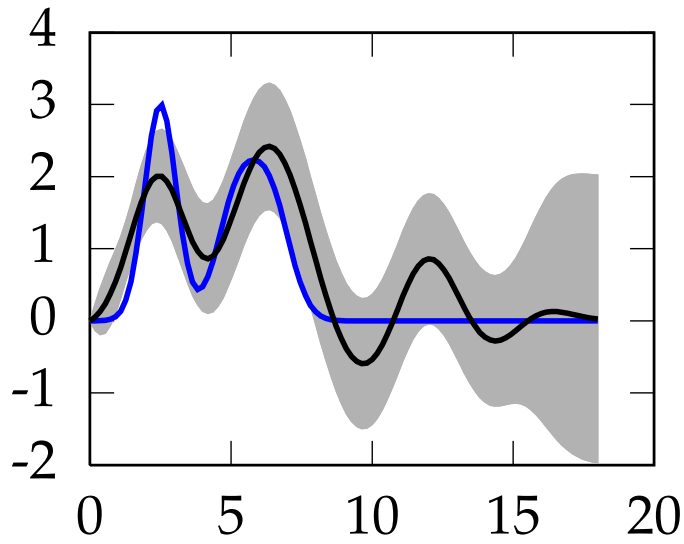

8
7
6
5
4
3
2
1
0
0
5
10
15
20

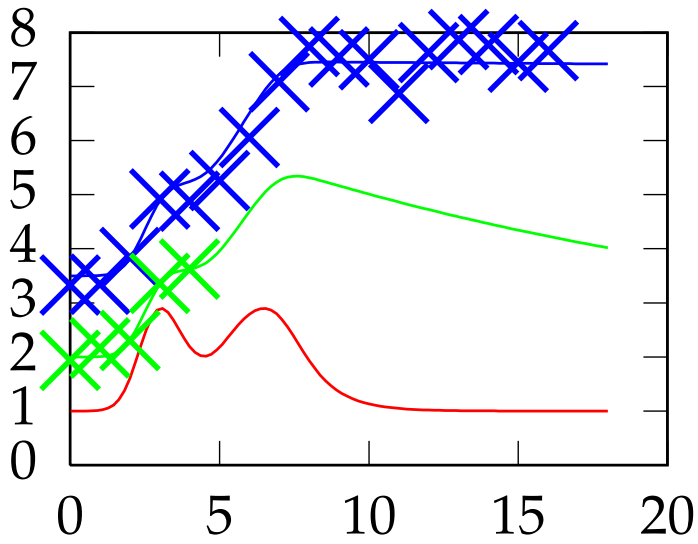

4
3
2
1
0
-1
-2
0
5
10
15
20

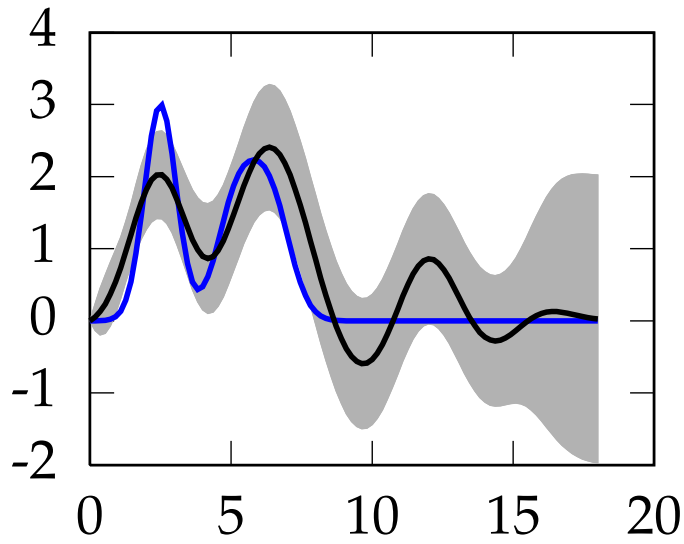

8
7
6
5
4
3
2
1
0
0
5
10
15
20

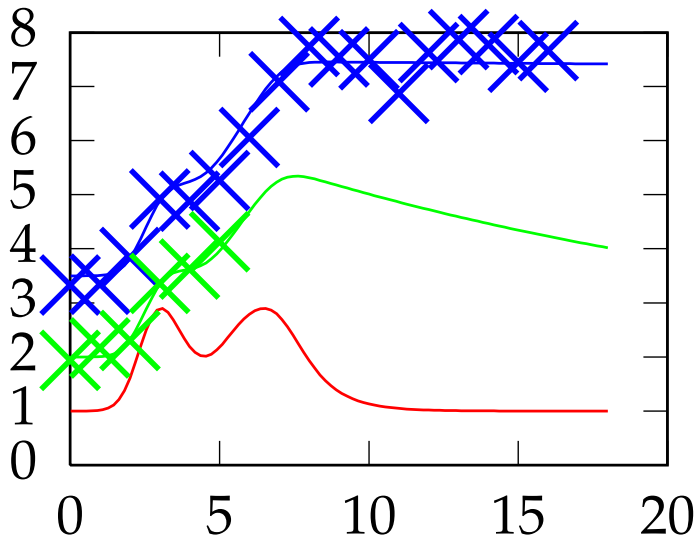

4
3
2
1
0
-1
-2
0
5
10
15
20

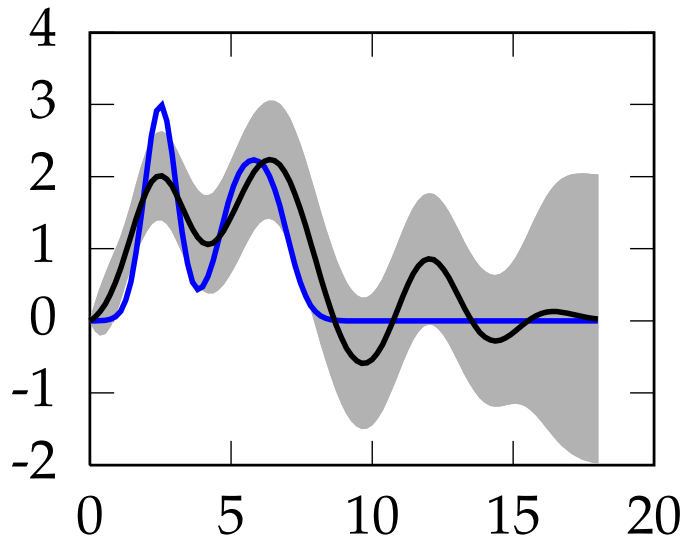

8
7
6
5
4
3
2
1
0
0
5
10
15
20

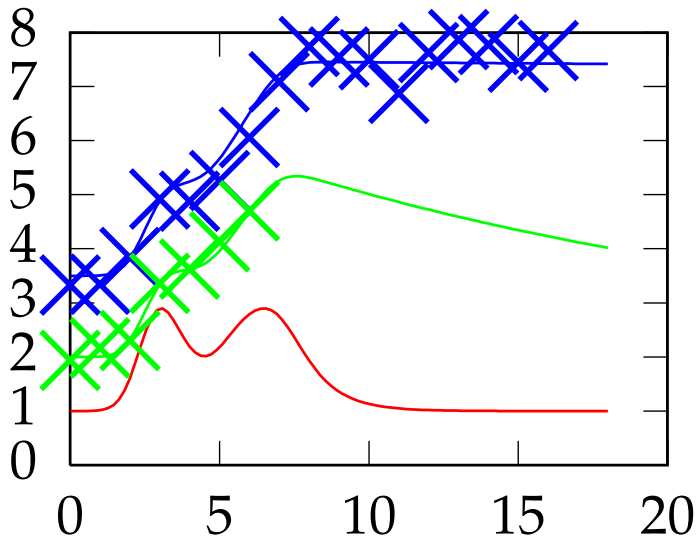

4
3
2
1
0
-1
-2
0
5
10
15
20

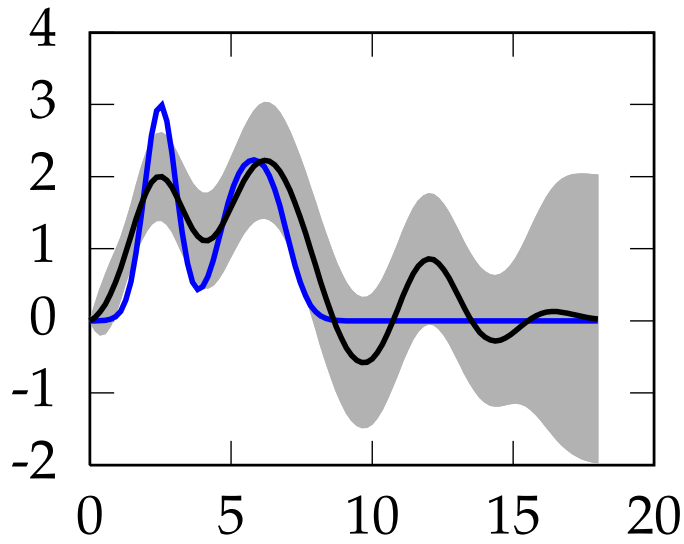

8
7
6
5
4
3
2
1
0
0
5
10
15
20

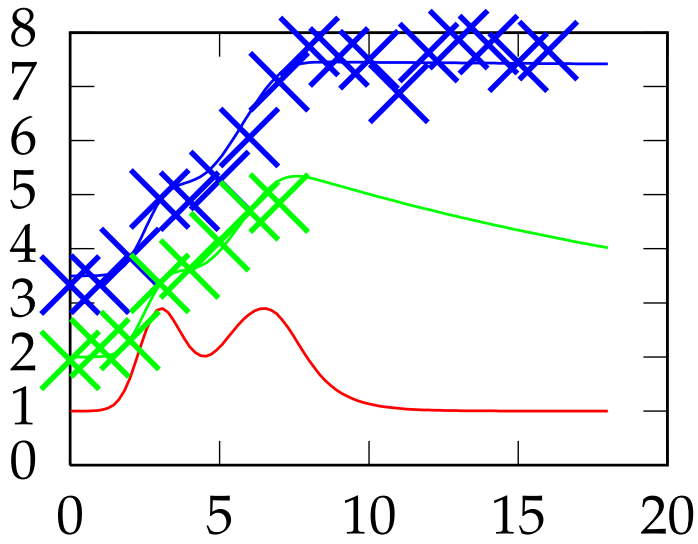

4
3
2
1
0
-1
-2
0
5
10
15
20

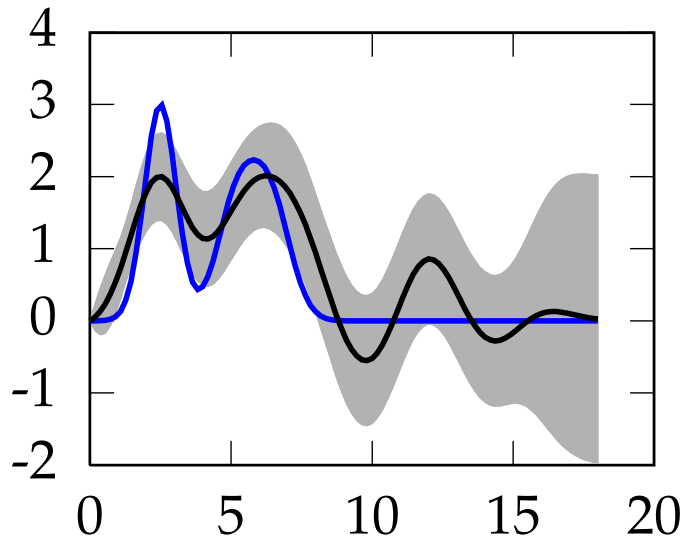

8
7
6
5
4
3
2
1
0
0
5
10
15
20

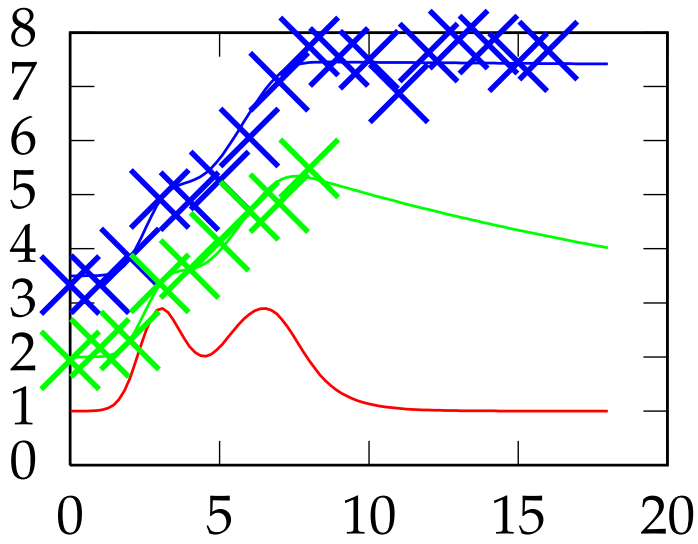

4
3
2
1
0
-1
-2
0
5
10
15
20

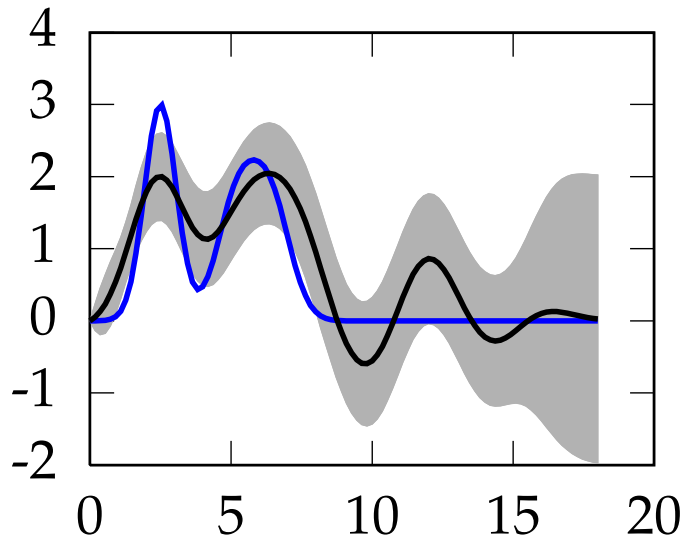

8
7
6
5
4
3
2
1
0
0
5
10
15
20

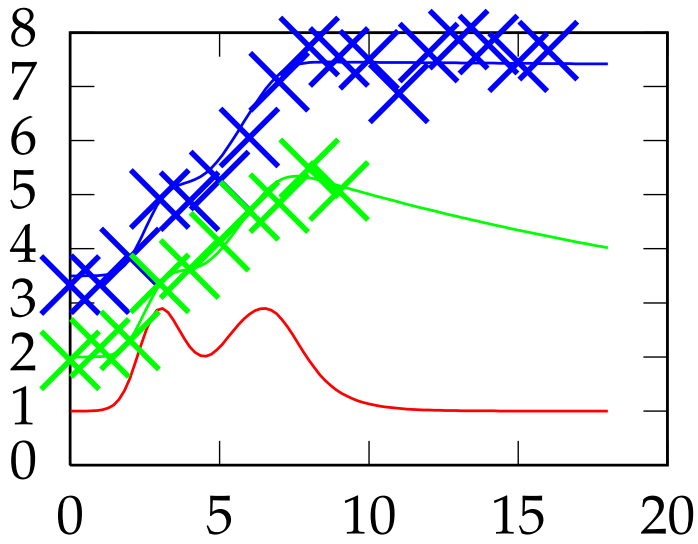

4
3
2
1
0
-1
-2
0
5
10
15
20

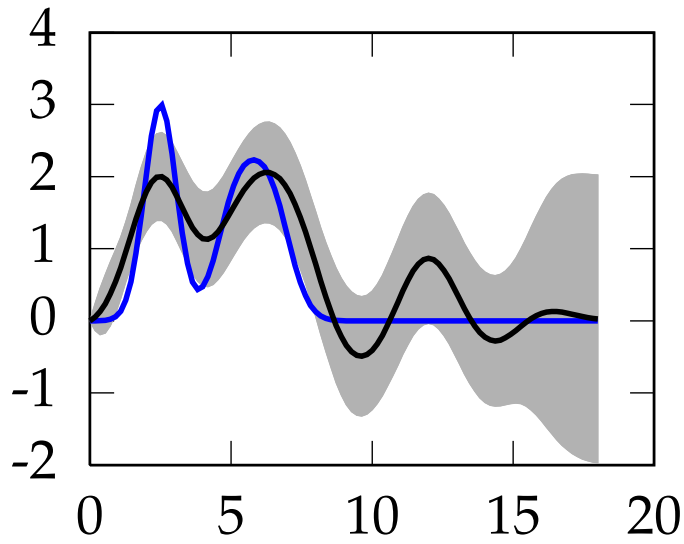

8
7
6
5
4
3
2
1
0
0
5
10
15
20

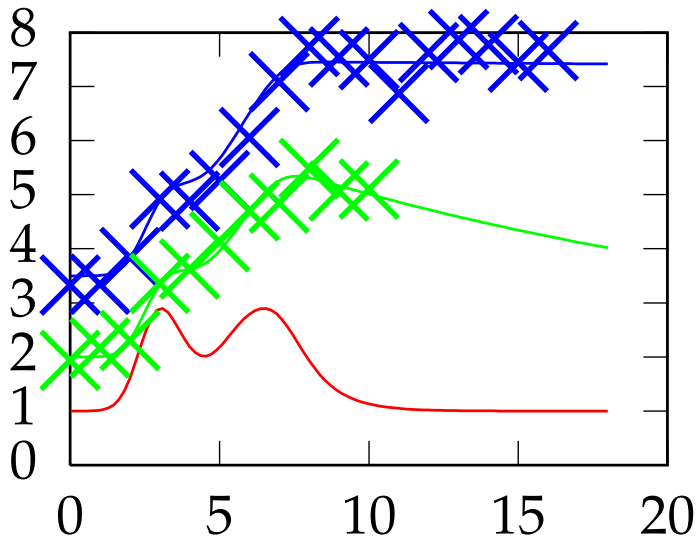

4
3
2
1
0
-1
-2
0
5
10
15
20

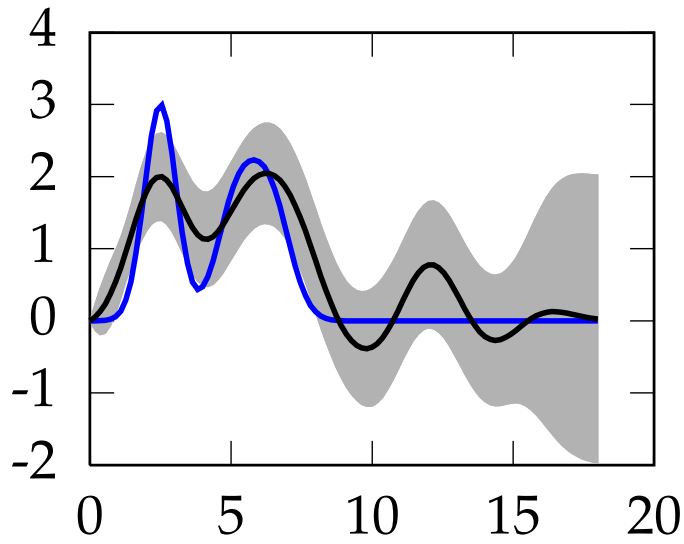

8
7
6
5
4
3
2
1
0
0
5
10
15
20

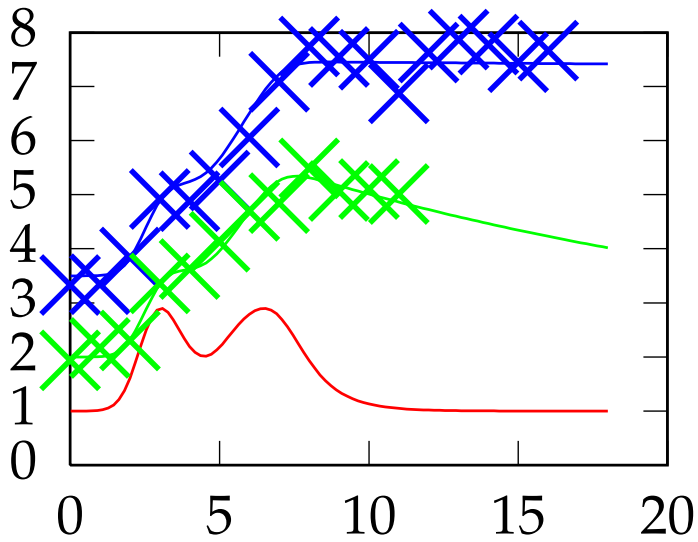

4
3
2
1
0
-1
-2
0
5
10
15
20

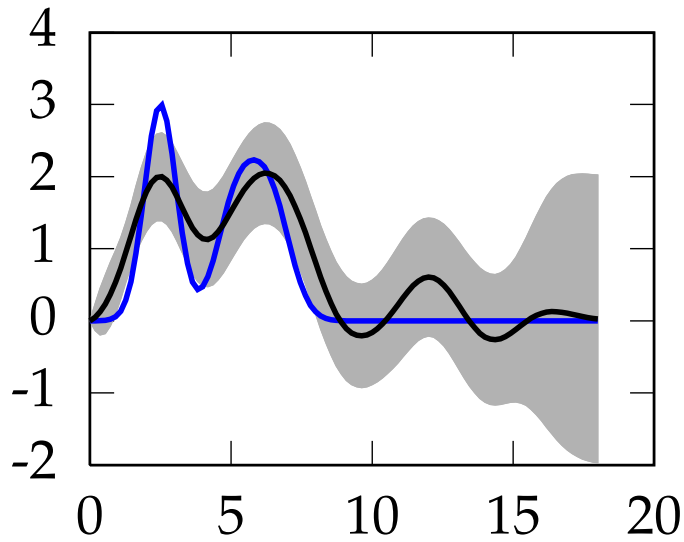

8
7
6
5
4
3
2
1
0
0
5
10
15
20

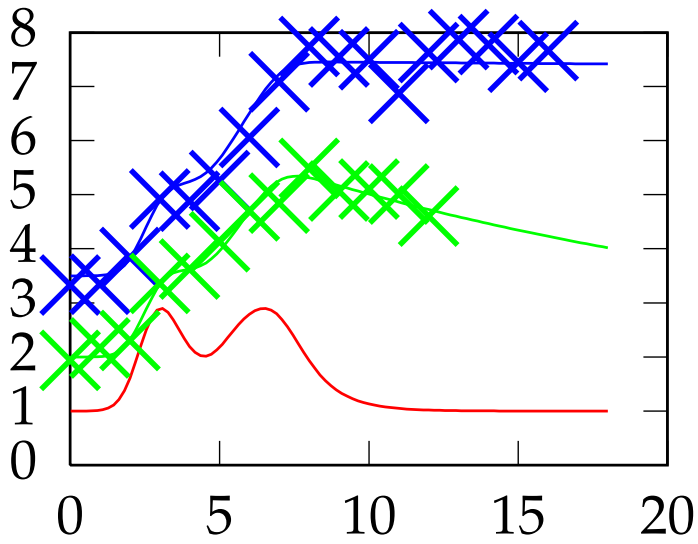

4
3
2
1
0
-1
-2
0
5
10
15
20

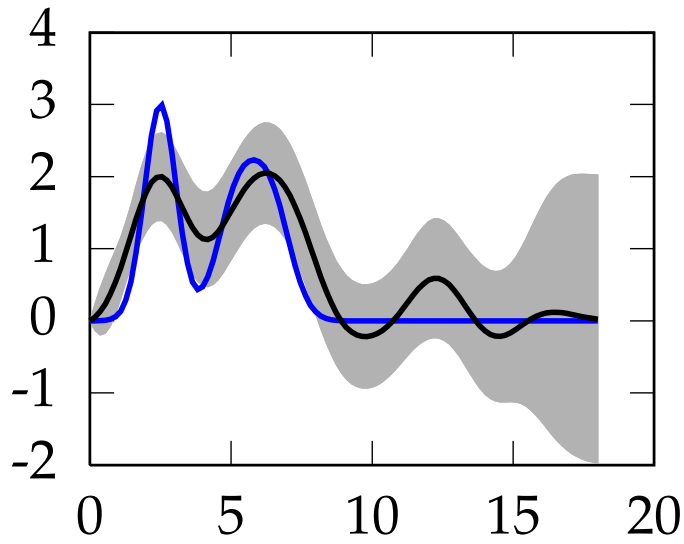

8
7
6
5
4
3
2
1
0
0
5
10
15
20

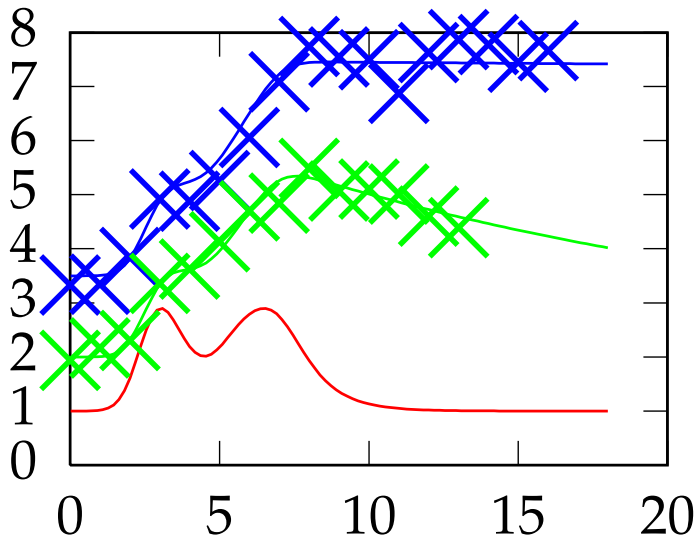

4
3
2
1
0
-1
-2
0
5
10
15
20

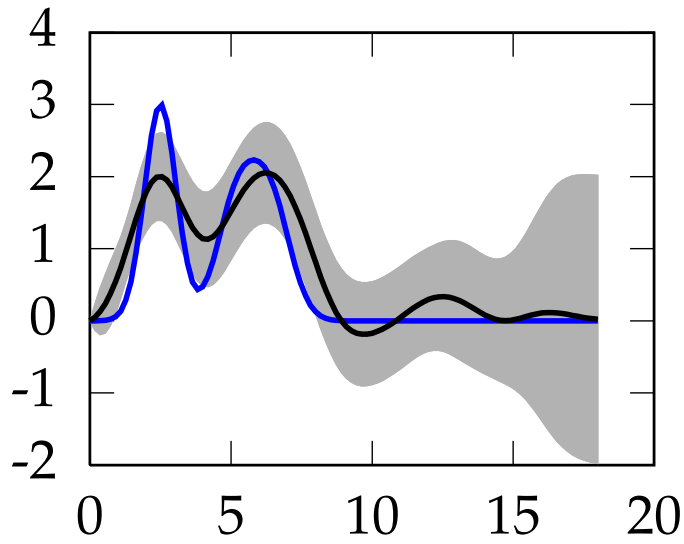

8
7
6
5
4
3
2
1
0
0
5
10
15
20

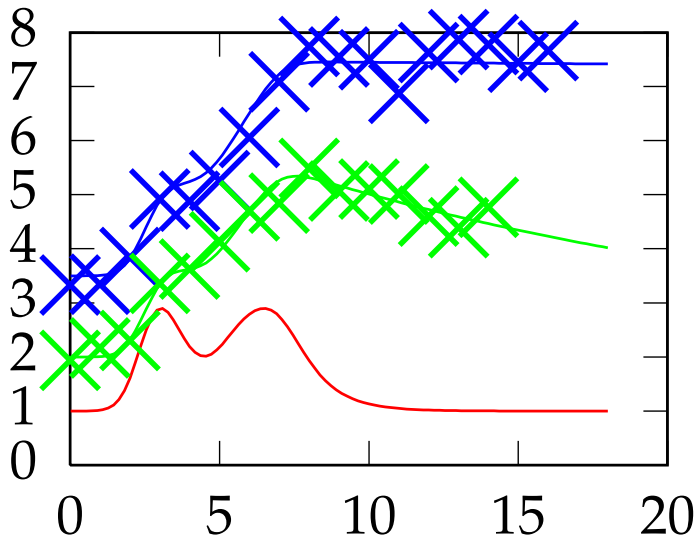

4
3
2
1
0
-1
-2
0
5
10
15
20

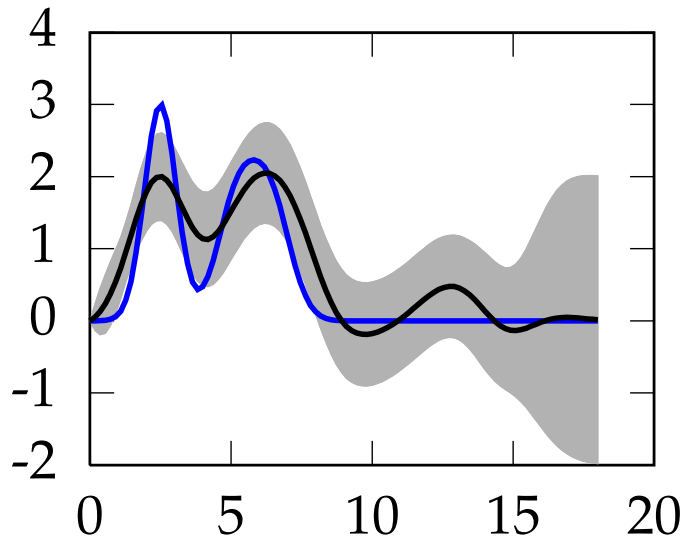

8
7
6
5
4
3
2
1
0
0
5
10
15
20

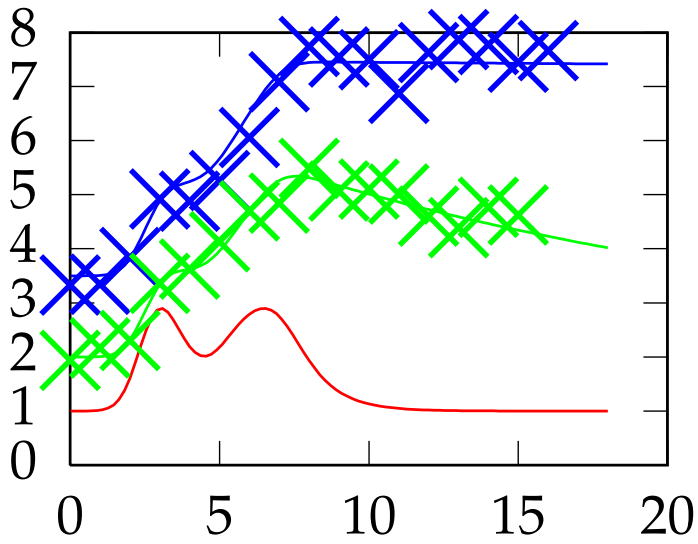

4
3
2
1
0
-1
-2
0
5
10
15
20

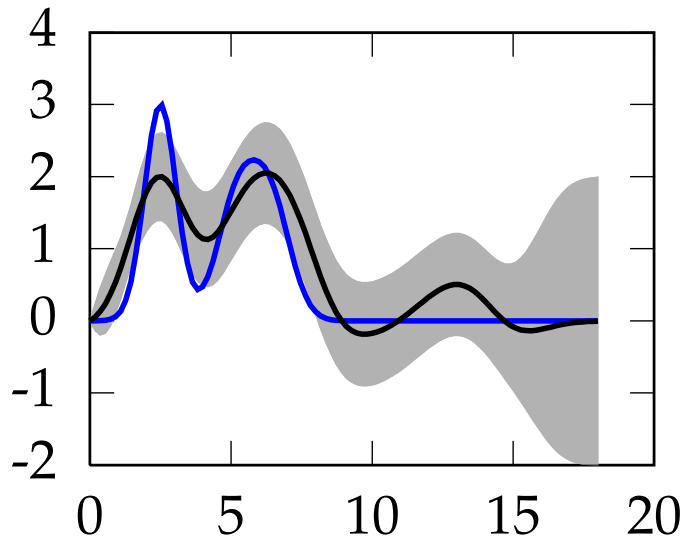

8
7
6
5
4
3
2
1
0
0
5
10
15
20

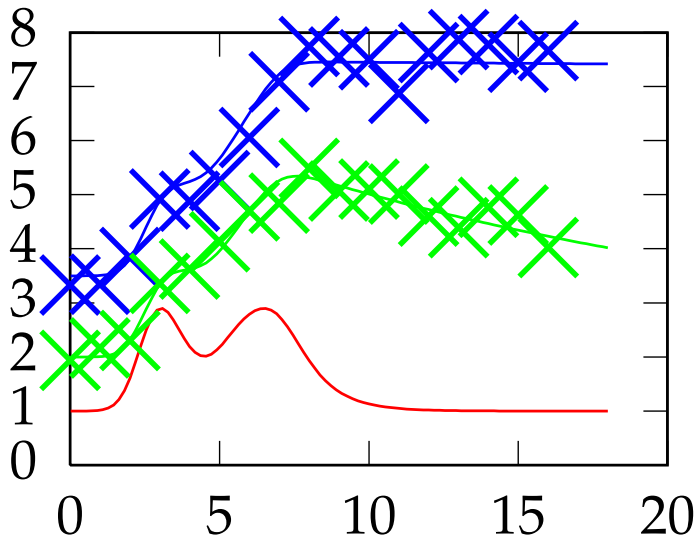

4
3
2
1
0
-1
-2
0
5
10
15
20

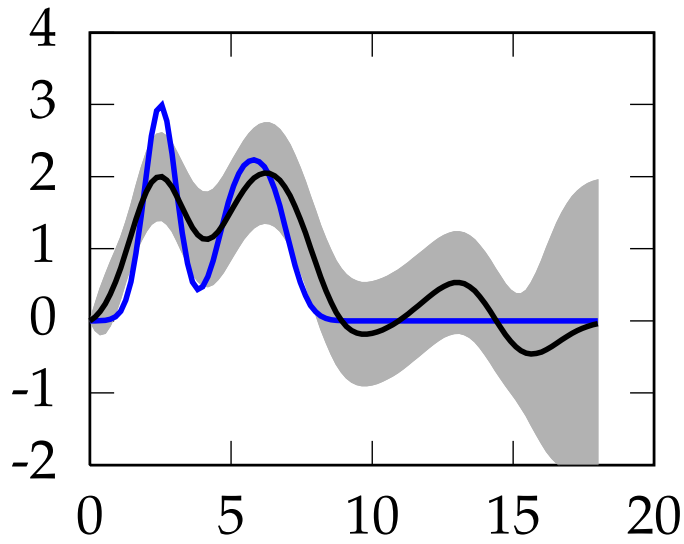

8
7
6
5
4
3
2
1
0
0
5
10
15
20

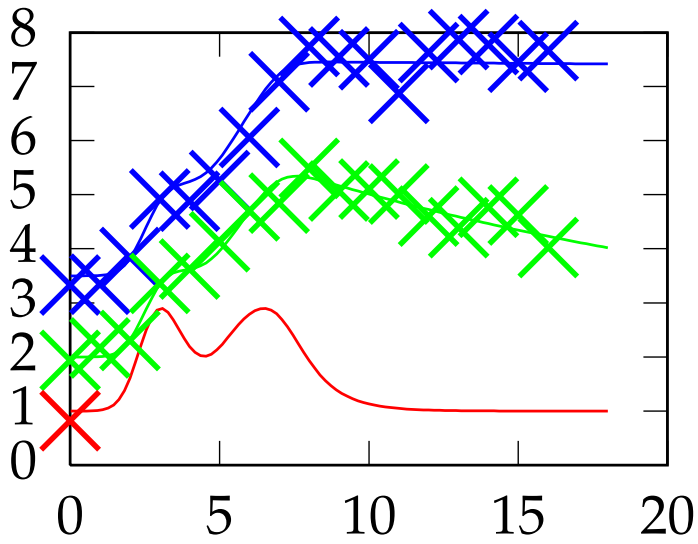

4
3
2
1
0
-1
-2
0
5
10
15
20

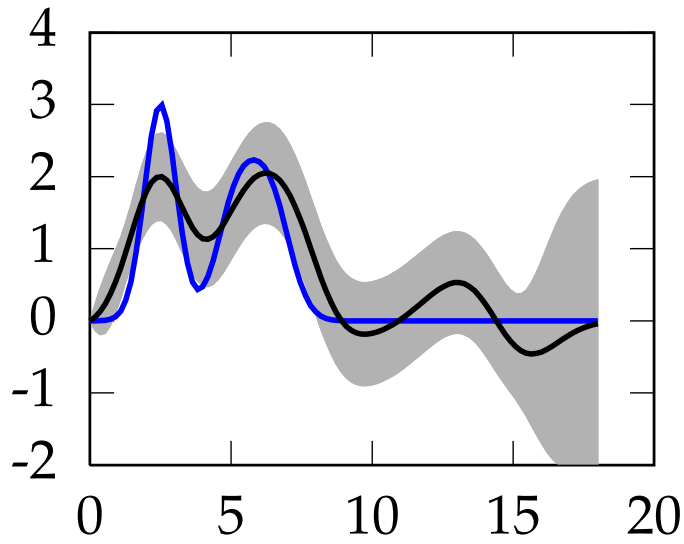

8
7
6
5
4
3
2
1
0
0
5
10
15
20

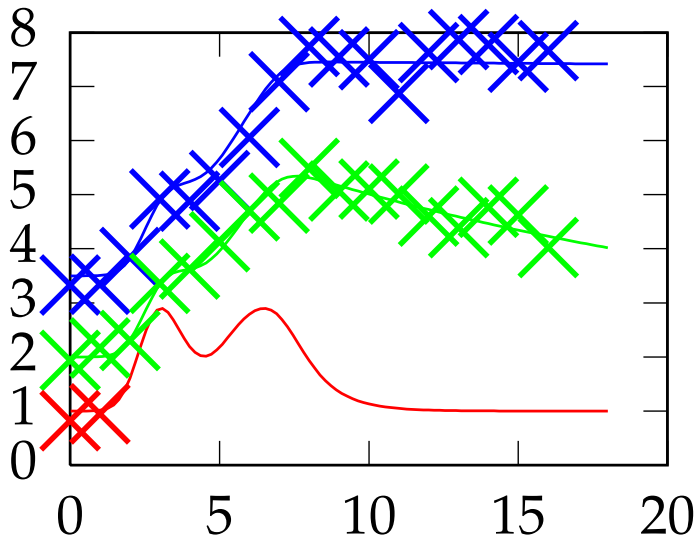

4
3
2
1
0
-1
-2
0
5
10
15
20

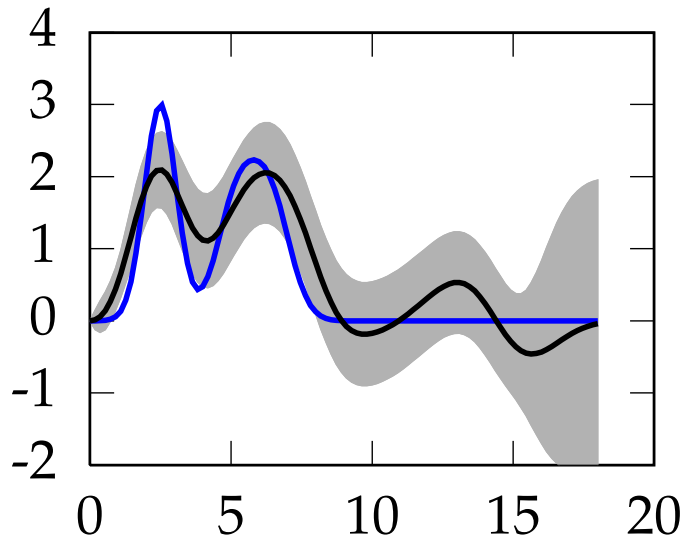

8
7
6
5
4
3
2
1
0
0
5
10
15
20

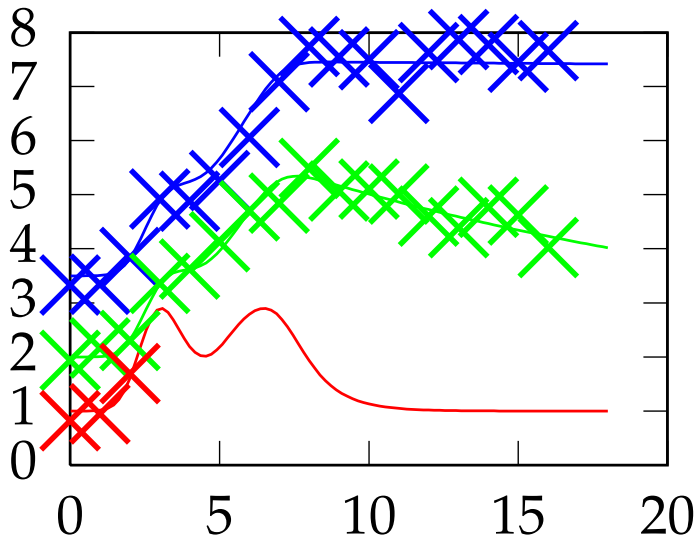

4
3
2
1
0
-1
-2
0
5
10
15
20

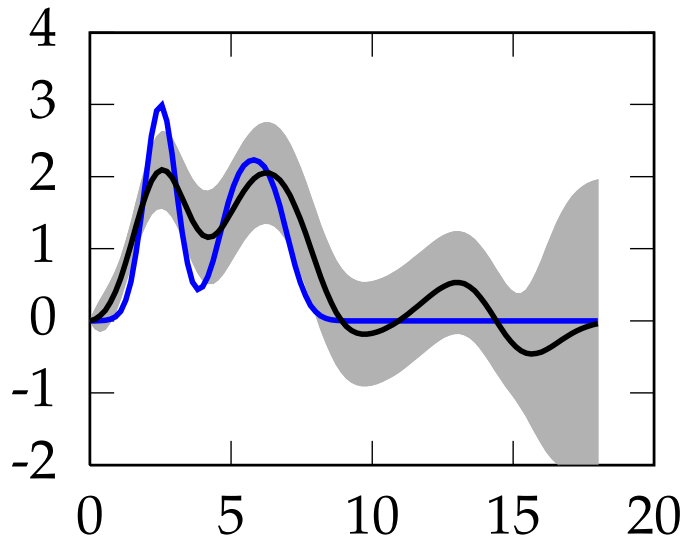

8
7
6
5
4
3
2
1
0
0
5
10
15
20

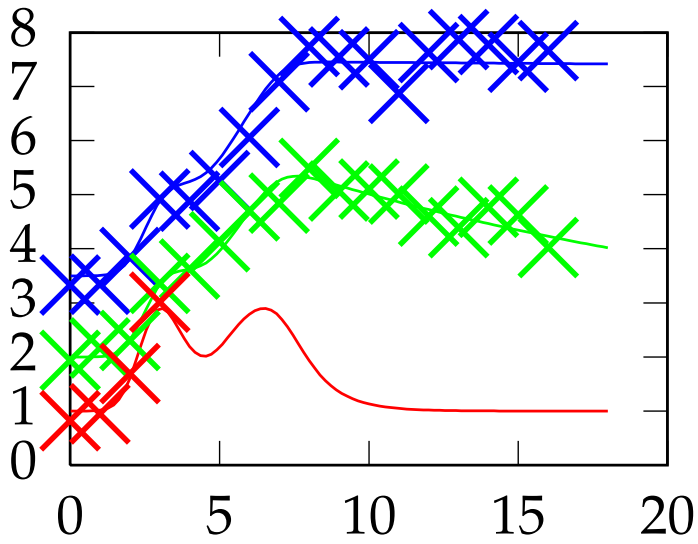

4
3
2
1
0
-1
-2
0
5
10
15
20

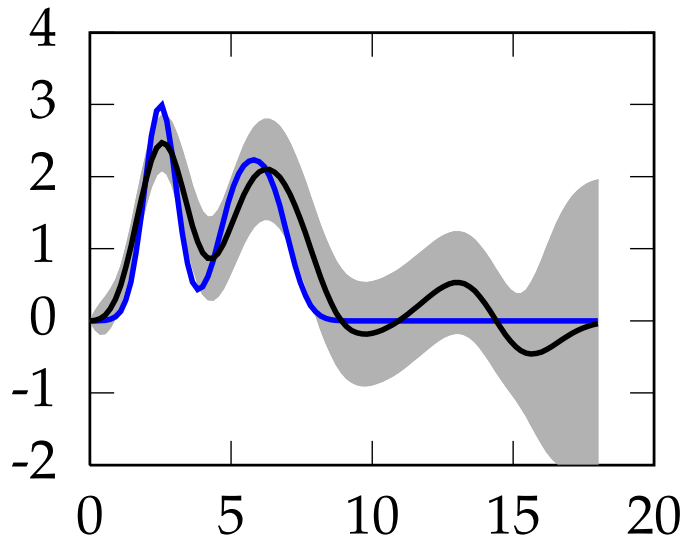

8
7
6
5
4
3
2
1
0
0
5
10
15
20

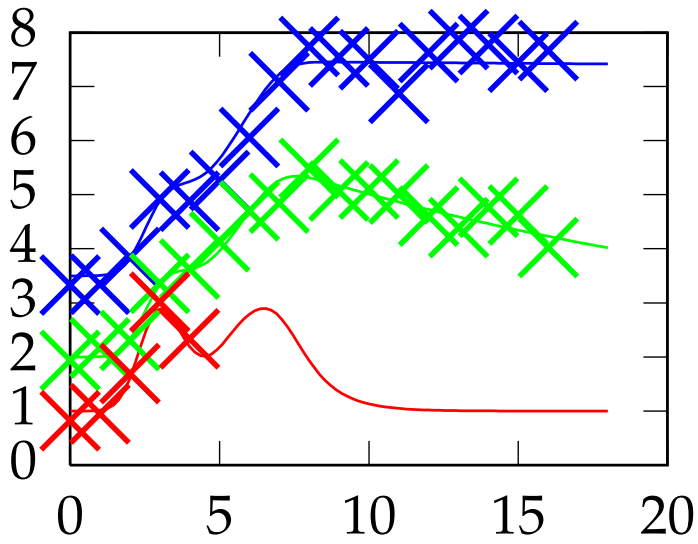

4
3
2
1
0
-1
-2
0
5
10
15
20

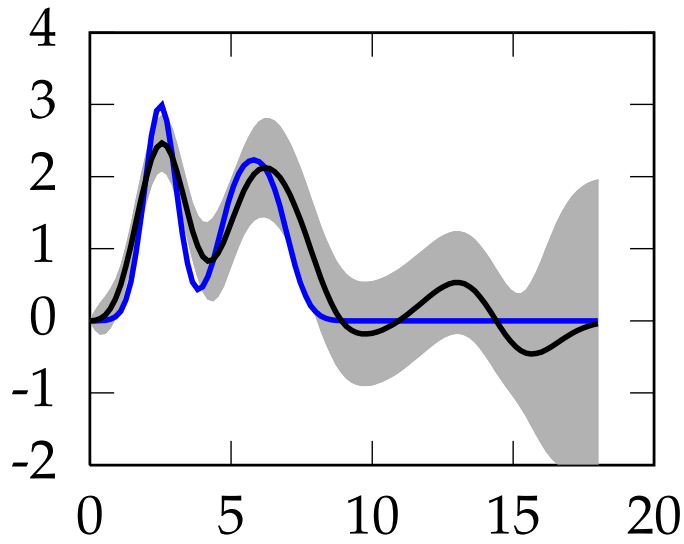

8
7
6
5
4
3
2
1
0
0
5
10
15
20

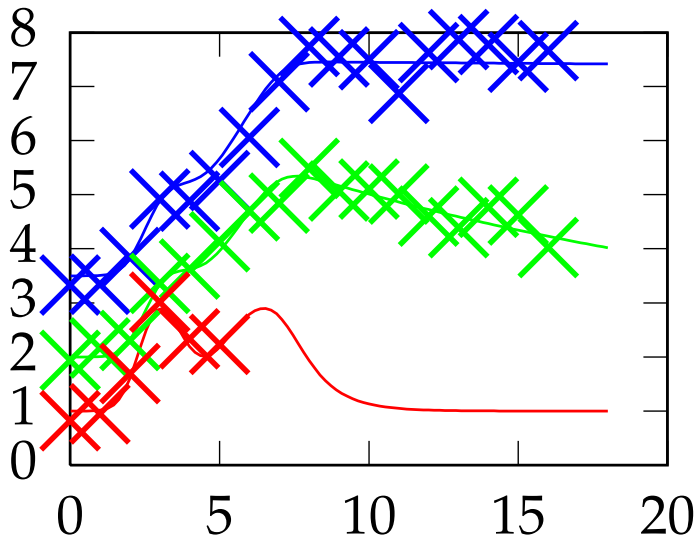

4
3
2
1
0
-1
-2
0
5
10
15
20

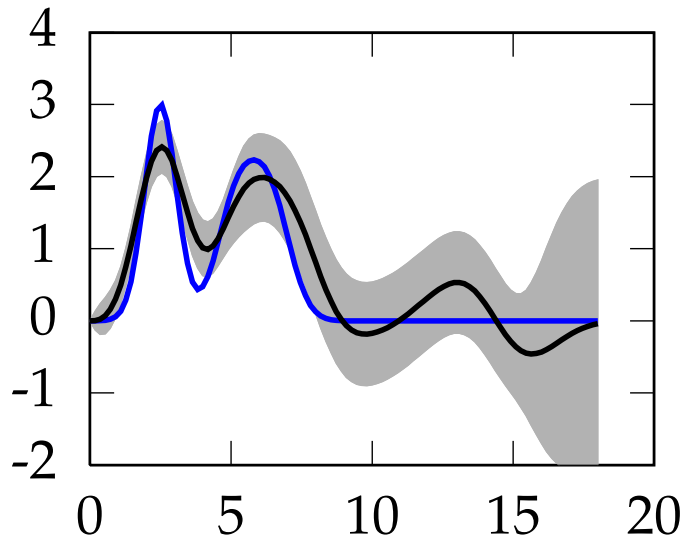

8
7
6
5
4
3
2
1
0
0
5
10
15
20

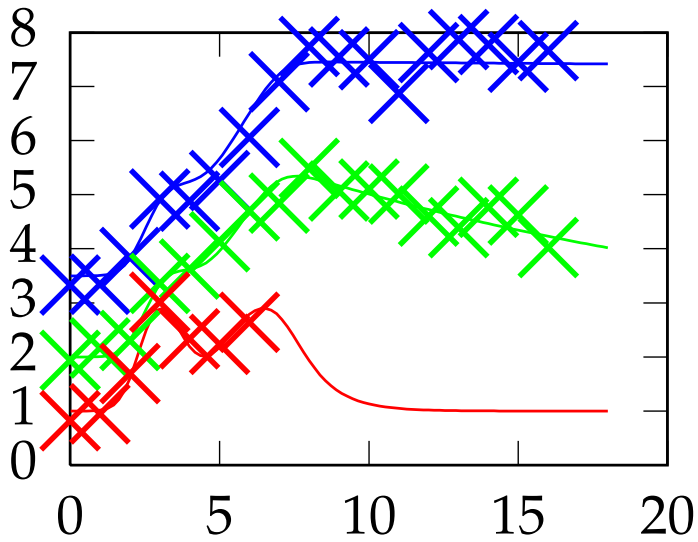

4
3
2
1
0
-1
-2
0
5
10
15
20

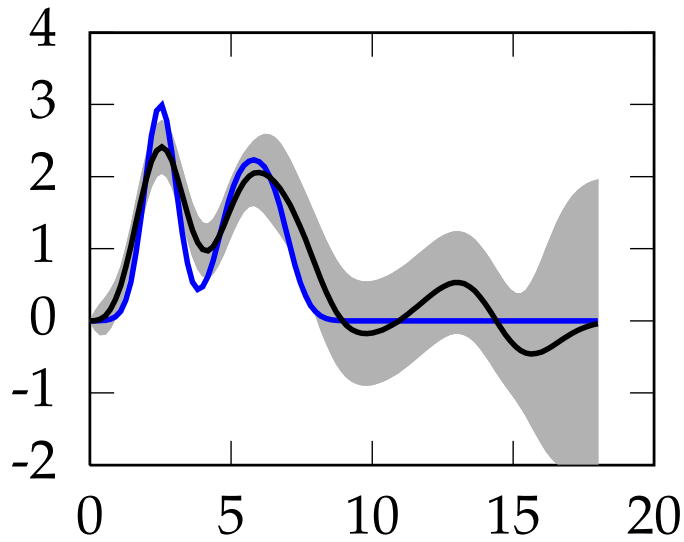

8
7
6
5
4
3
2
1
0
0
5
10
15
20

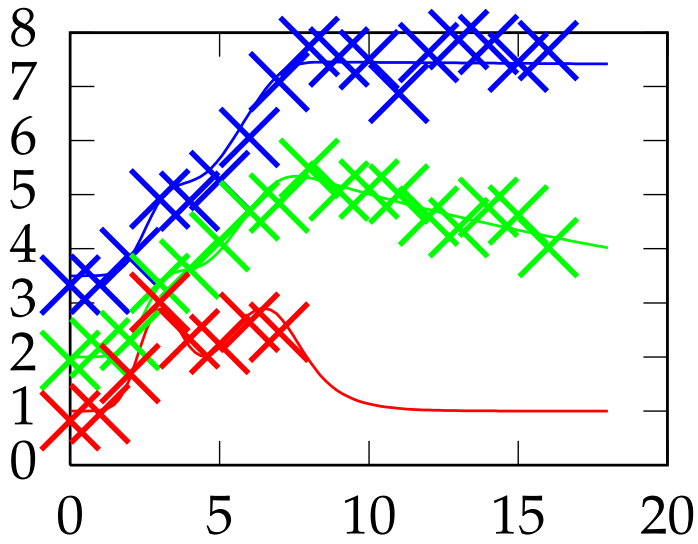

4
3
2
1
0
-1
-2
0
5
10
15
20

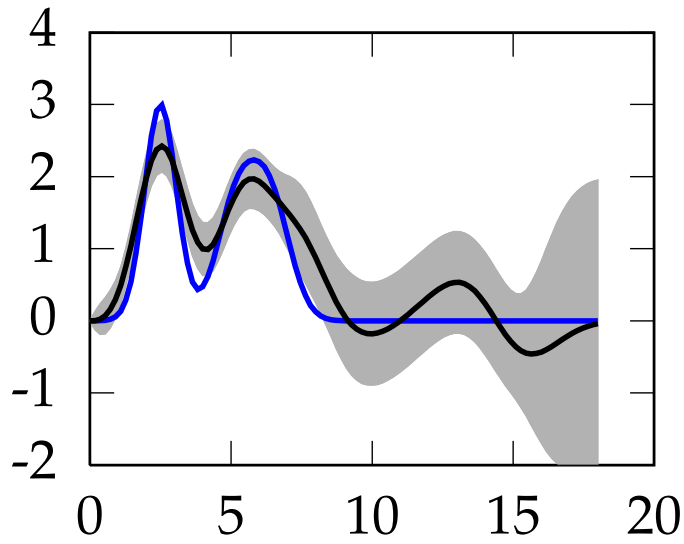

8
7
6
5
4
3
2
1
0
0
5
10
15
20

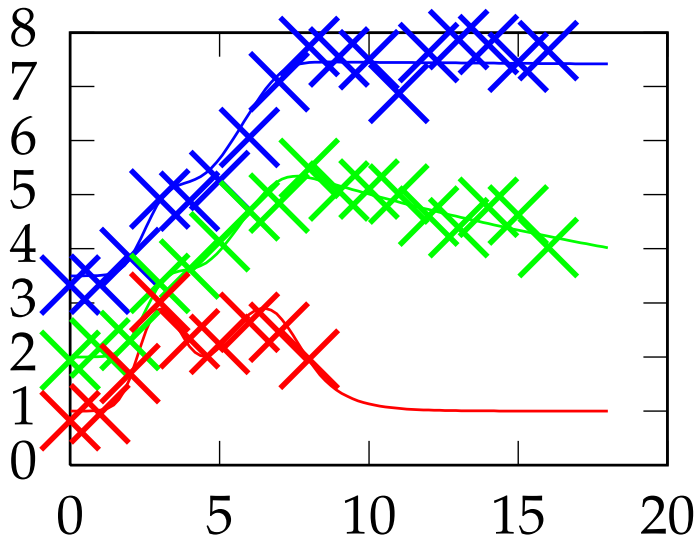

4
3
2
1
0
-1
-2
0
5
10
15
20

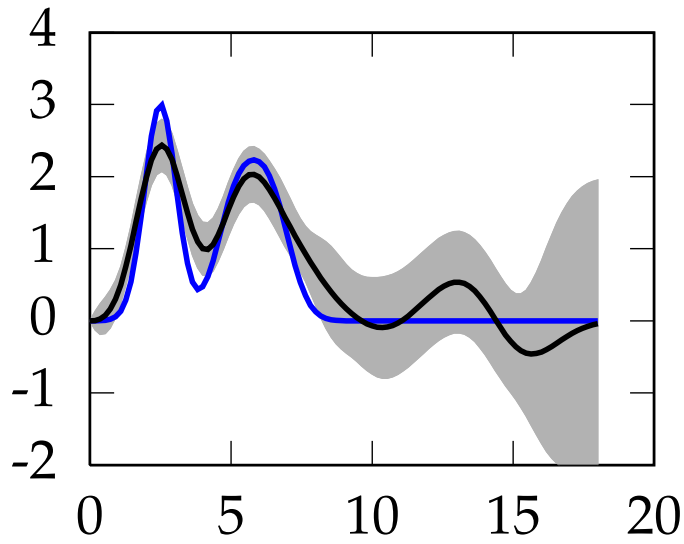

8
7
6
5
4
3
2
1
0
0
5
10
15
20

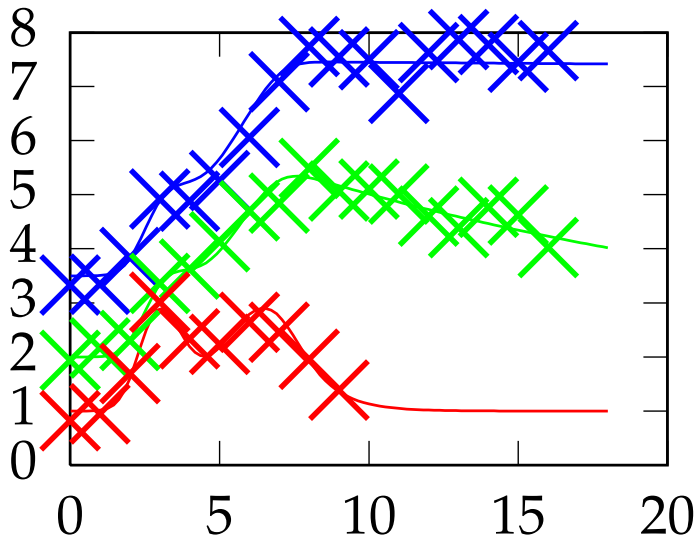

4
3
2
1
0
-1
-2
0
5
10
15
20

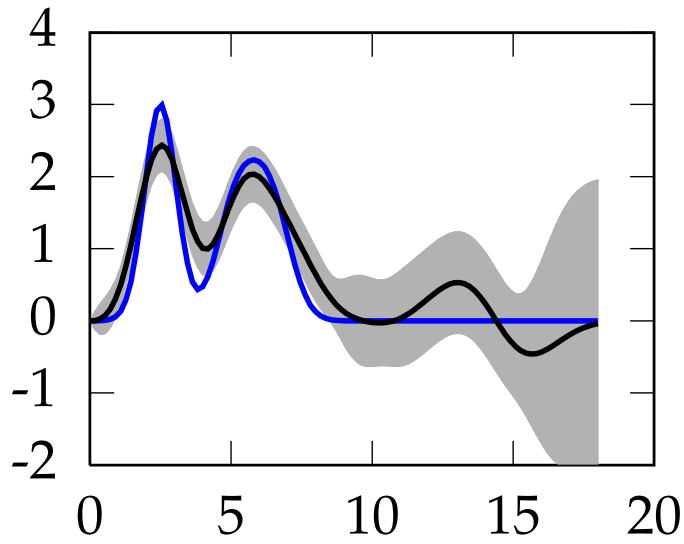

8
7
6
5
4
3
2
1
0
0
5
10
15
20

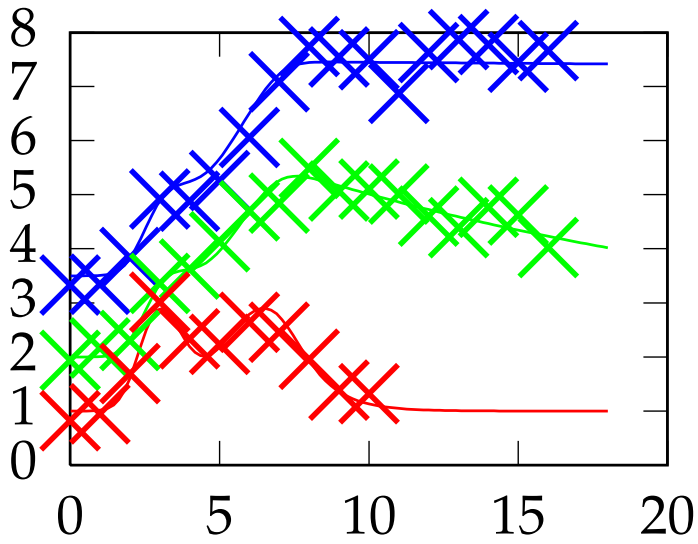

4
3
2
1
0
-1
-2
0
5
10
15
20

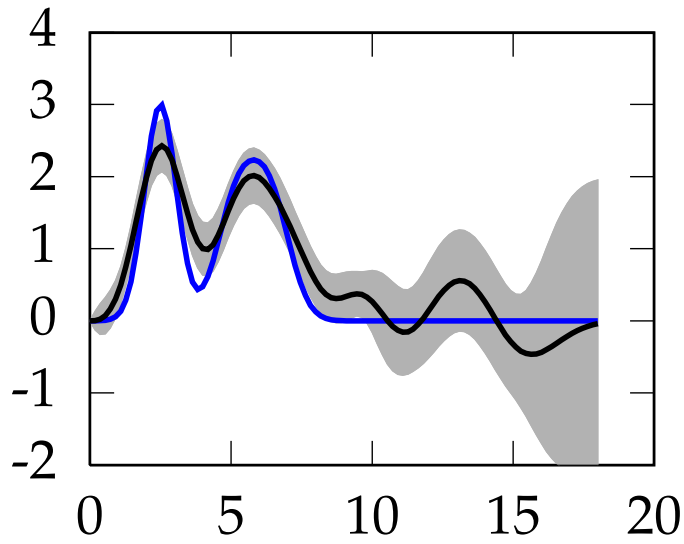

8
7
6
5
4
3
2
1
0
0
5
10
15
20

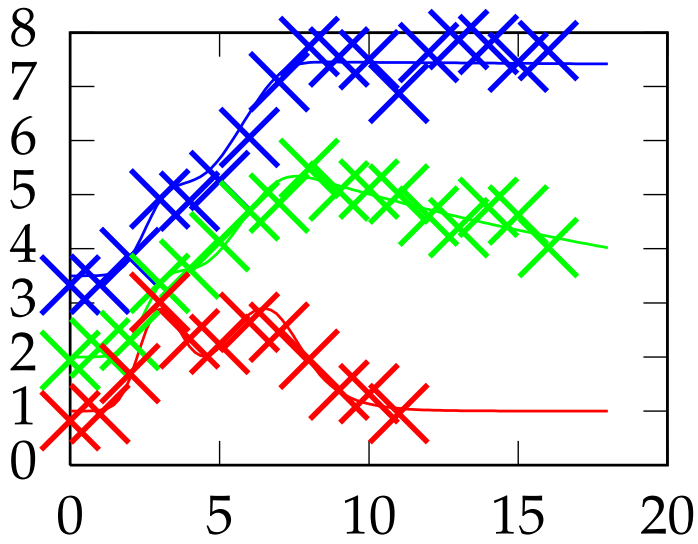

4
3
2
1
0
-1
-2
0
5
10
15
20

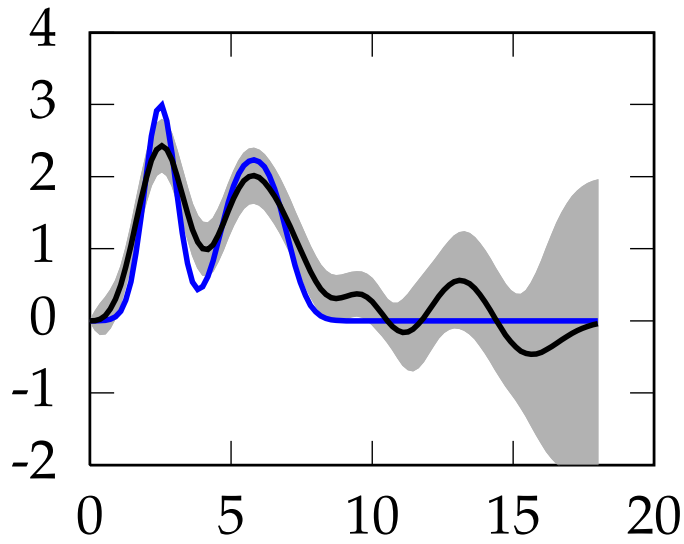

8
7
6
5
4
3
2
1
0
0
5
10
15
20

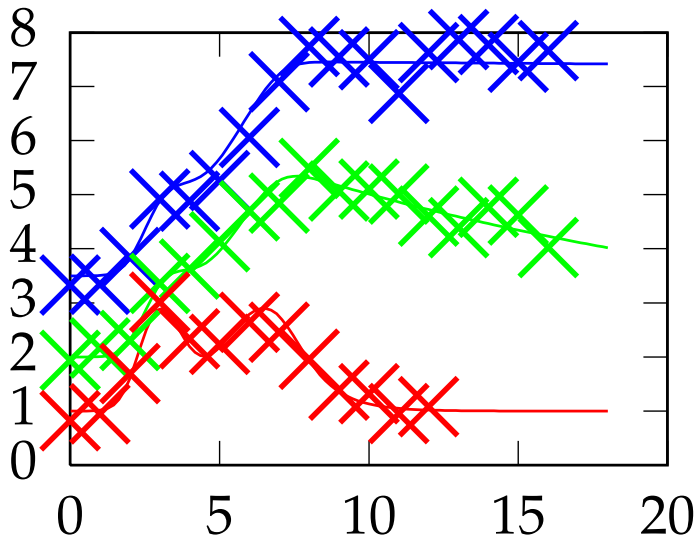

4
3
2
1
0
-1
-2
0
5
10
15
20

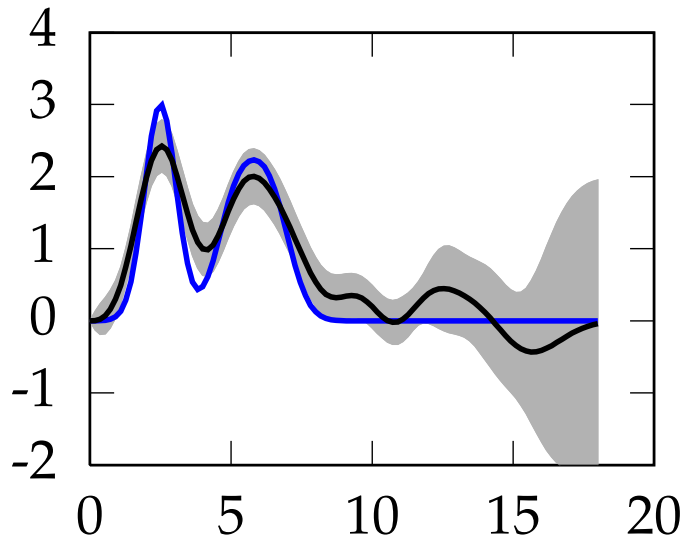

8
7
6
5
4
3
2
1
0
0
5
10
15
20

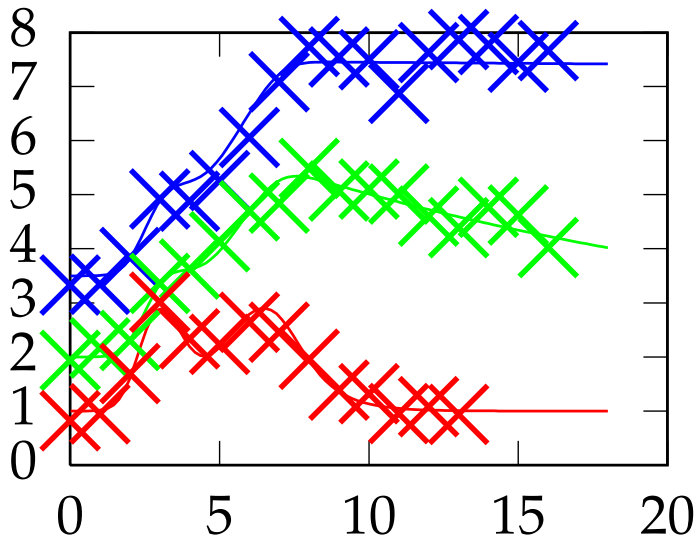

4
3
2
1
0
-1
-2
0
5
10
15
20

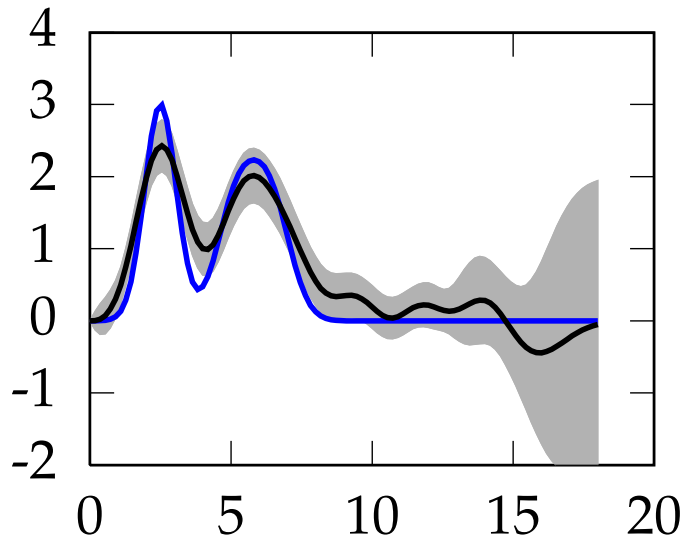

8
7
6
5
4
3
2
1
0
0
5
10
15
20

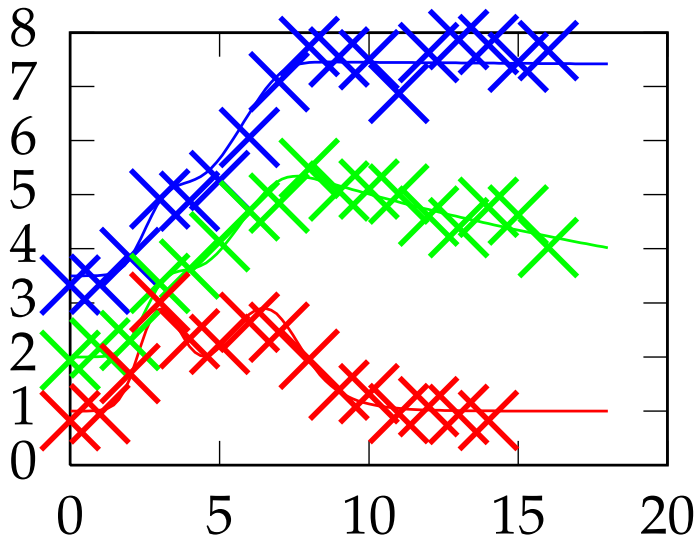

4
3
2
1
0
-1
-2
0
5
10
15
20

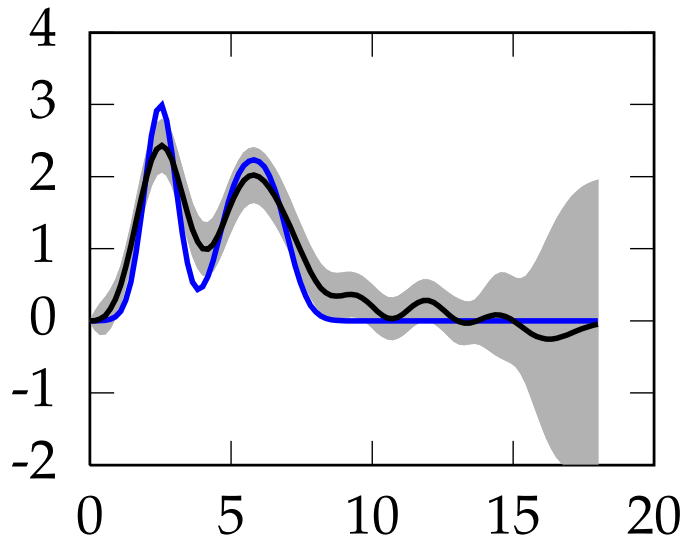

8
7
6
5
4
3
2
1
0
0
5
10
15
20

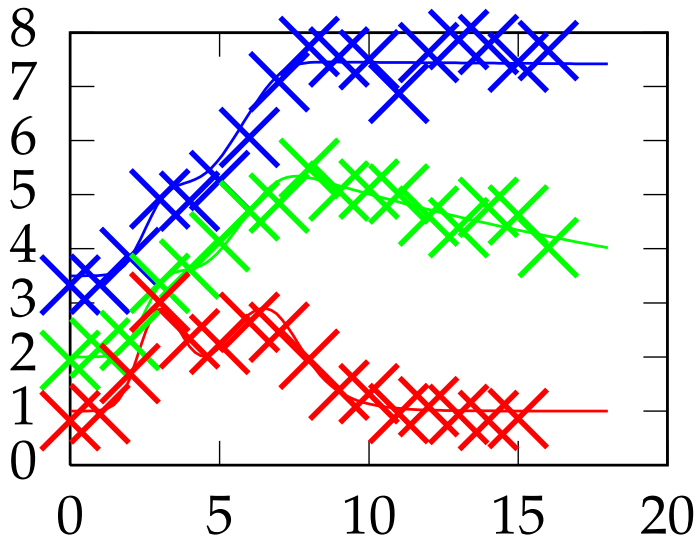

4
3
2
1
0
-1
-2
0
5
10
15
20

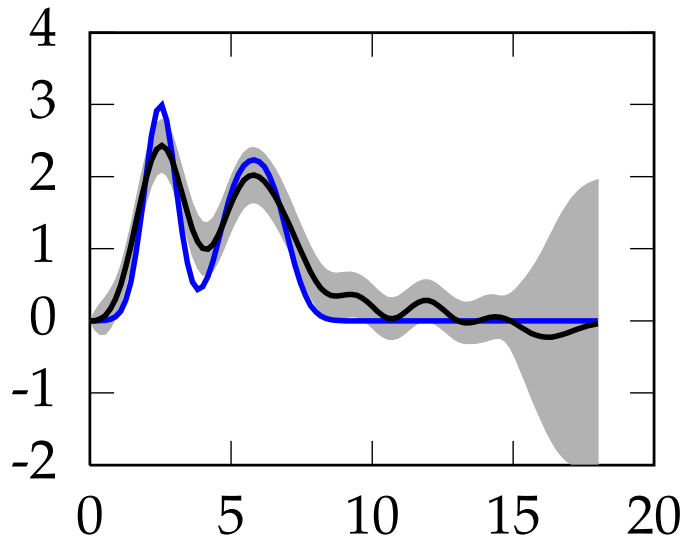

8
7
6
5
4
3
2
1
0
0
5
10
15
20

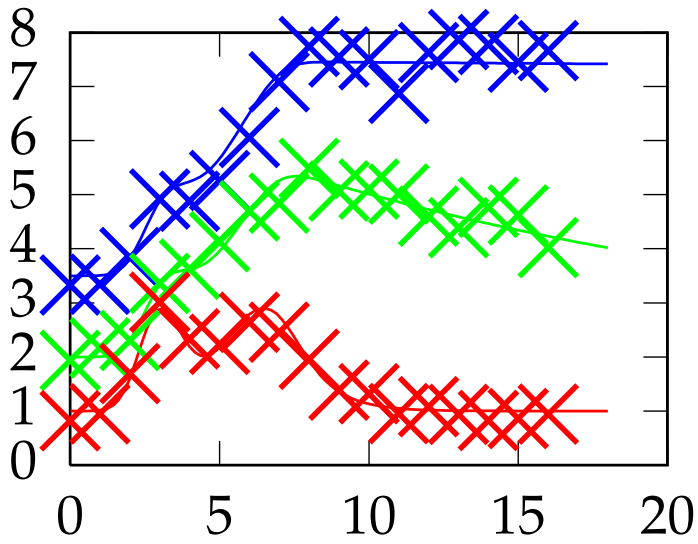

4
3
2
1
0
-1
-2
0
5
10
15
20

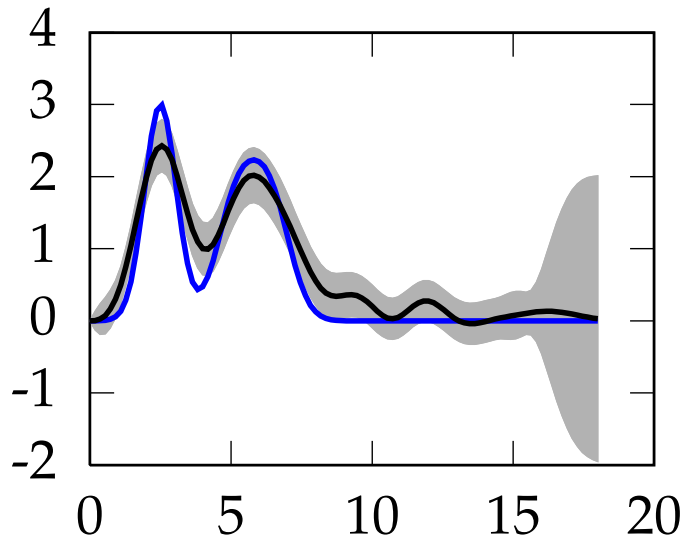

8
7
6
5
4
3
2
1
0
0
5
10
15
20


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


8
7
6
5
4
3
2
1
0
0
5
10
15
20


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


8
7
6
5
4
3
2
1
0
0
5
10
15
20


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

Radiation Damage in the Cell

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

p53 “Guardian of the Cell”

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

p53 DNA Damage Repair

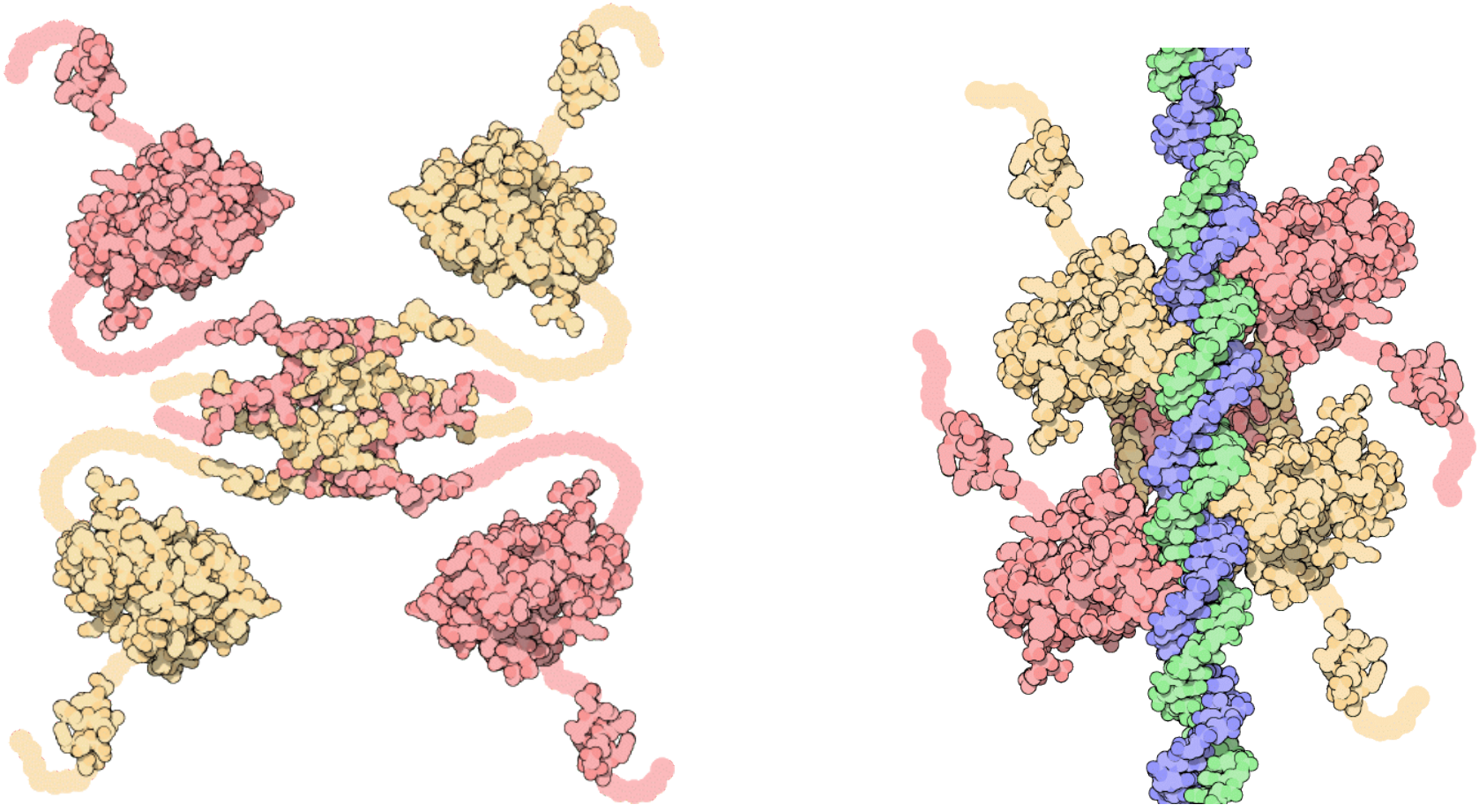


Figure : p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the"Molecule of the Month" feature).

p53

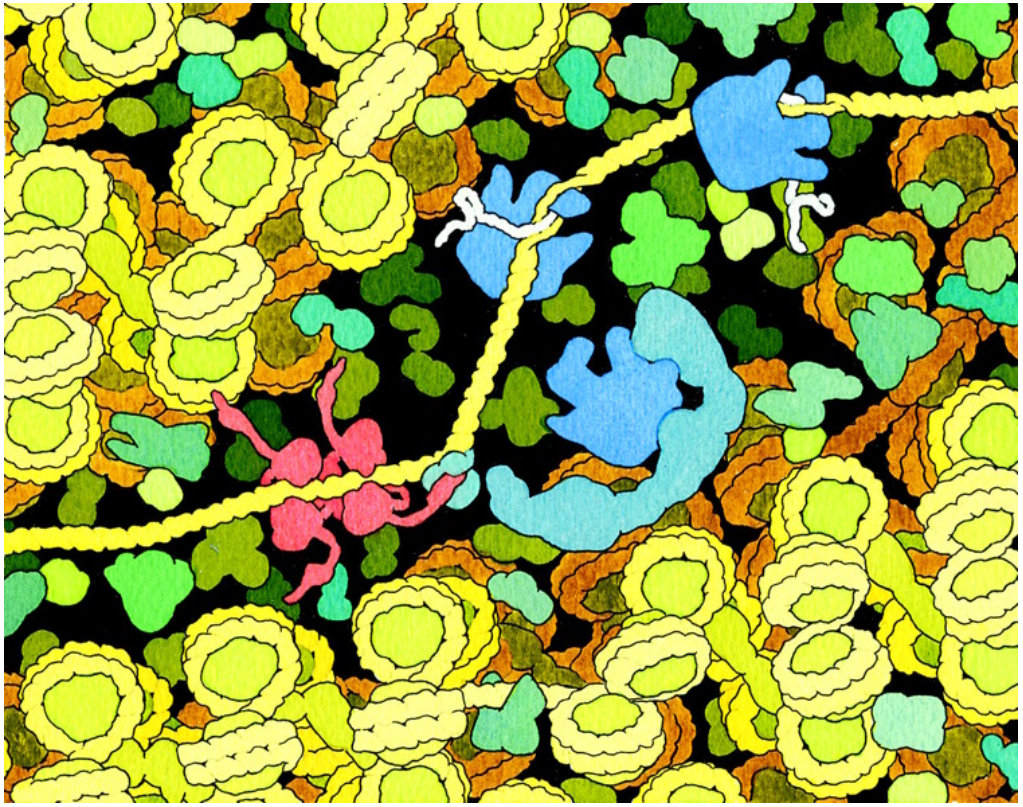


Figure : Repair of DNA damage by p53. Image from Goodsell (1999).

Some p53 Targets

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

Modelling Assumption

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

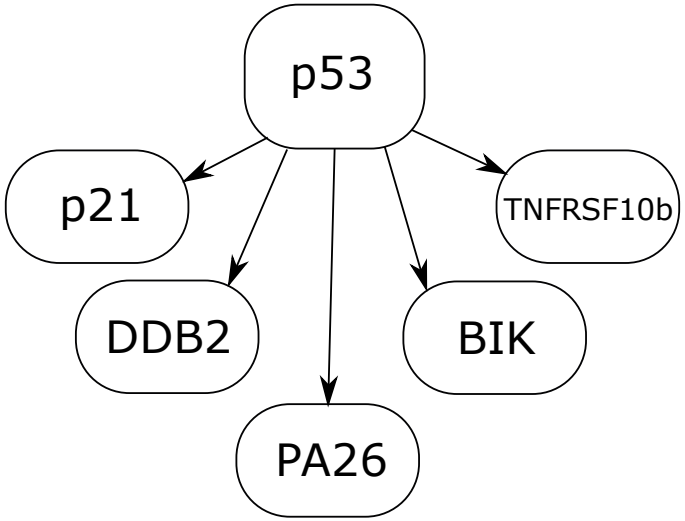


Figure : p53 SIM network motif as modelled by Barenco et al. 2006.

Ordinary Differential Equation Model

- First Order Differential Equation

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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?

p53 Results with GP

BIOINFORMATICS

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

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

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

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

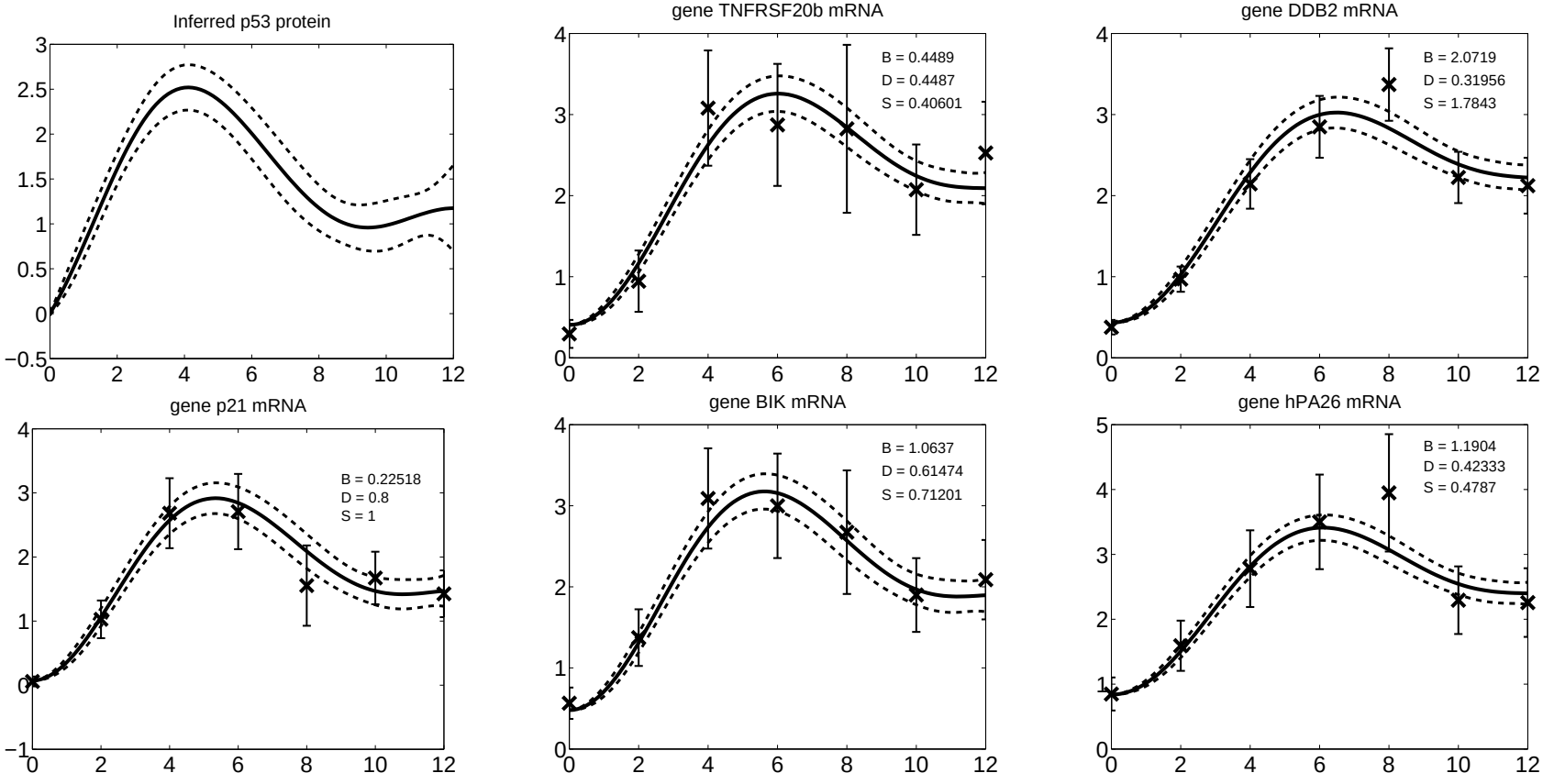
ABSTRACT

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

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

p53 Results with GP

(Gao et al., 2008)

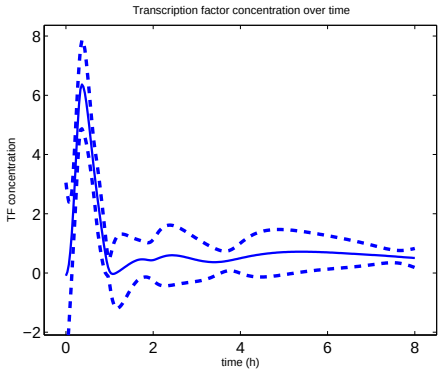


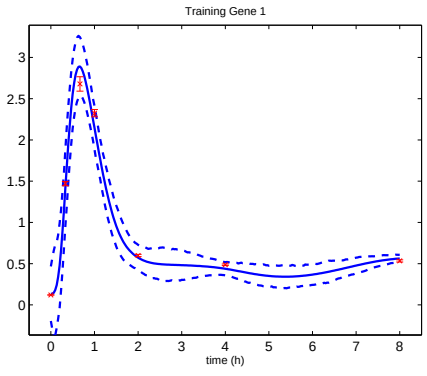
Ranking with ERK Signalling

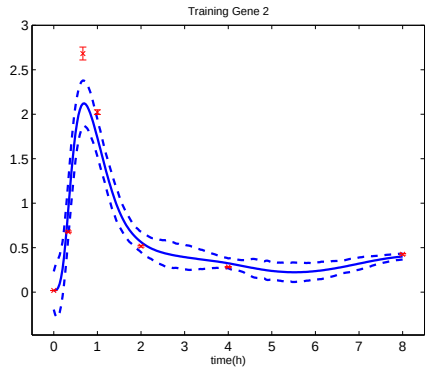
- Target Ranking for Elk-1.
- Elk-1 is phosphorylated by ERK from the EGF signalling pathway.
- Predict concentration of Elk-1 from known targets.
- Rank other targets of Elk-1.

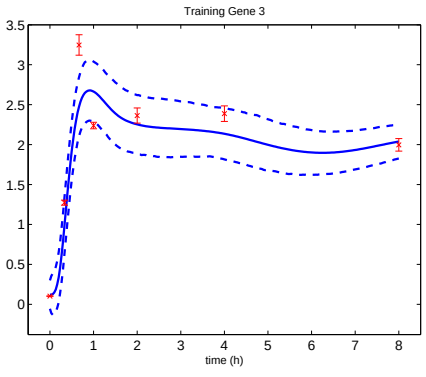
Elk-1 (MLP covariance)

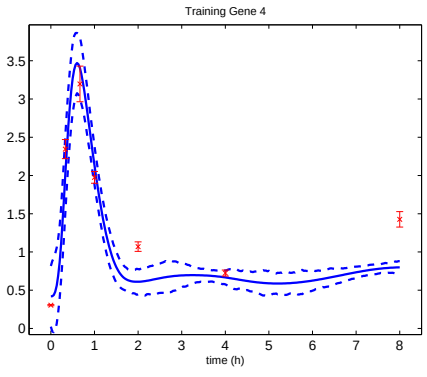
Jennifer Withers

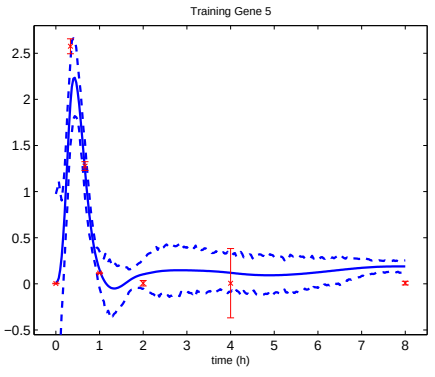






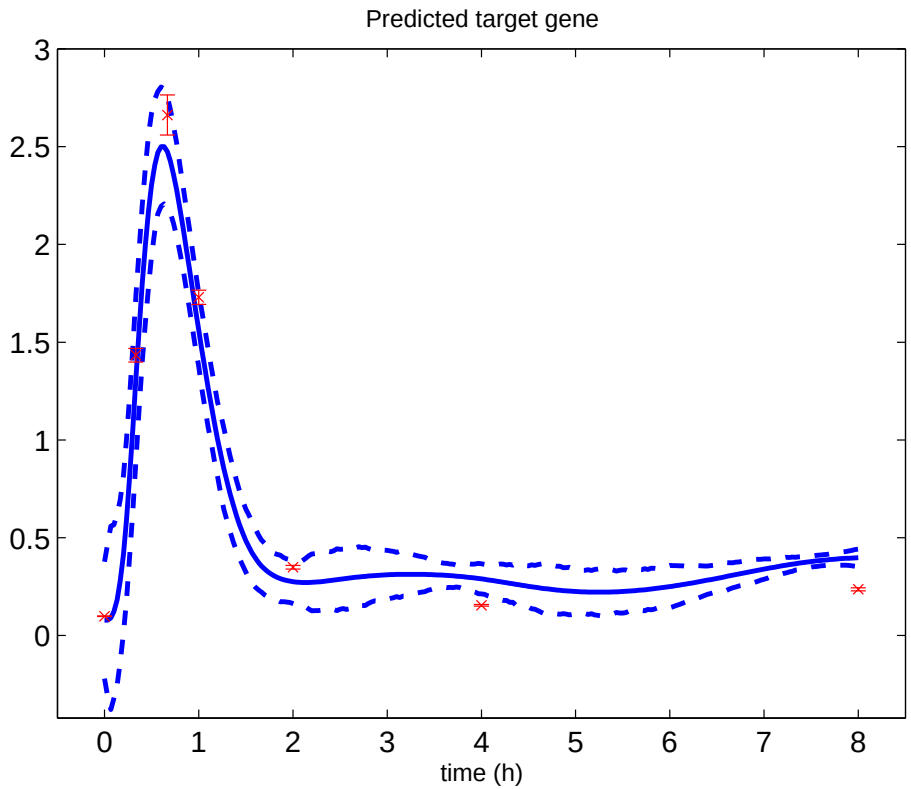


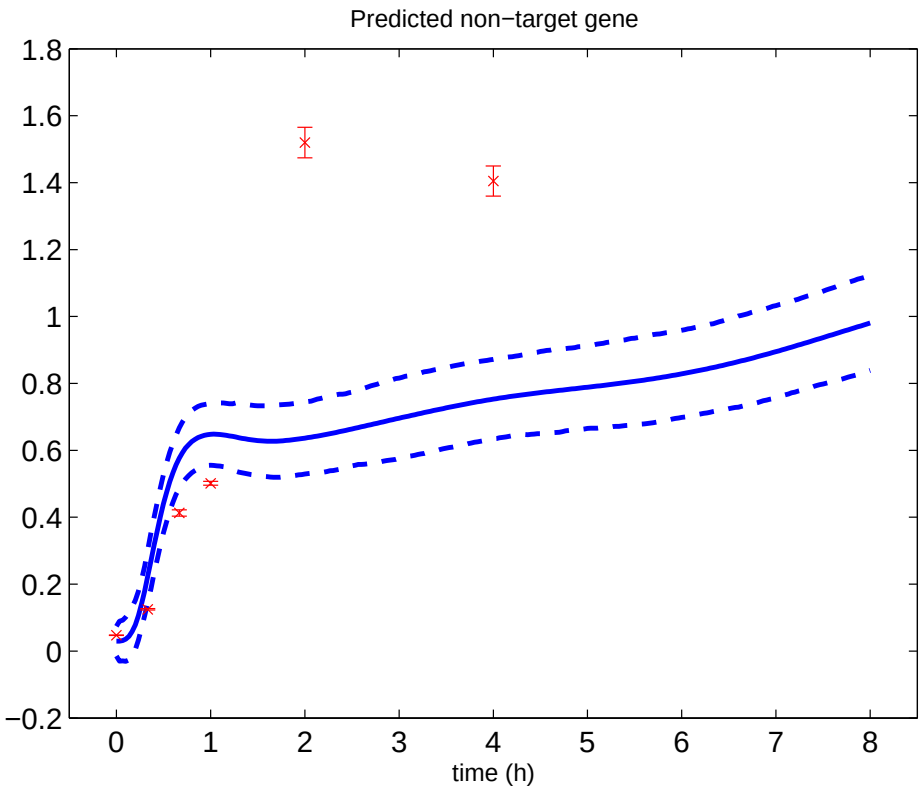




Elk-1 target selection

Fitted model used to rank potential targets of Elk-1





PNAS

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

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

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

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

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

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

Cascaded Differential Equations

(Honkela et al., 2010)

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

Covariance for Translation/Transcription Model

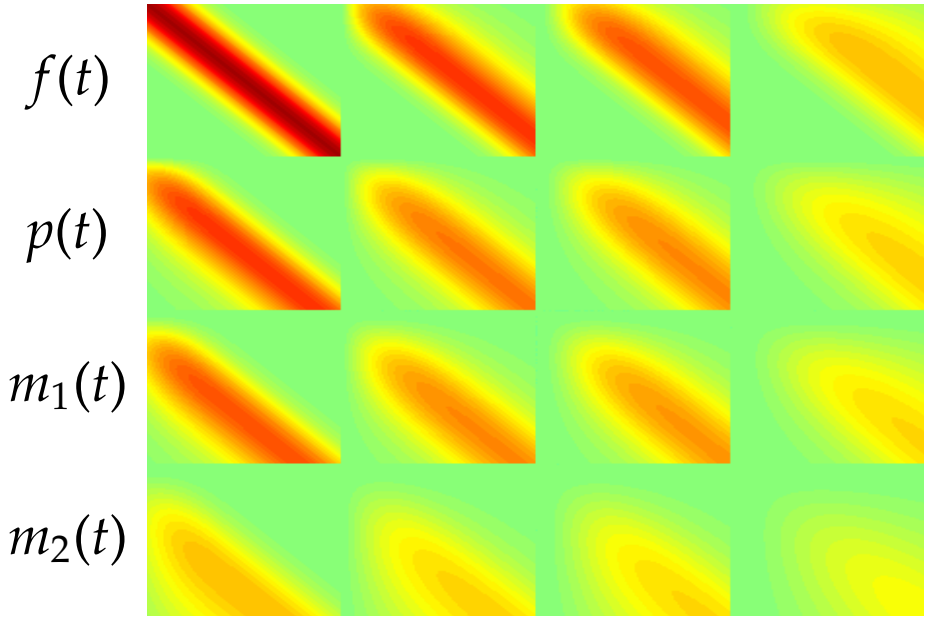
RBF covariance function for $f(t)$

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

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

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

δ	d_1	s_1	d_2	s_2
1	5	5	0.5	0.5



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

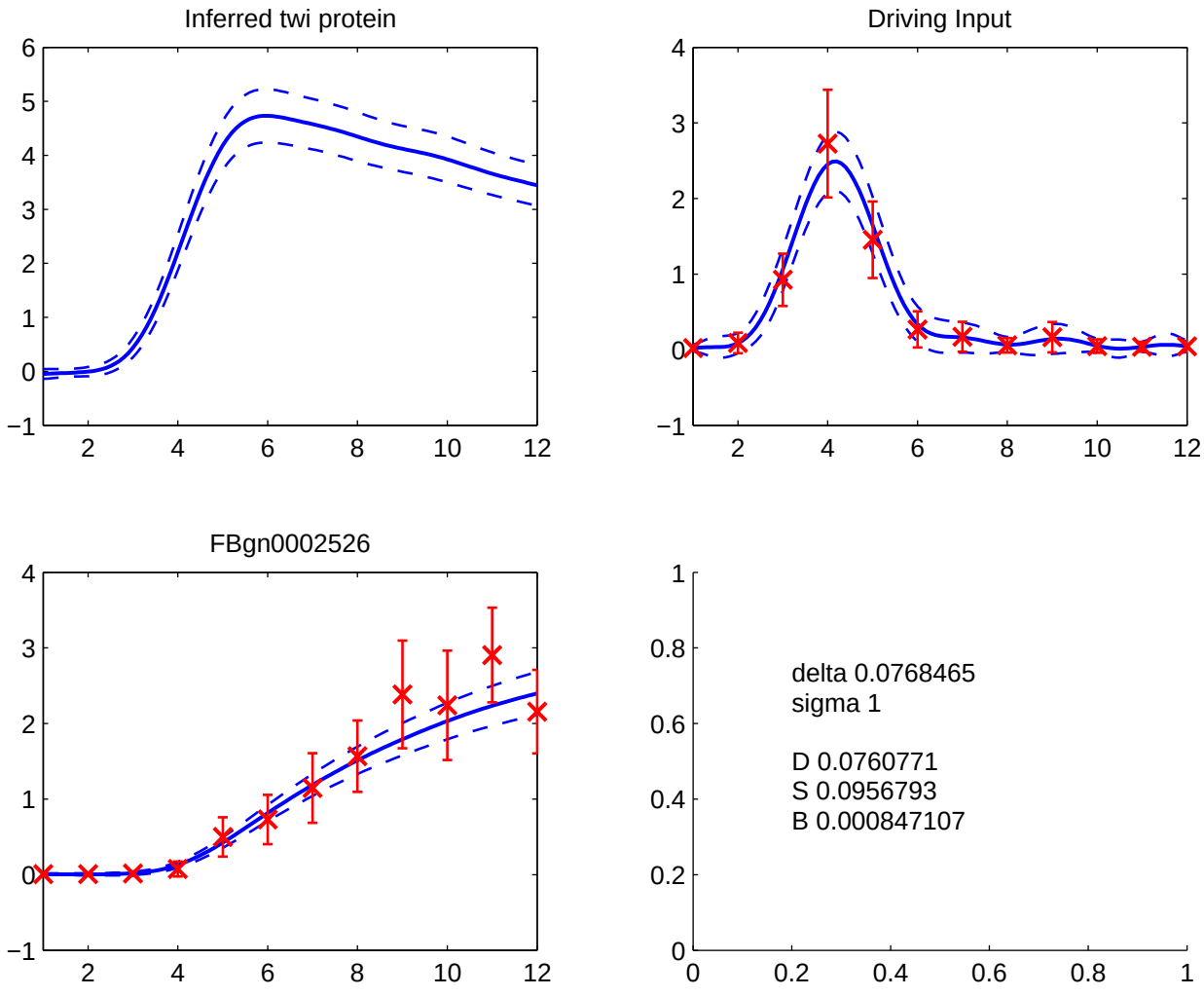


Figure : Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

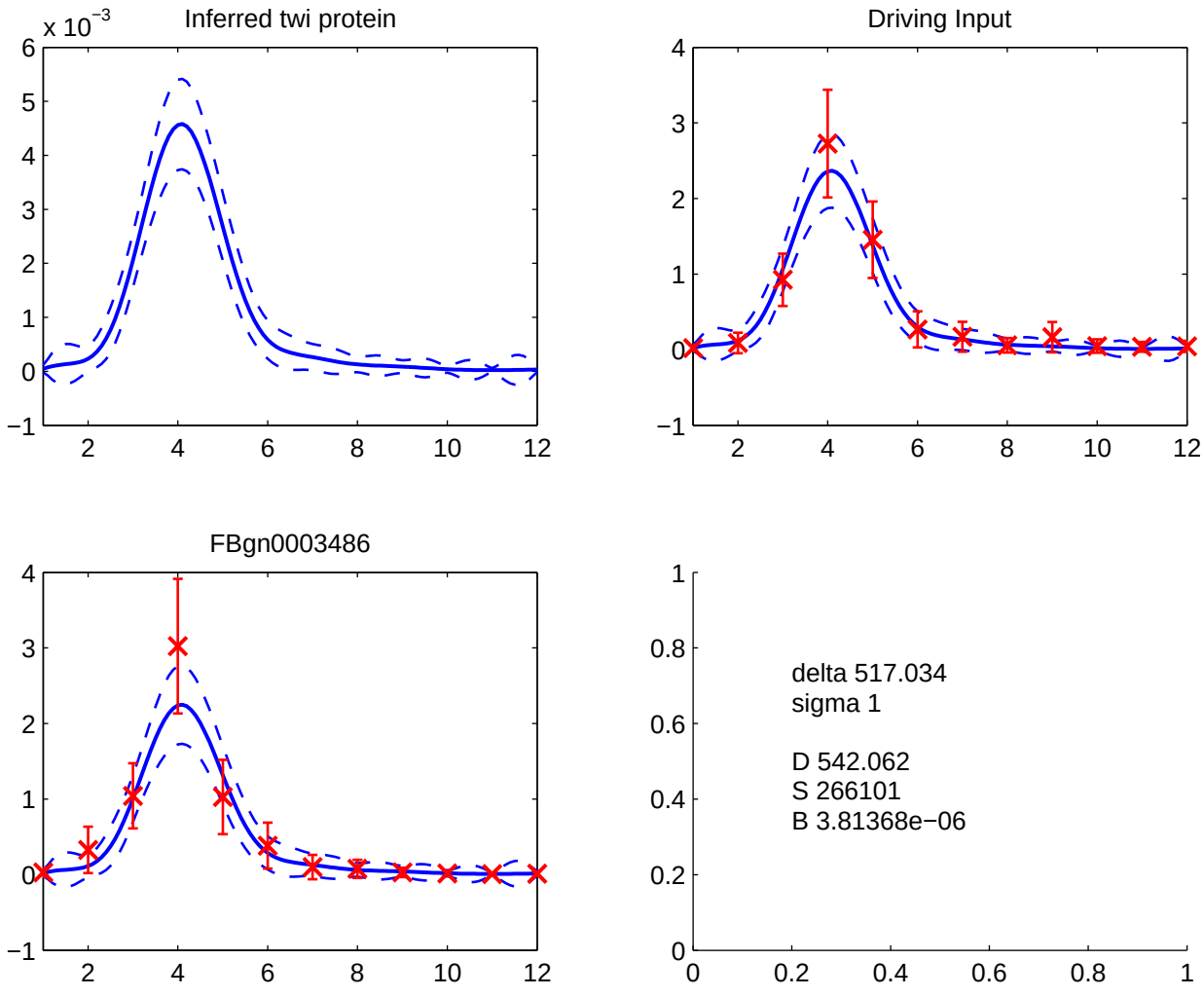


Figure : Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

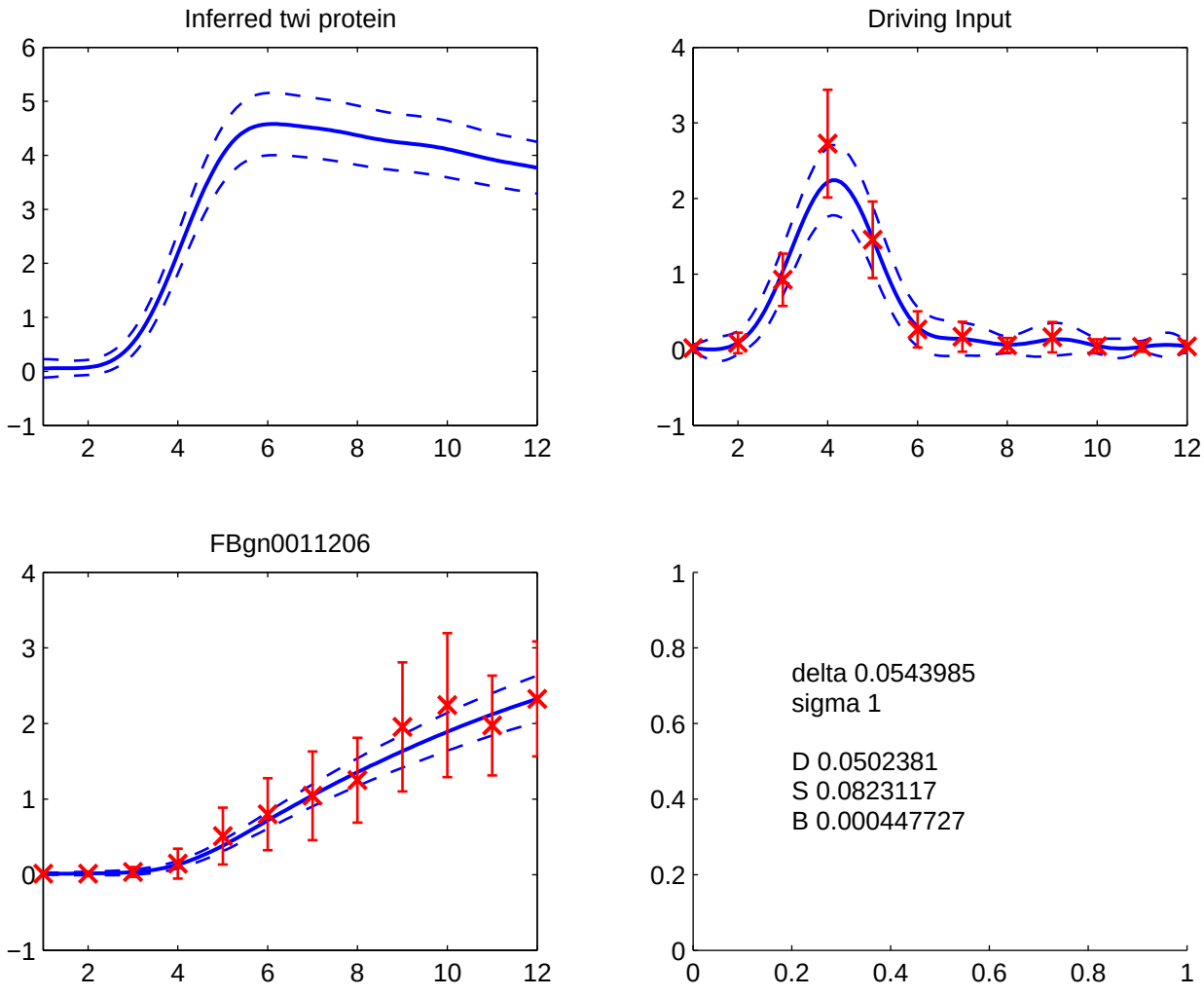


Figure : Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

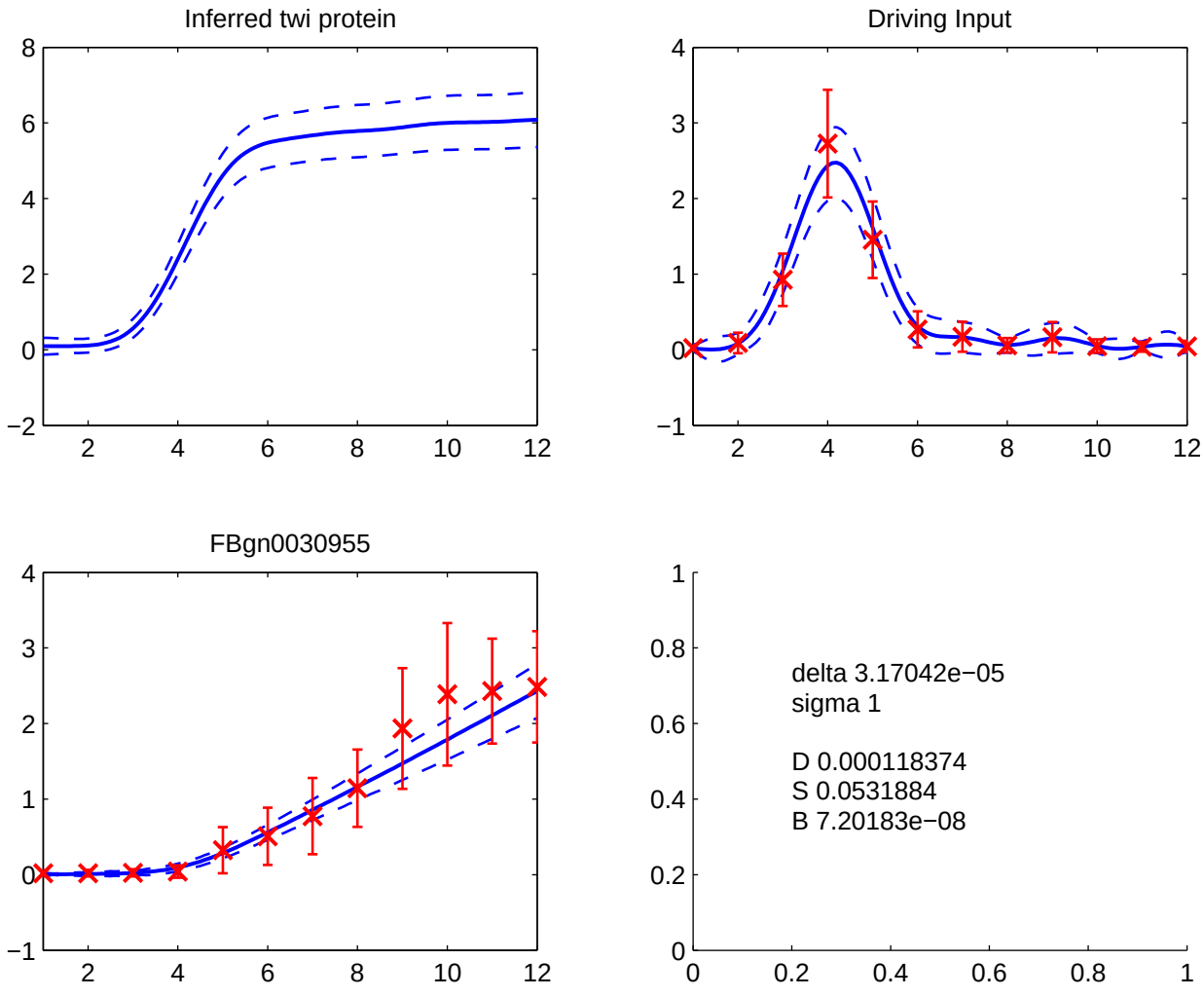


Figure : Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

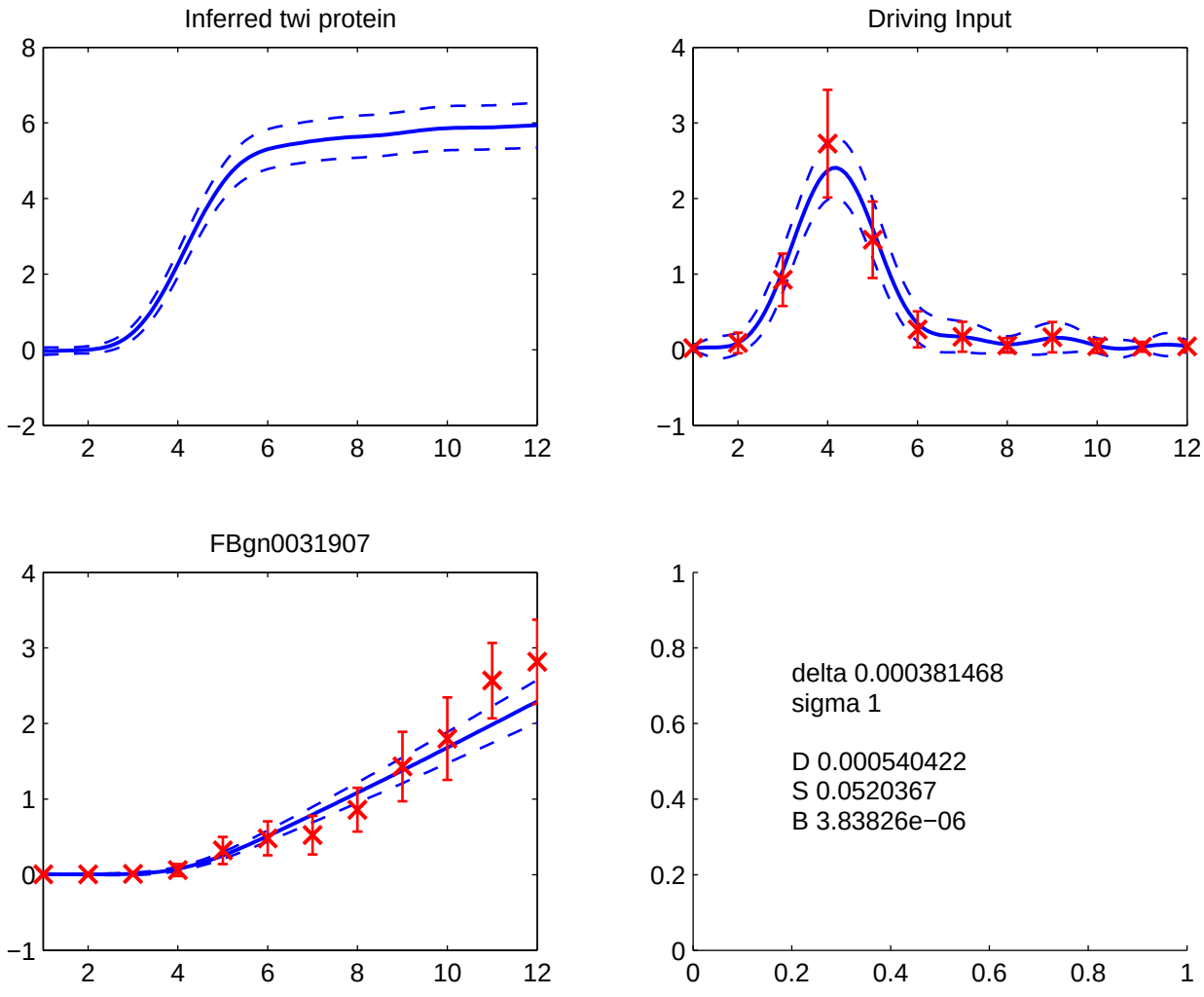


Figure : Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

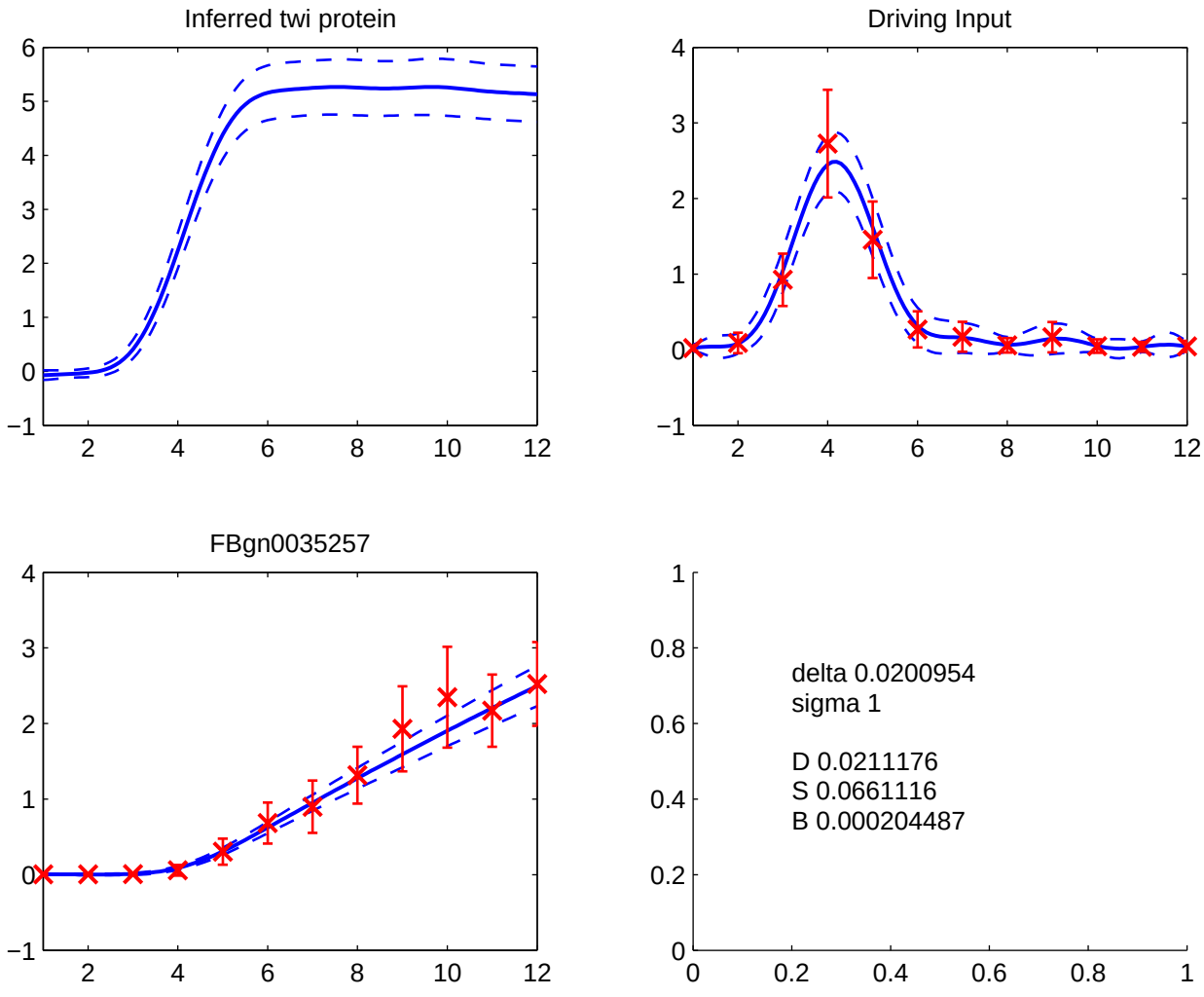


Figure : Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

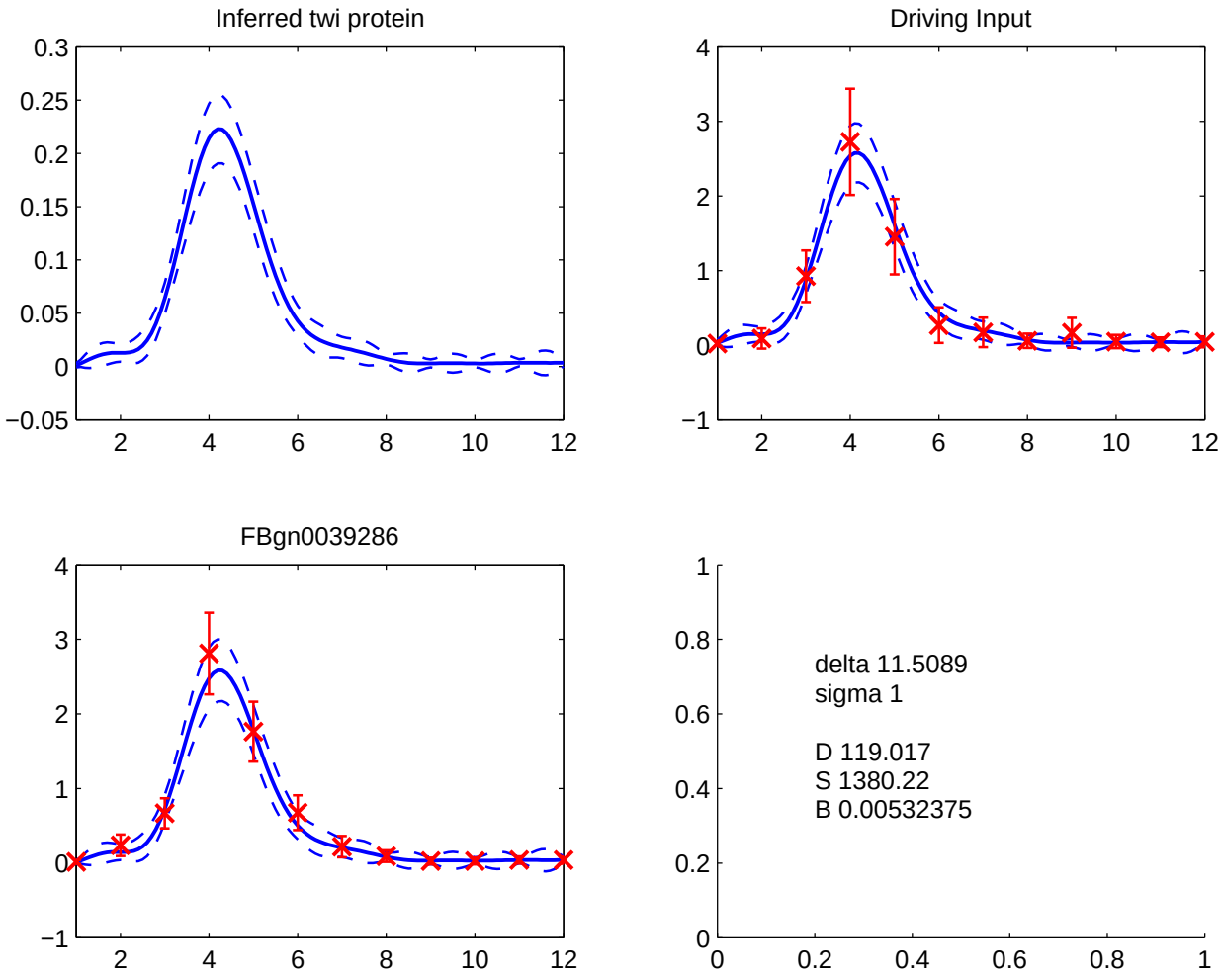
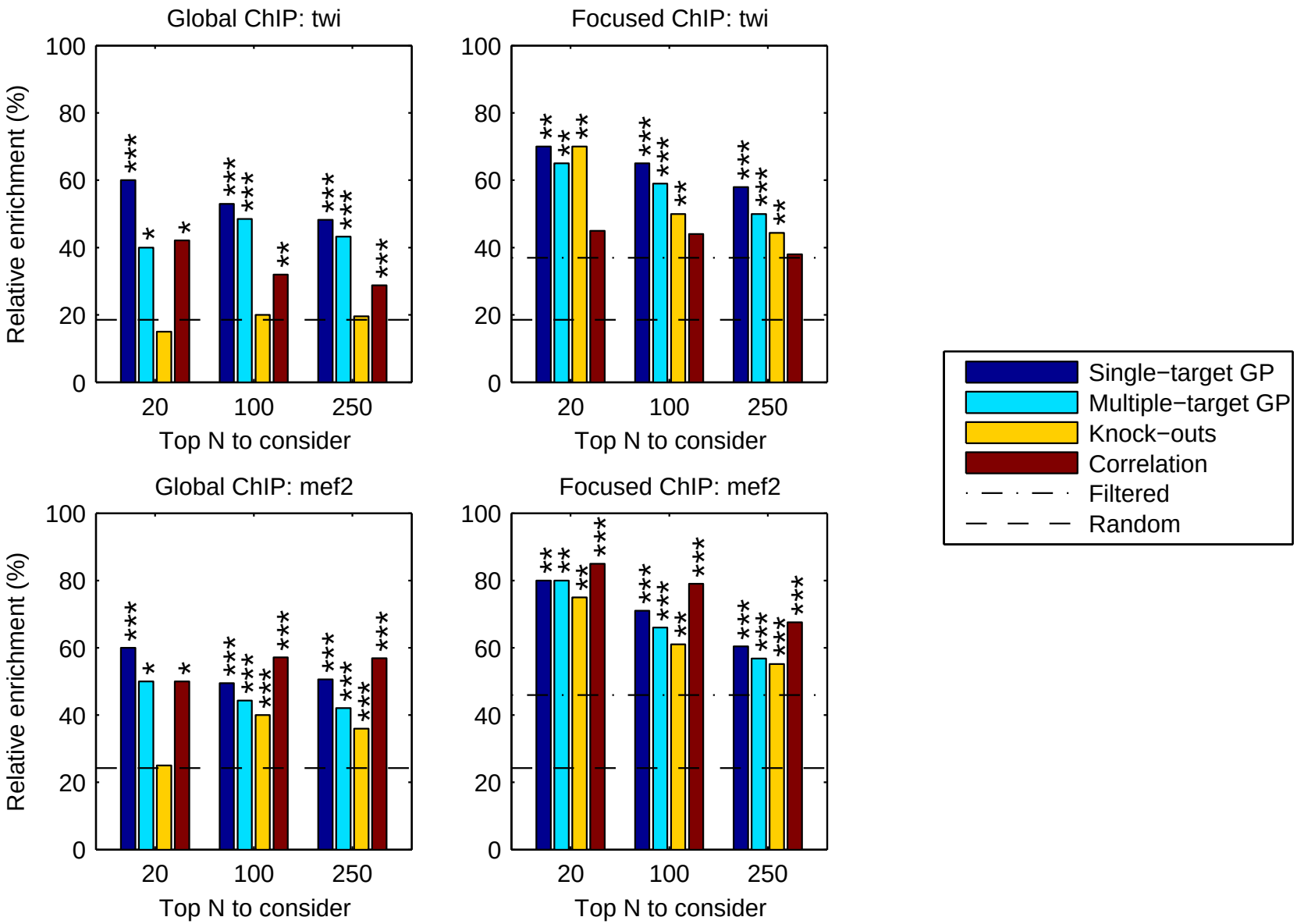


Figure : Model for flybase gene identity FBgn0039286.

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.

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.

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

- 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

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

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 $T_q = \kappa_q t$.

Covariance Function

Mauricio Alvarez

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

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

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

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

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).[2]
- 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).

Jura Results

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±0.0133	0.4505±0.0126	0.5
Cu	15.9357±0.0907	7.1677±0.2266	7.8
Pb	22.9141±0.6076	10.1097±0.2842	10.7
Co	2.0735±0.1070	1.7546±0.0895	1.5

Convolutions and Computational Complexity

Mauricio Alvarez

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

$$x_i(t) = \int f(u)\, k_i(u-t)\, \mathrm{d}u + h_i(t)$$

$$x_i(t) = \int_0^t f(u)\, g_i(u)\, \mathrm{d}u + h_i(t)$$

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

Independence Assumption

Mauricio Alvarez

- Can assume conditional independence given 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 $O\left(N^3dk^2\right)$ complexity, $O\left(N^2dk\right)$ storage.
  - Can also do a FITC style approximation (Snelson and Ghahramani, 2006).
  - Reduces to $O\left(Ndk^2\right)$ complexity, $O(Ndk)$ storage.

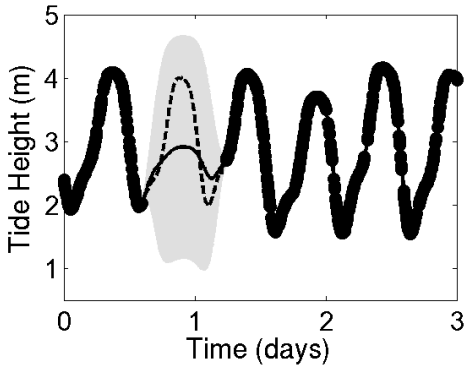
Tide Sensor Network

Mauricio Alvarez

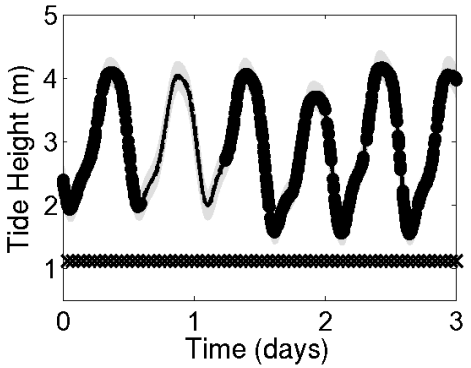
- Network of tide height sensors in the solent — tide heights are correlated.
- Data kindly provided by Alex Rogers (see Osborne et al., 2008).
- $d = 3$ and $N = 1000$ of the 4320 for the training set.
- Simulate sensor failure by knocking out onse sensor for a given time.
- For the other two sensors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

Tide Height Results

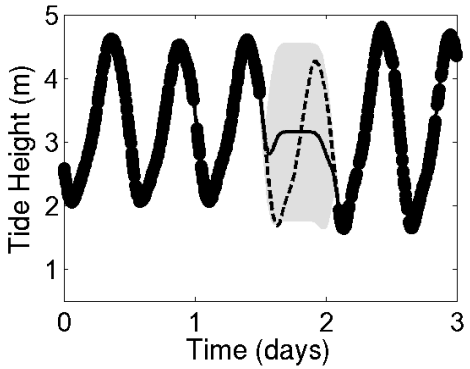
Mauricio Alvarez



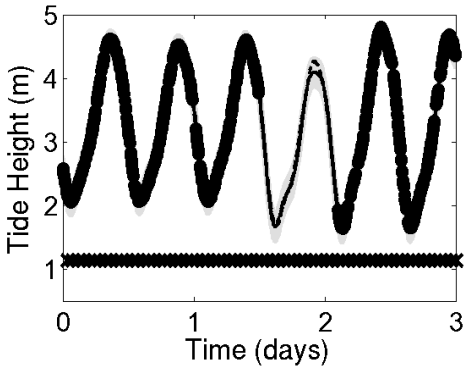
(a) Bramblemet Independent



(b) Bramblemet PITC



(c) Cambermet Independent



(d) Cambermet PITC

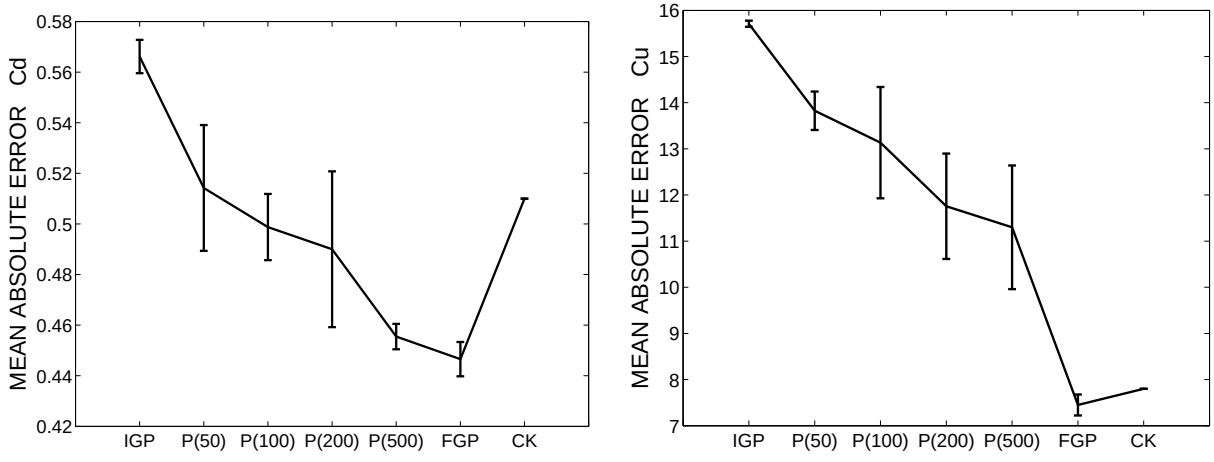
Cokriging Jura

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).

Swiss Jura Results

Mauricio Alvarez



(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.

MAP-Laplace Approximation

Laplace's method: approximate posterior mode as Gaussian

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

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

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

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

MAP-Laplace Approximation

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

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

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

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

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

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

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

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

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

Model Parameter Estimation

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

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

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

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

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

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

SOS Response

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

LexA Experimental Description

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

Khanin et al. Model

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

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

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

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

Khanin et al. Results

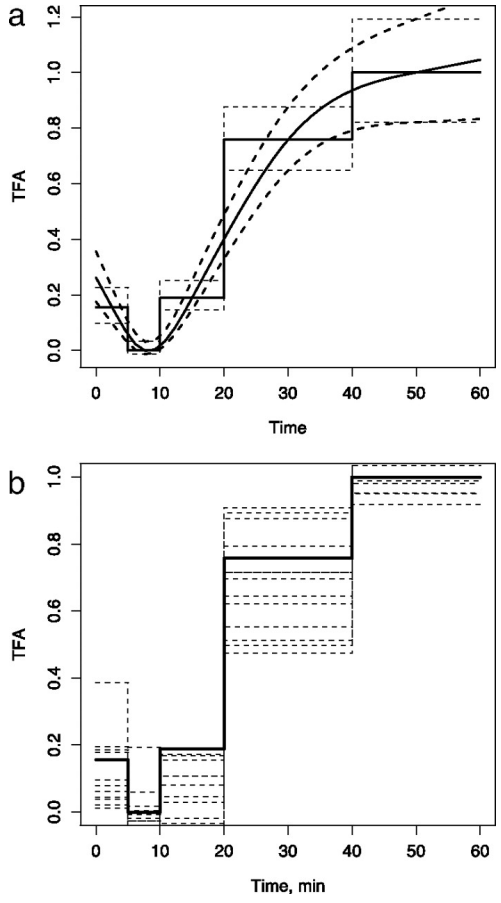


Figure : Fig. 2 from Khanin et al. (2006): Reconstructed activity level of master repressor LexA, following a UV dose of 40 J/m2.

Khanin et al. Results

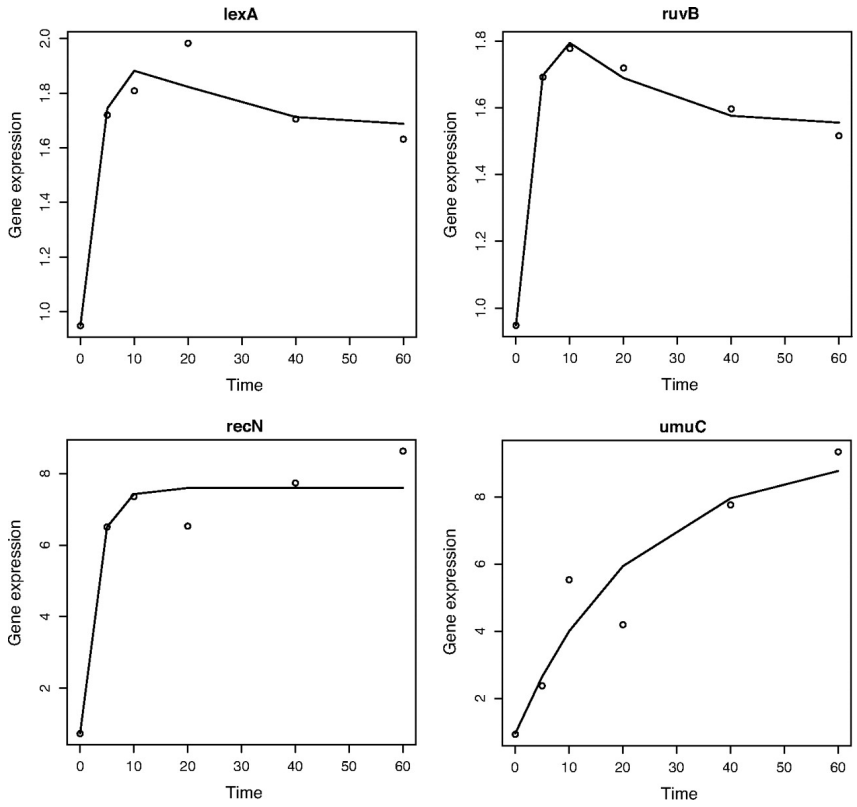


Figure : Fig. 3 from Khanin et al. (2006): Reconstructed profiles for four genes in the LexA SIM.

Repression Model

Pei Gao

- We can use the same model of repression,

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

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

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

Results for the repressor LexA

Pei Gao

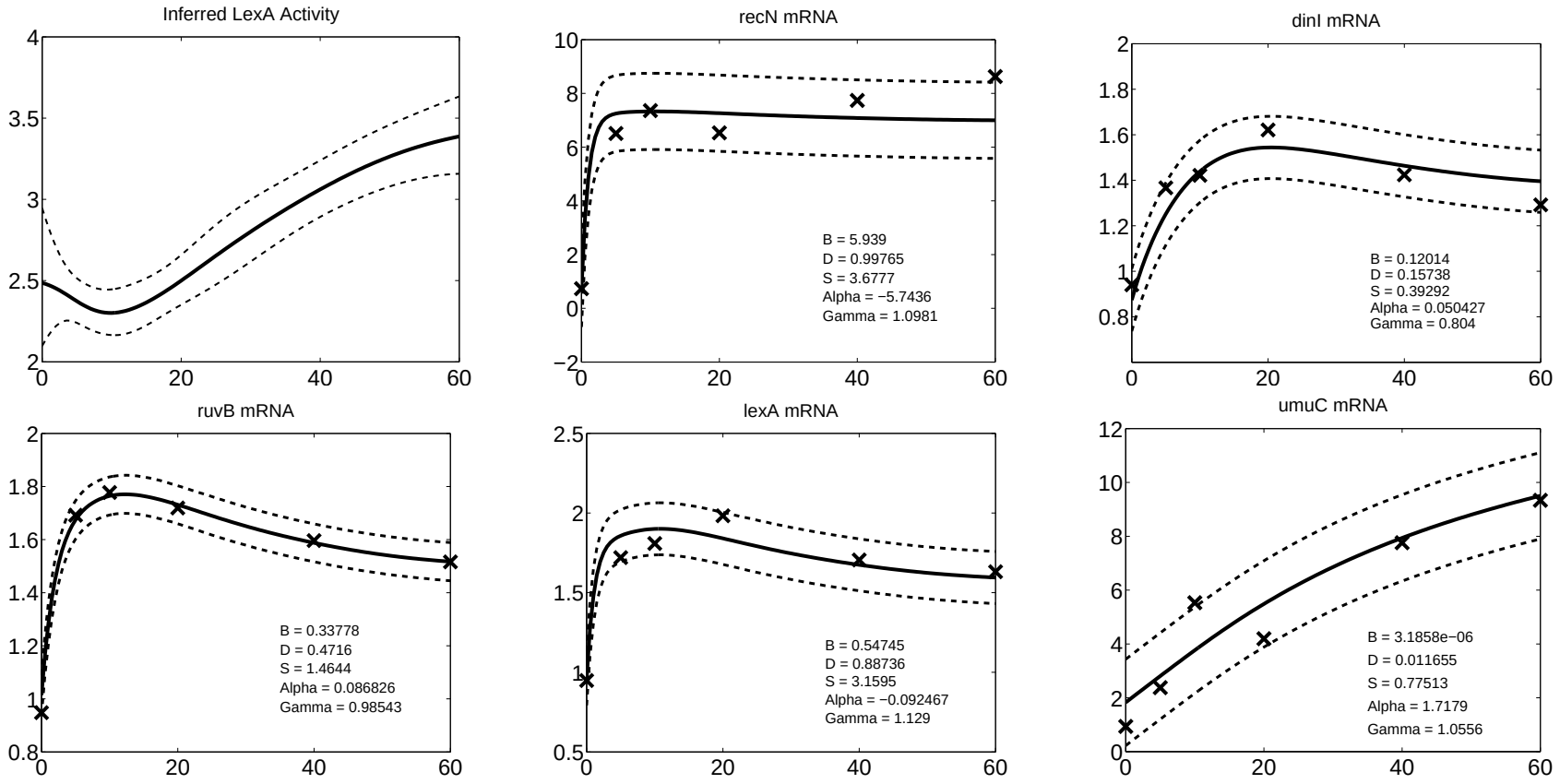


Figure : Our results using an MLP kernel. From Gao et al. (2008).

Use Samples to Represent Posterior

Michalis Titsias

- Sample in Gaussian processes

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

- Likelihood relates GP to data through

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

- We use *control points* for fast sampling.

MCMC for Non Linear Response

The Metropolis-Hastings algorithm

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

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

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

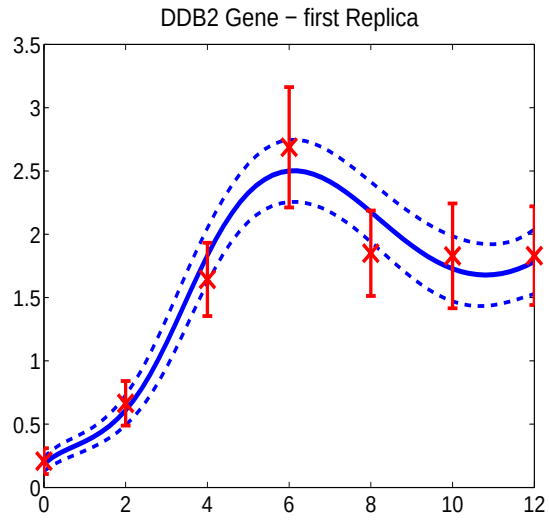
p53 System Again

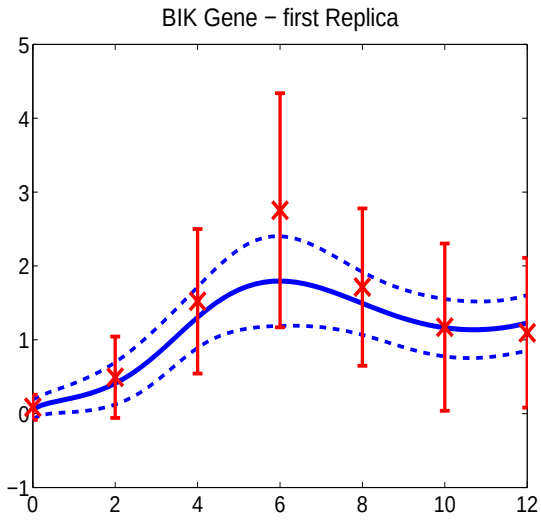
- One transcription factor (p53) that acts as an activator. We consider the Michaelis-Menten kinetic equation

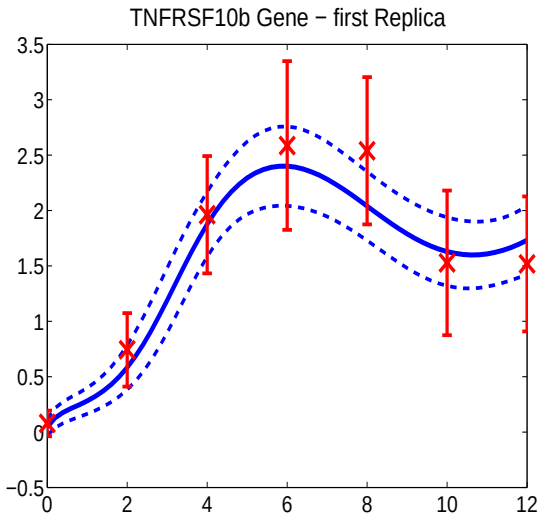
$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j \frac{\exp(p(t))}{\exp(p(t)) + \gamma_j} - d_j m_j(t)$$

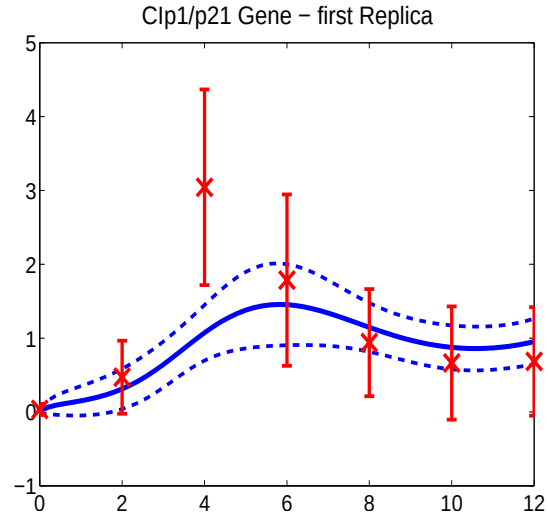
- We have 5 genes
- Gene expressions are available for $T = 7$ times and there are 3 replicas of the time series data
- TF ($\mathbf{p}$) is discretized using 121 points
- MCMC details:
  - 7 control points are used (placed in a equally spaced grid)
  - Running time 4/5 hours for 2 million sampling iterations plus burn in
  - Acceptance rate for $\mathbf{p}$ after burn in was between 15% – 25%

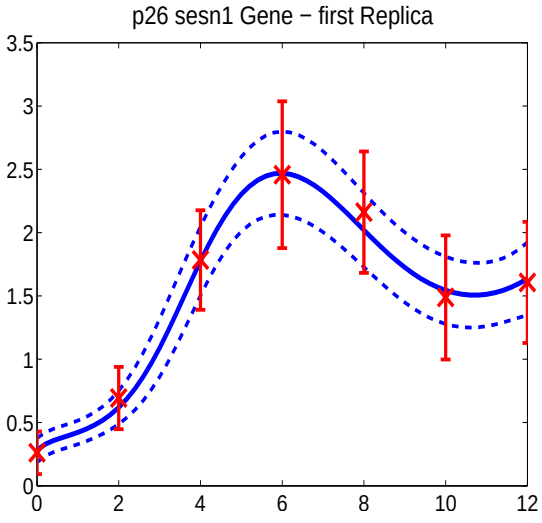
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica



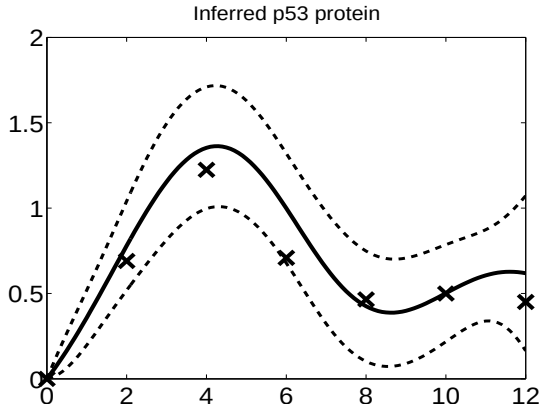


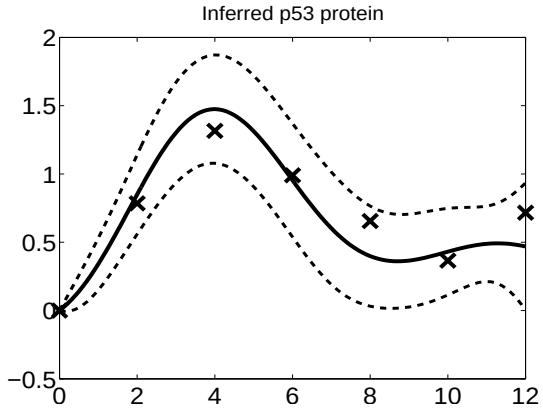


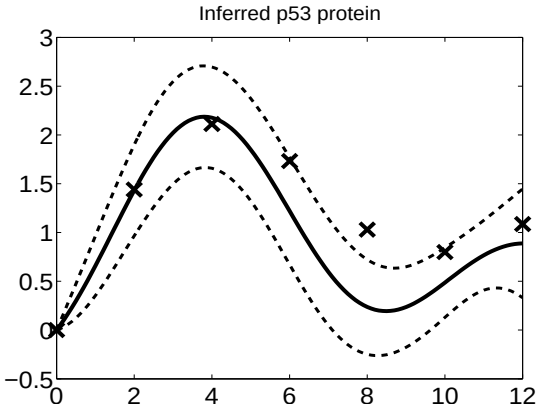




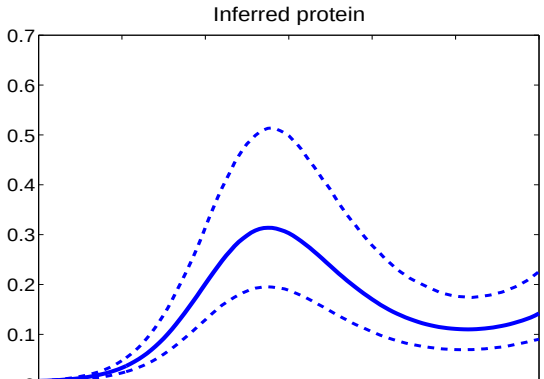
Data used by Barenco et al. (2006): Protein concentrations

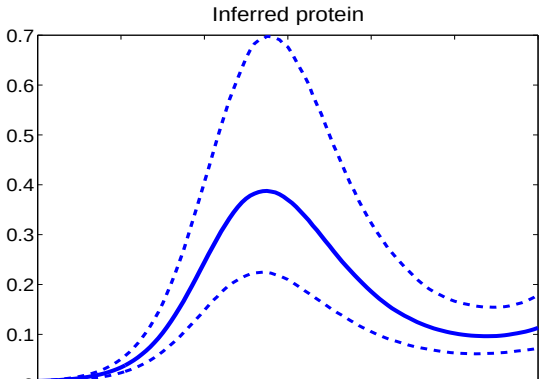


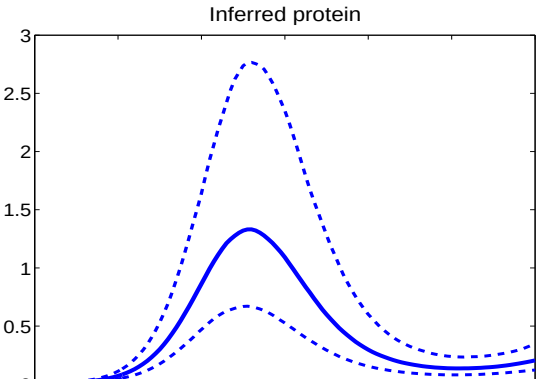




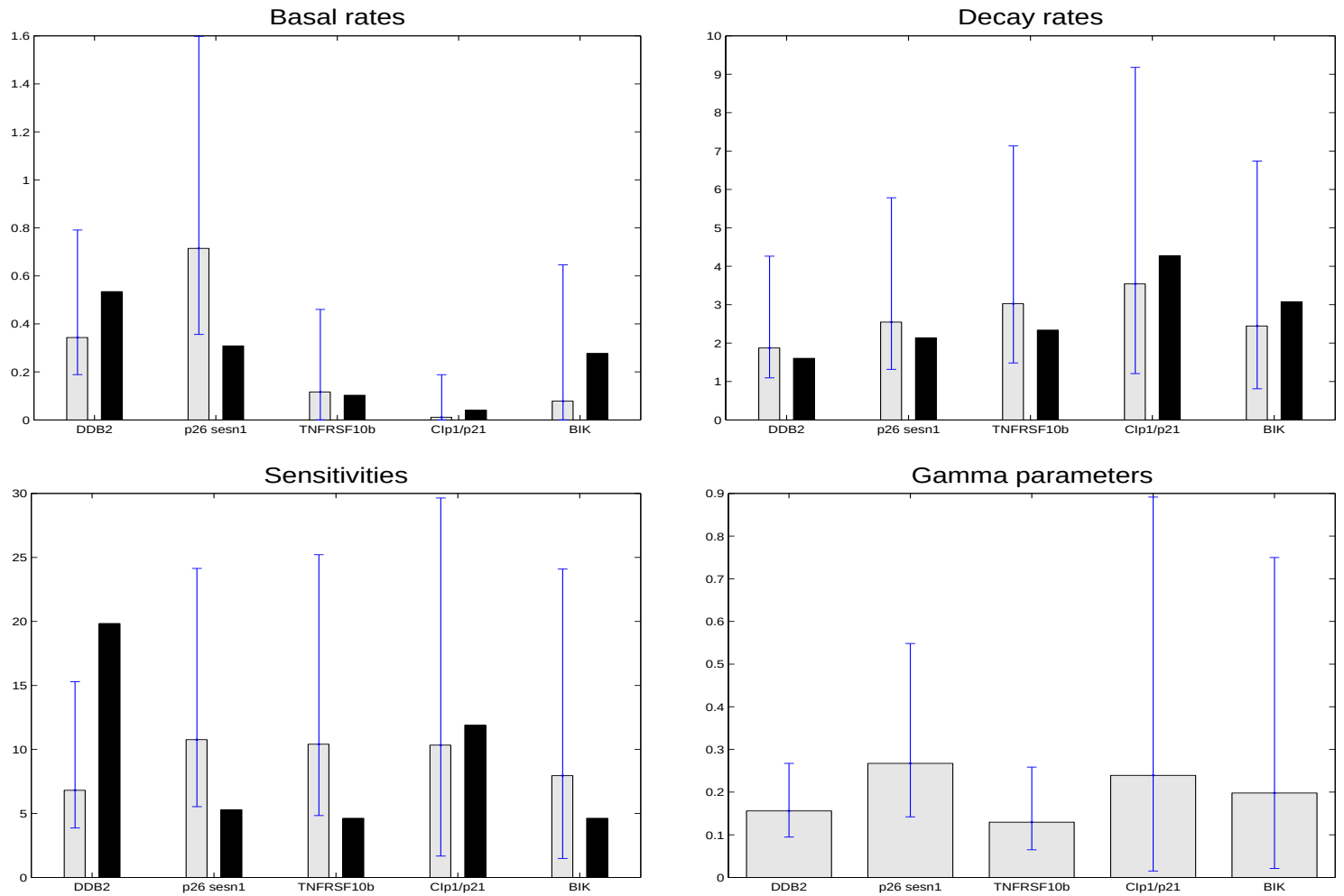
Linear model (Barenco et al. predictions are shown as crosses)







p53 Data Kinetic parameters



Our results (grey) compared with Barenco et al. (2006) (black).
Note that Barenco et al. use a linear model

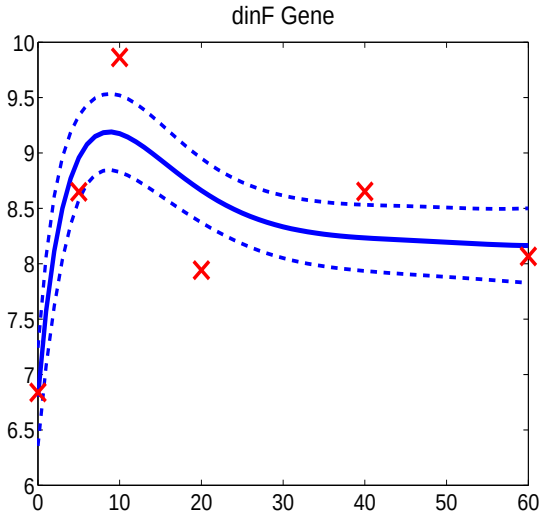
Results on SOS System

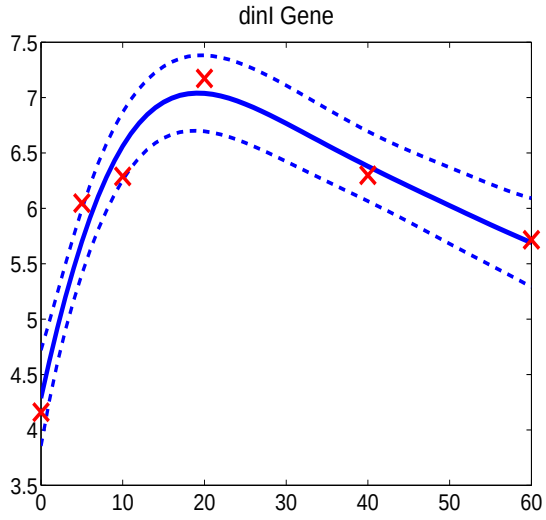
- Again consider the Michaelis-Menten kinetic equation

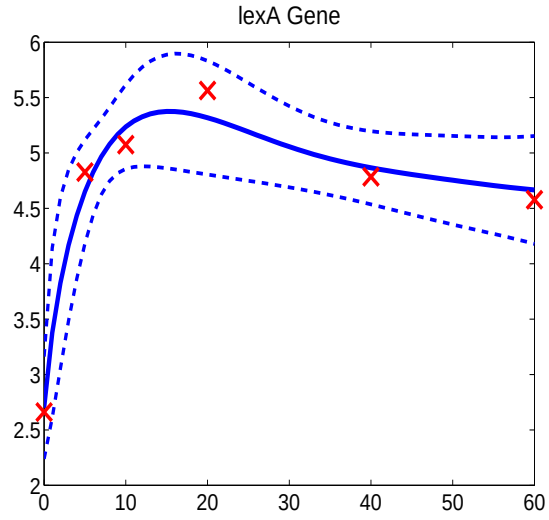
$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = b_j + s_j \frac{1}{\exp(p(t)) + \gamma_j} - d_j m_j(t)$$

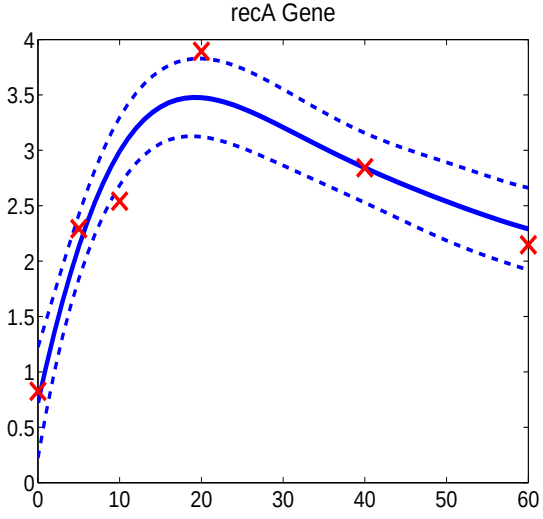
- We have 14 genes (5 kinetic parameters each)
- Gene expressions are available for $T = 6$ time slots
- TF ($\mathbf{p}$) is discretized using 121 points
- MCMC details:
  - 6 control points are used (placed in a equally spaced grid)
  - Running time was 5 hours for 2 million sampling iterations plus burn in
  - Acceptance rate for $\mathbf{p}$ after burn in was between 15% – 25%

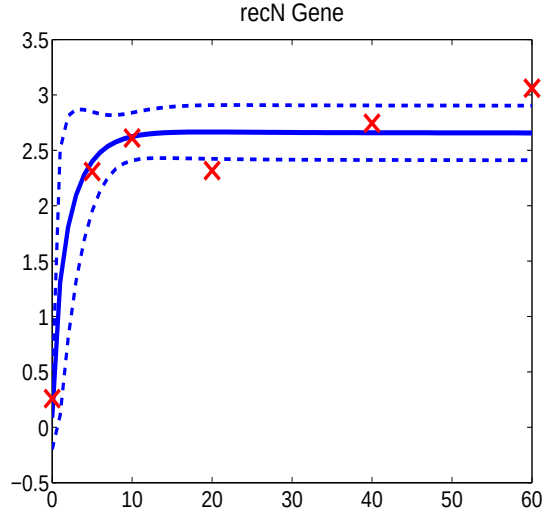
Results in E.coli data: Predicted gene expressions

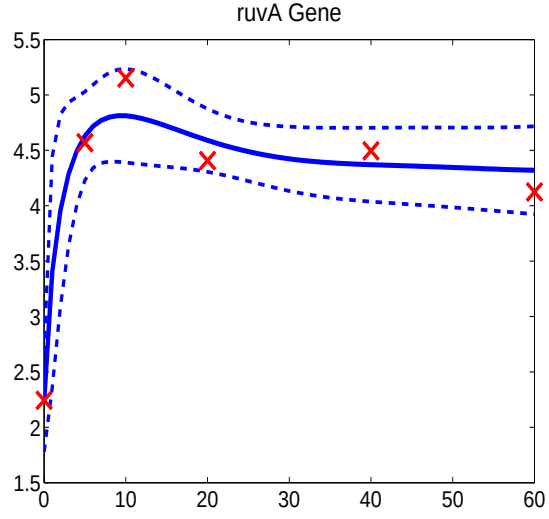




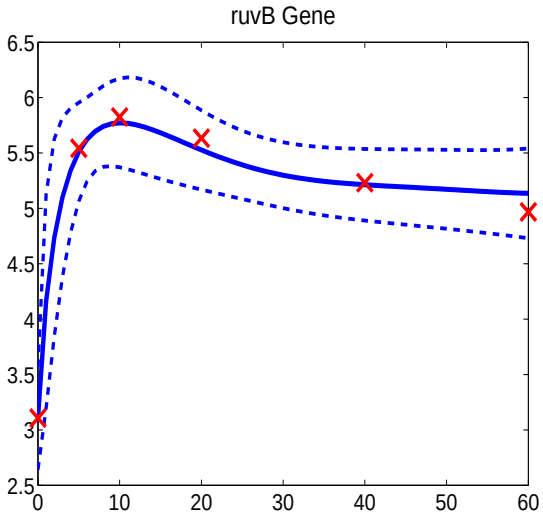


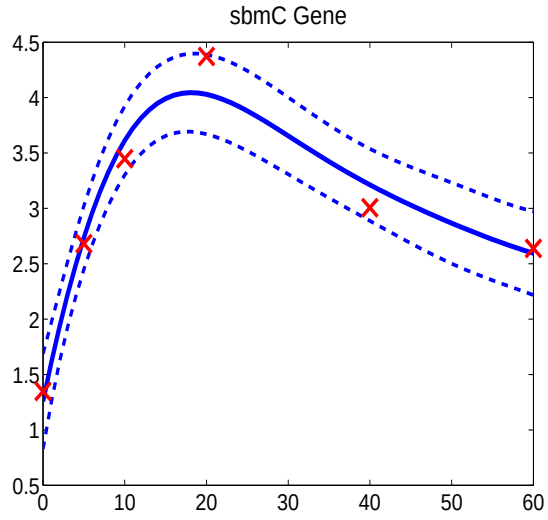


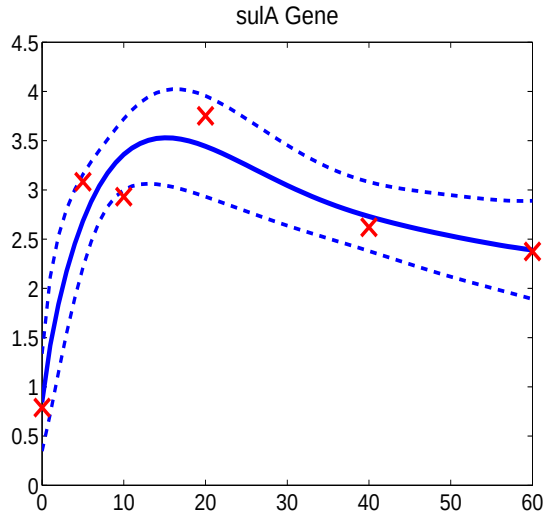


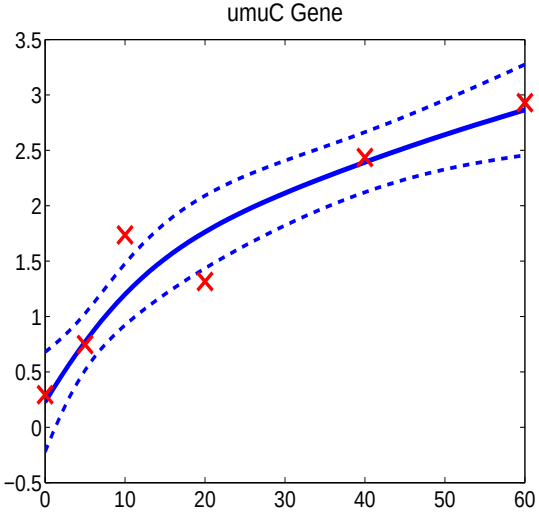


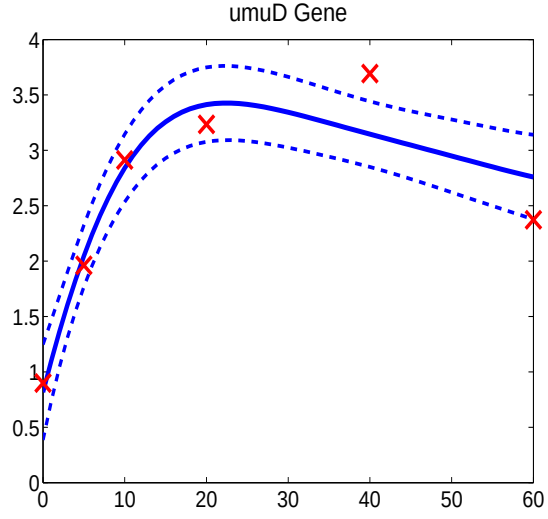
Results in E.coli data: Predicted gene expressions

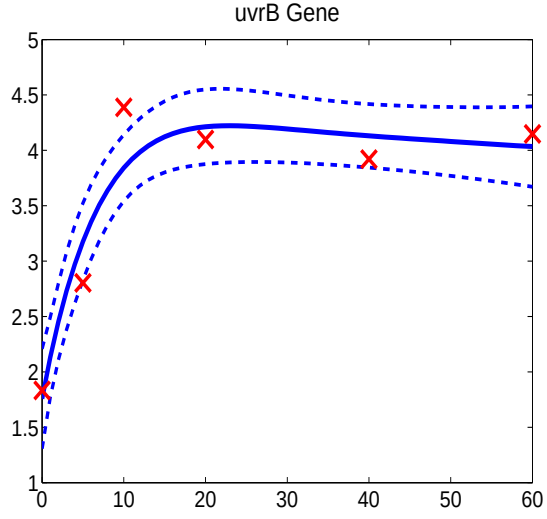




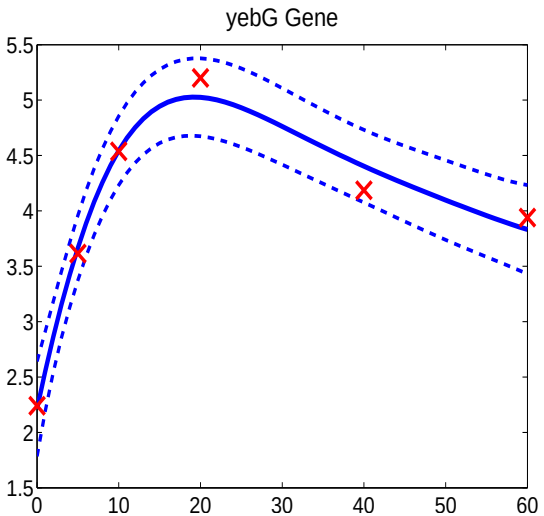


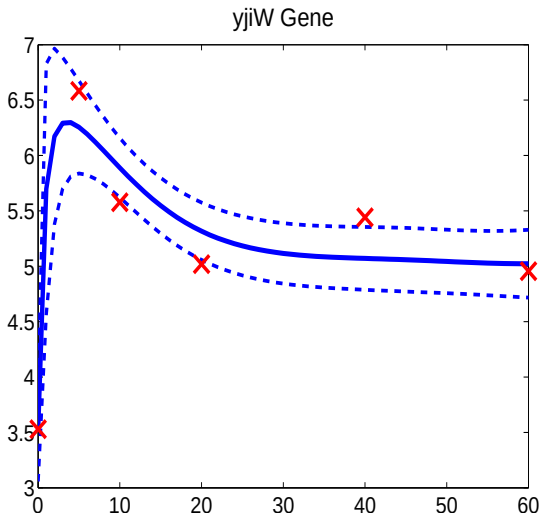




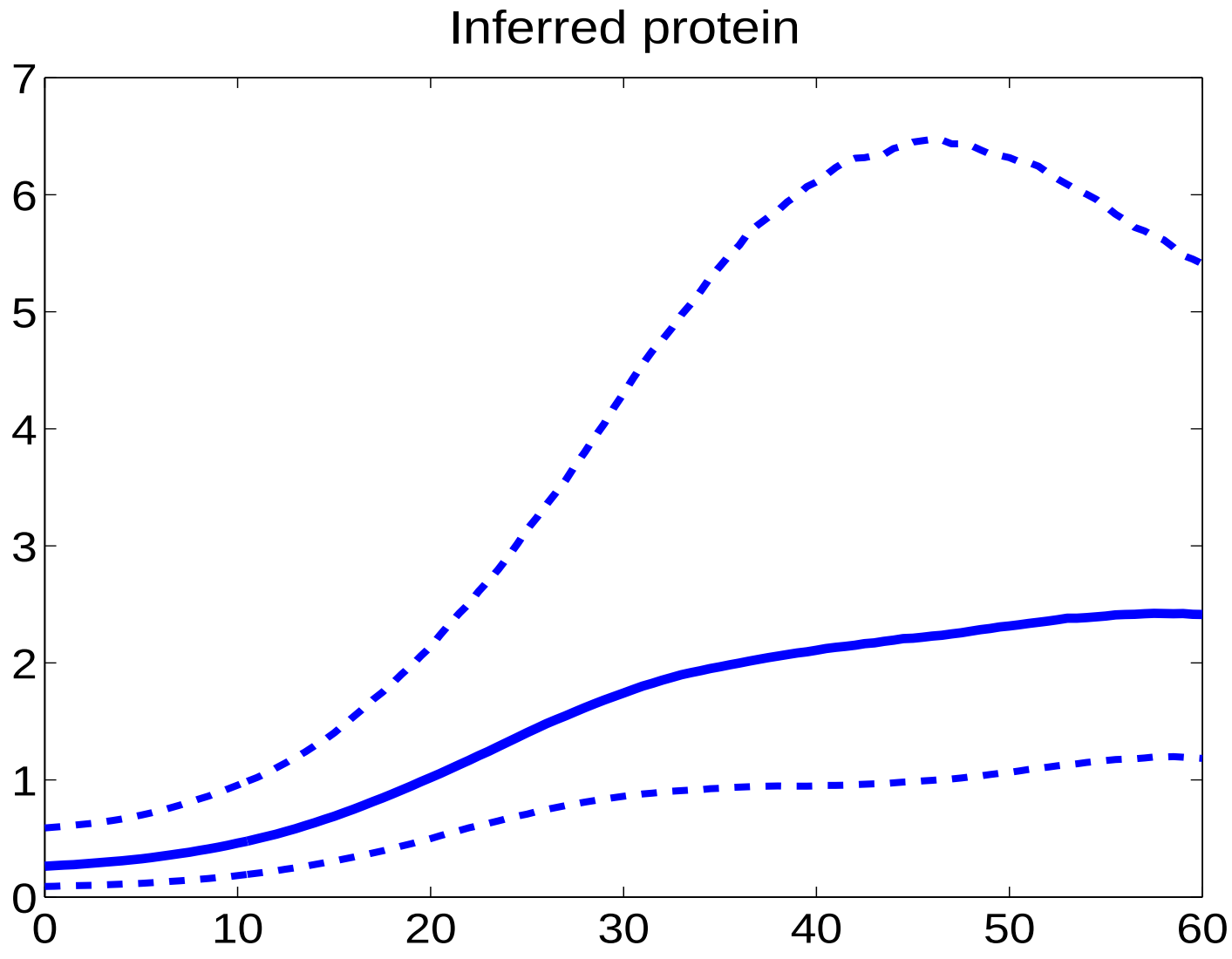


Results in E.coli data: Predicted gene expressions

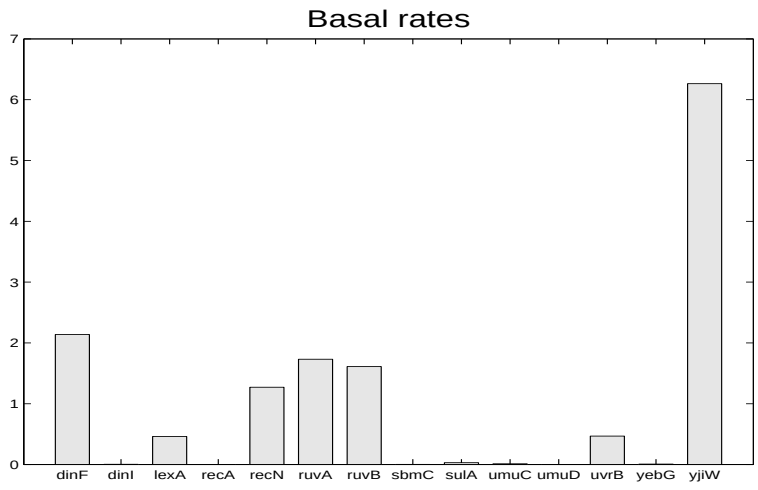


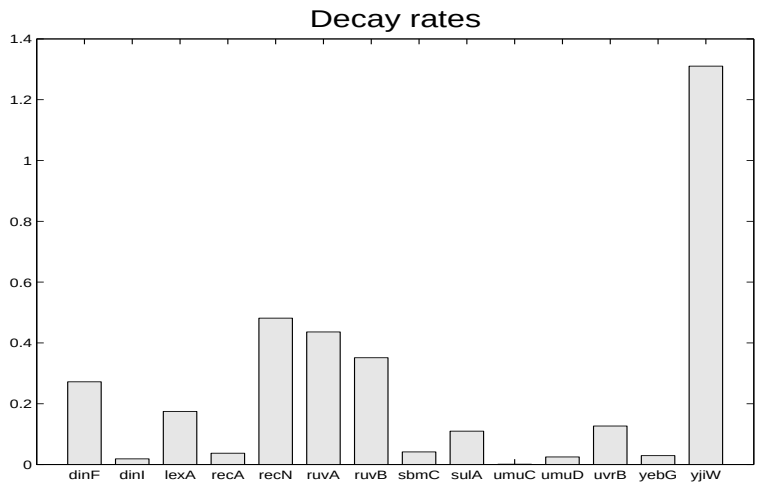


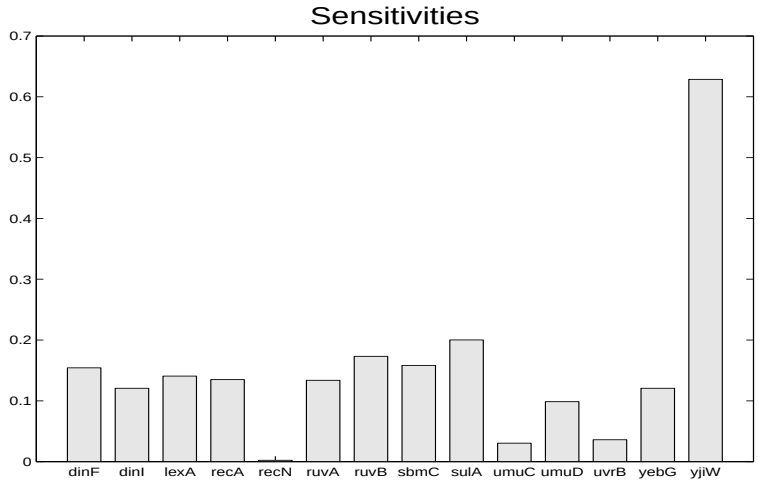
Results in E.coli data: Protein concentration

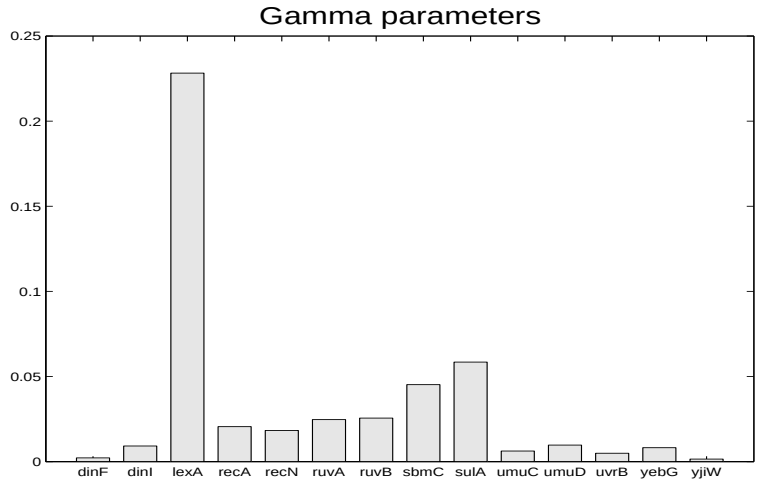


Results in E.coli data: Kinetic parameters

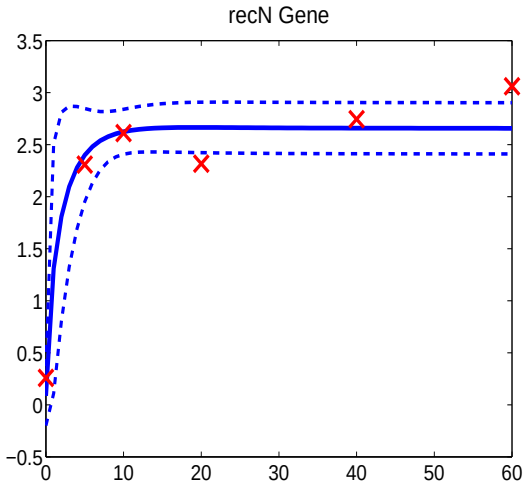


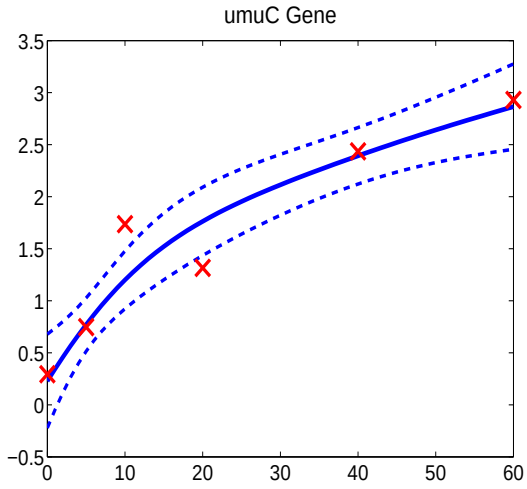


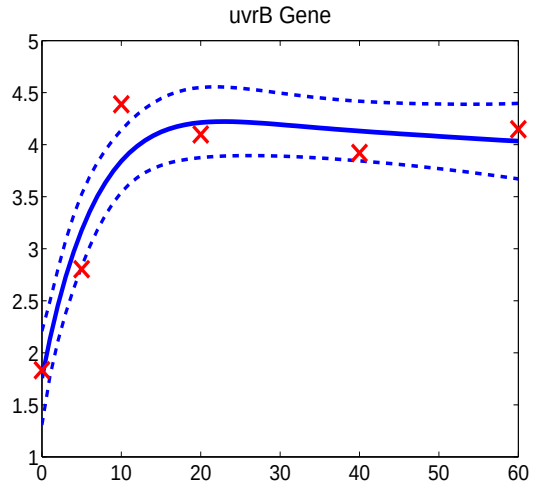


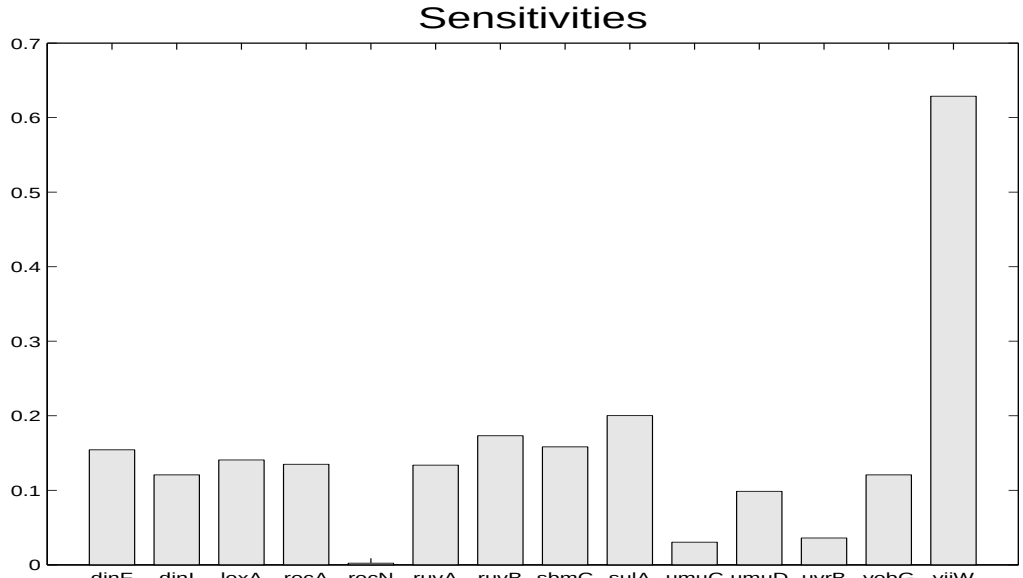


Results in E.coli data: Genes with low sensitivity value

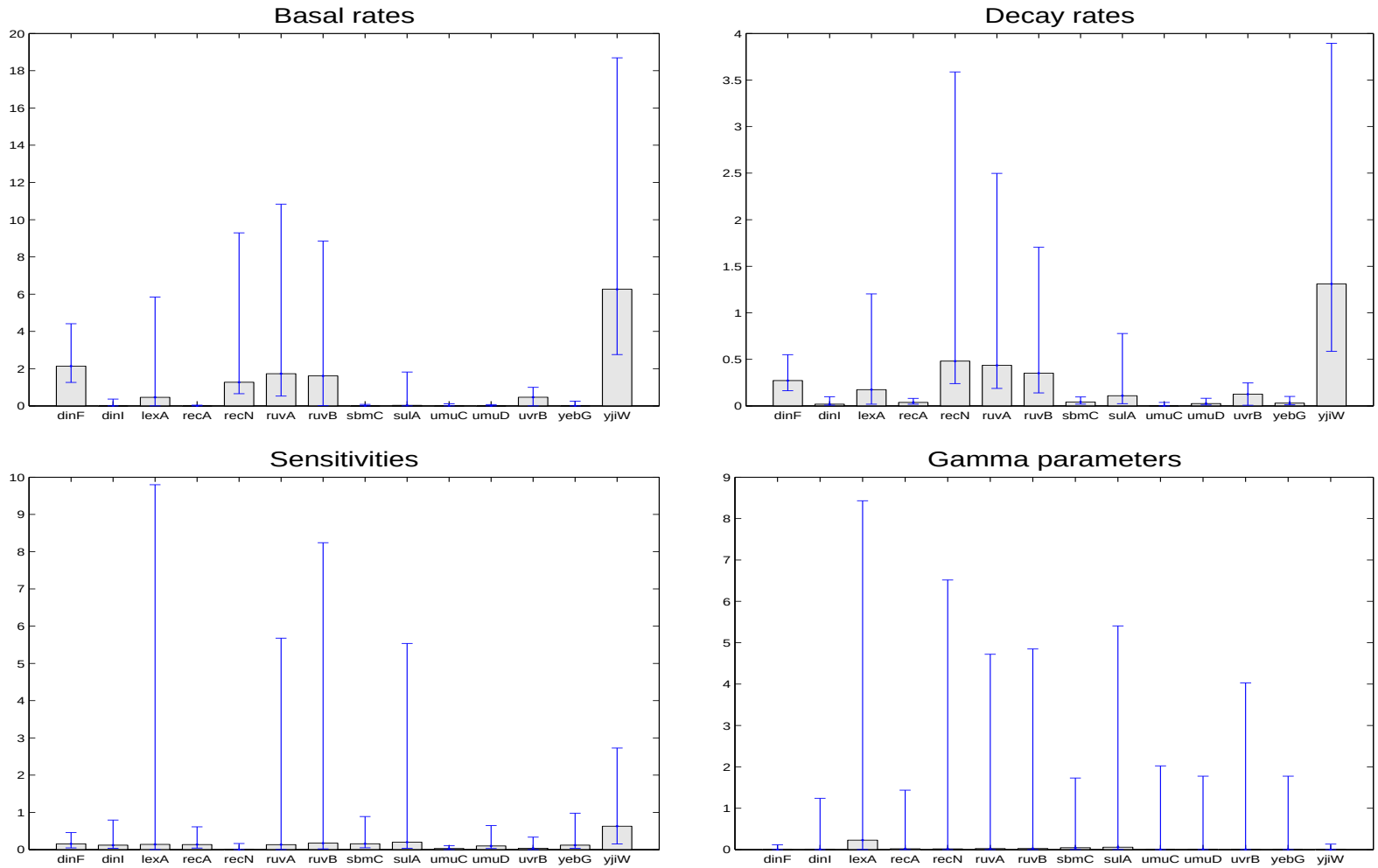








Results in E.coli data: Confidence intervals for the kinetic parameters



BMC Systems Biology

BioMed Central
The Open Access Publisher

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

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

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

ISSN 1752-0509

Article type Methodology article

A "middle-out" approach for inferring regulatory networks

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

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

A "middle-out" approach for inferring regulatory networks

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

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

A "middle-out" approach for inferring regulatory networks

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

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

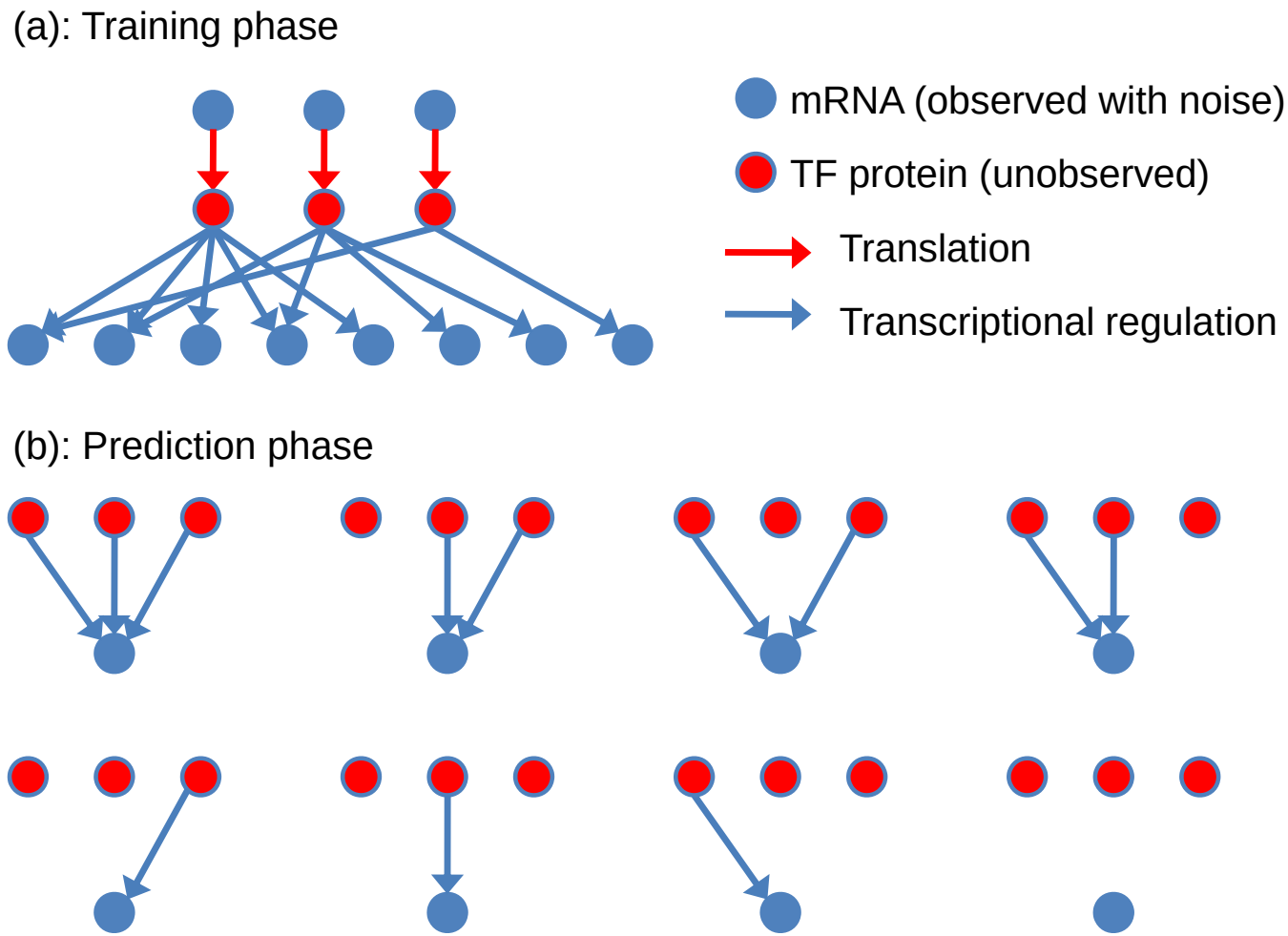
A "middle-out" approach for inferring regulatory networks

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

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

A “middle-out” approach for inferring regulatory networks

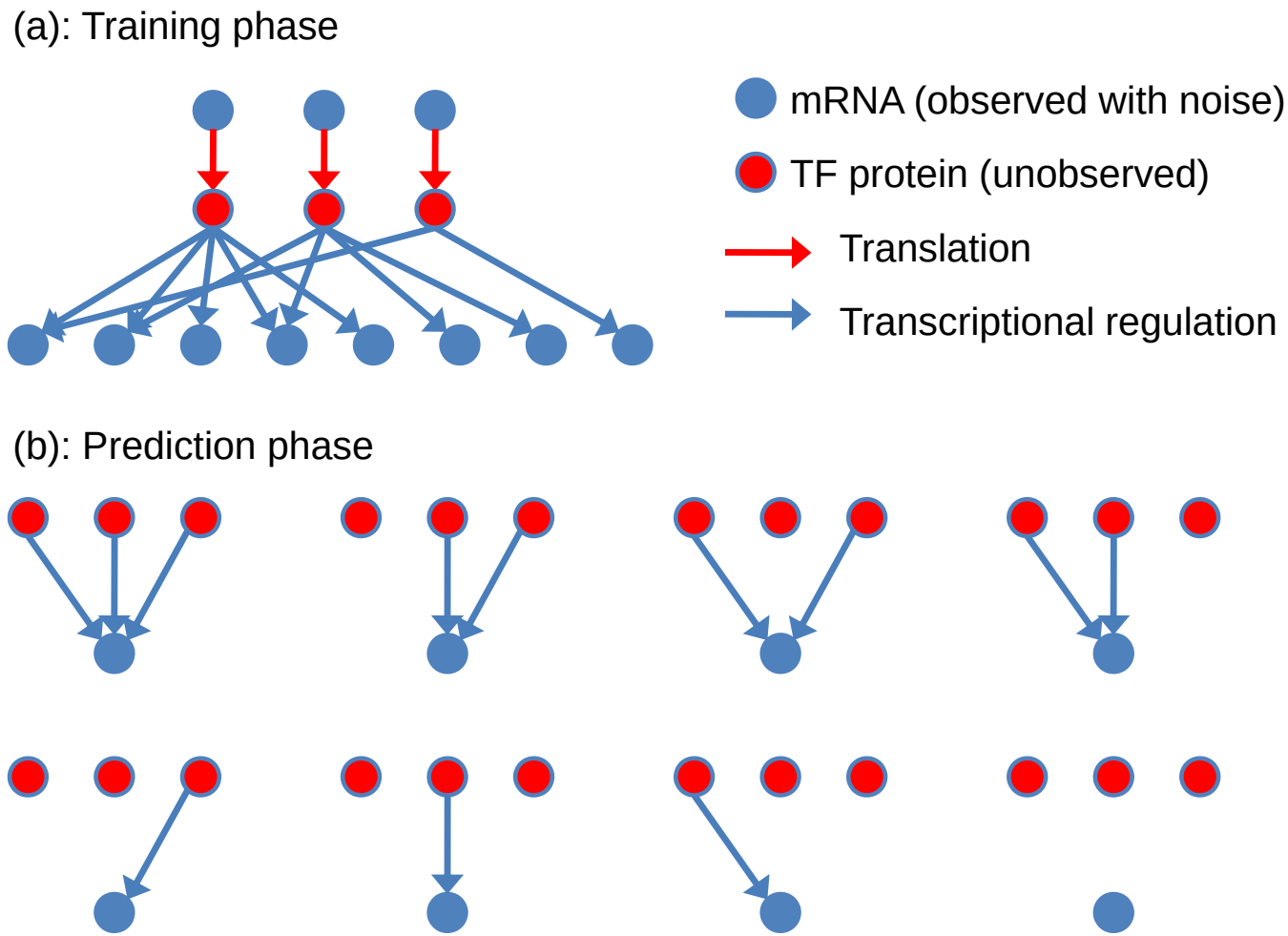
- Training stage: Parameter estimation on known network



- Scanning stage: Bayesian evidence model scoring for

A “middle-out” approach for inferring regulatory networks

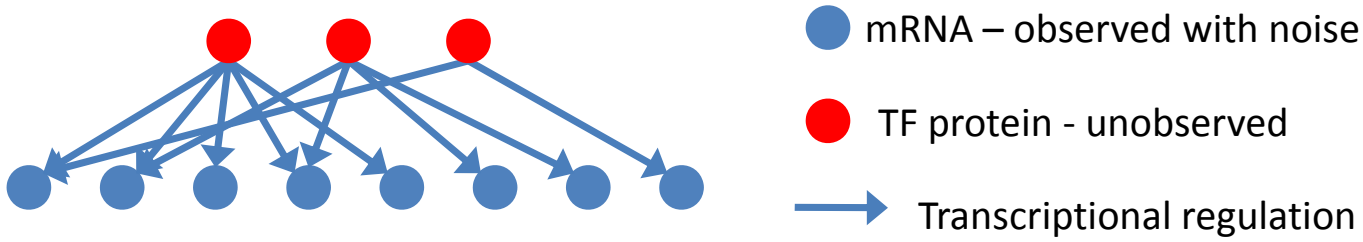
- Training stage: Parameter estimation on known network



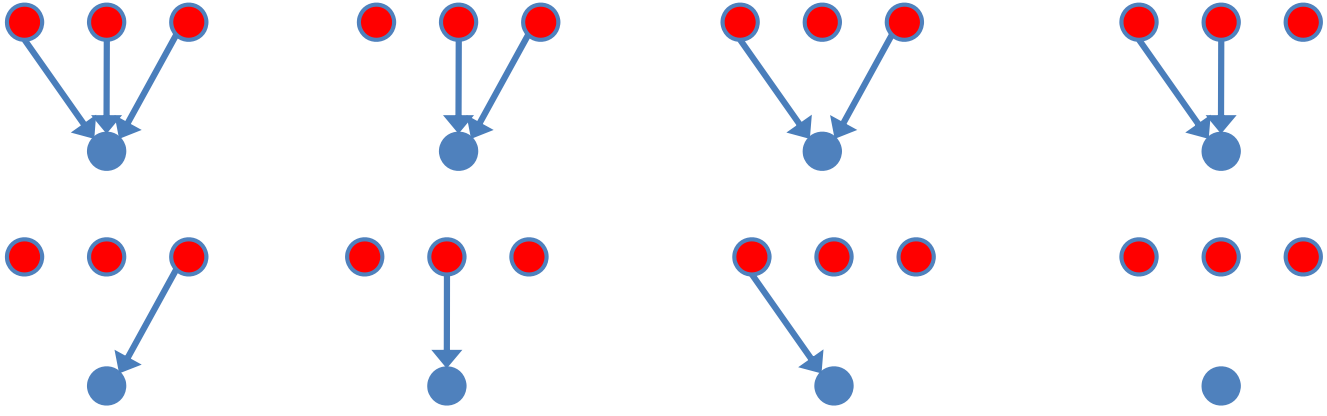
- Scanning stage: Bayesian evidence model scoring for

A “middle-out” approach for inferring regulatory networks

- Training stage with post-translational modification



- Scanning stage: Bayesian evidence model scoring for target inference



Model of transcriptional regulation

- Transcription

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

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

- Translation (optional)

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

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

Model of transcriptional regulation

- Transcription

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

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

- Translation (optional)

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

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

Gaussian process inference over latent functions

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

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010); Titsias et al. (2012)

Gaussian process inference over latent functions

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

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010); Titsias et al. (2012)

Gaussian process inference over latent functions

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

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010); Titsias et al. (2012)

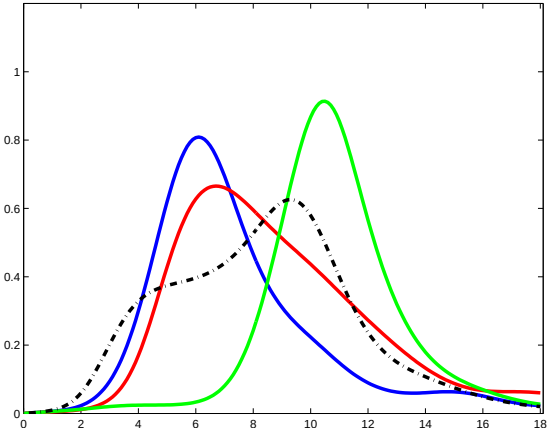
Gaussian process inference over latent functions

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

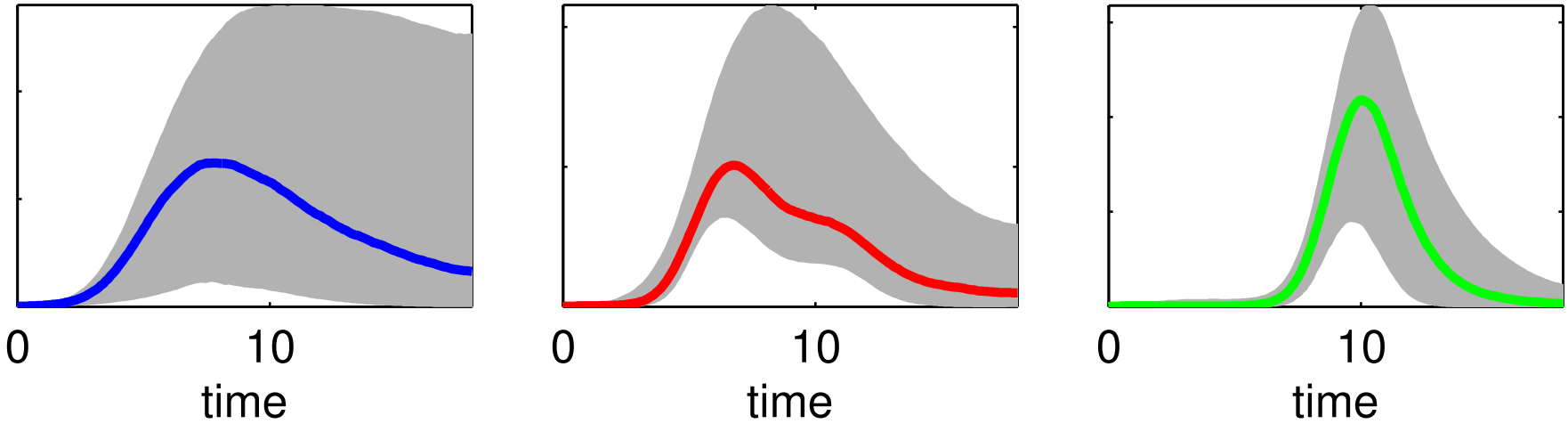
Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010); Titsias et al. (2012)

Artificial data: one experimental condition

Ground Truth TFs

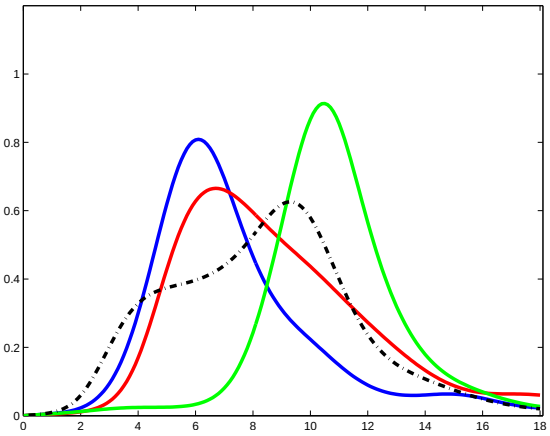


Inferred TF concentrations after training stage

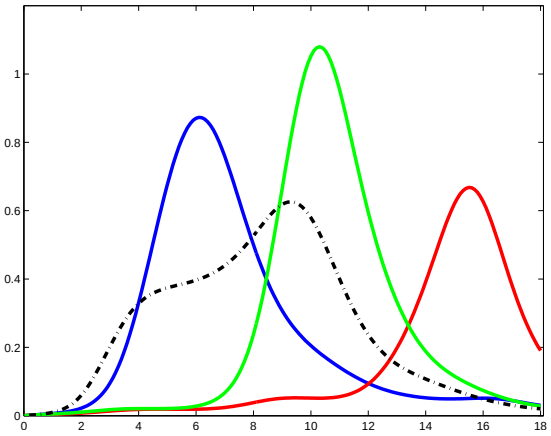


Artificial data: two experimental conditions

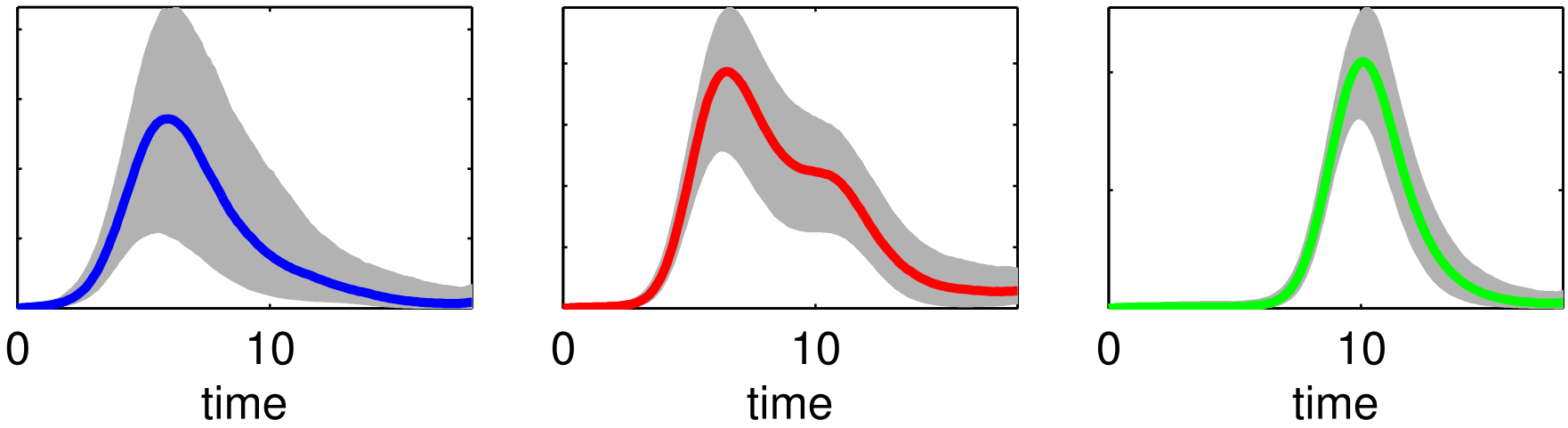
True TFs condition 1



True TFs condition 2

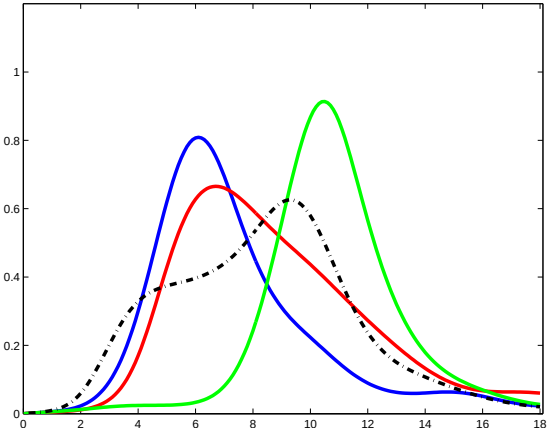


Inferred TF concentrations for condition 1

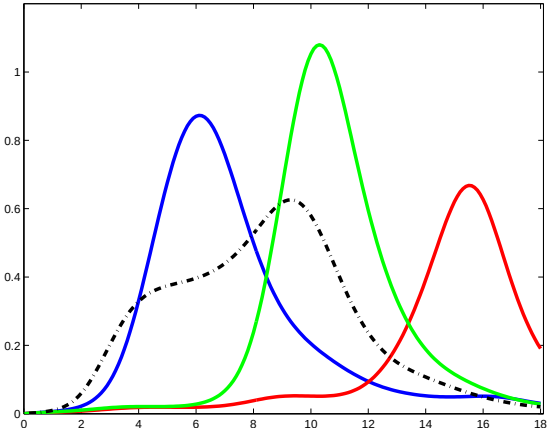


Artificial data: two experimental conditions

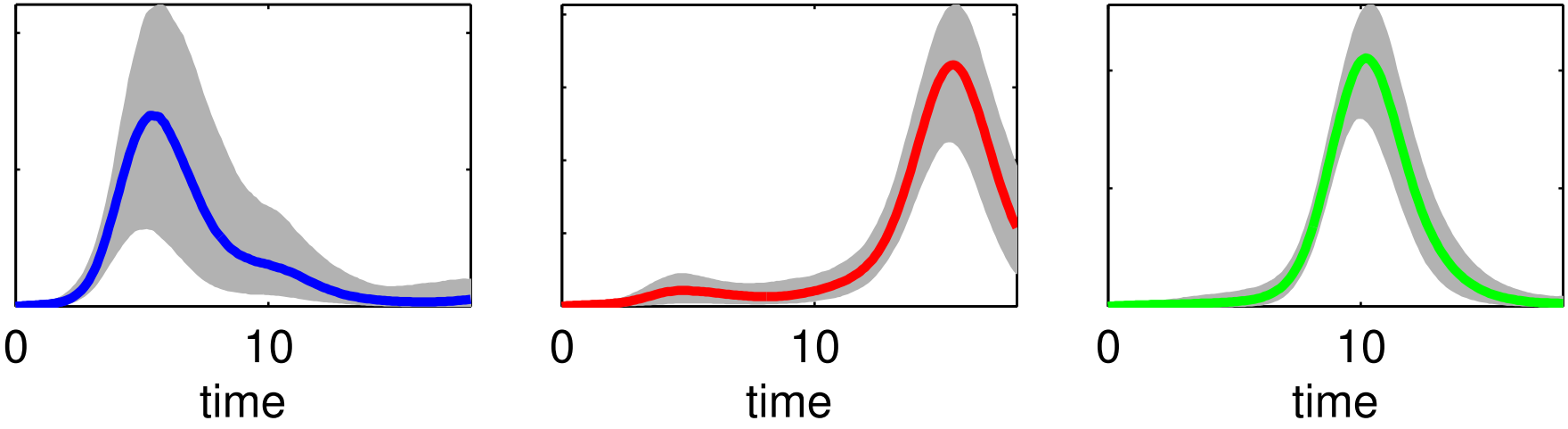
True TFs condition 1



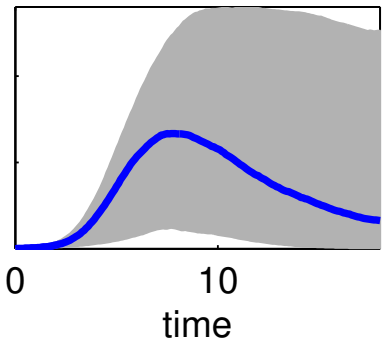
True TFs condition 2

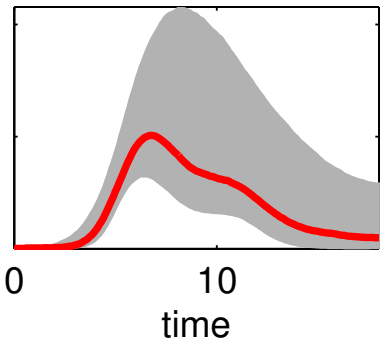


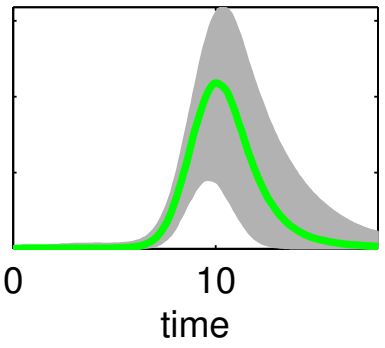
Inferred TF concentrations for condition 2

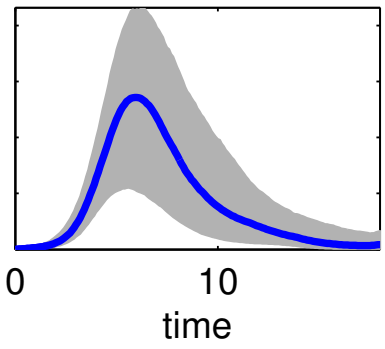


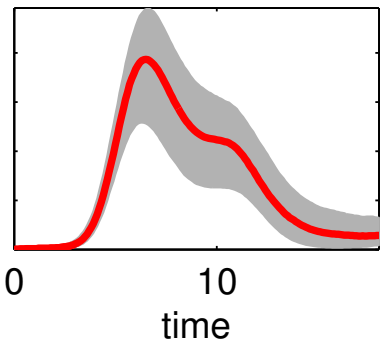
Artificial data: scanning performance for each TF

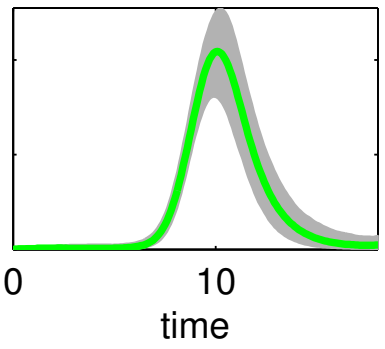


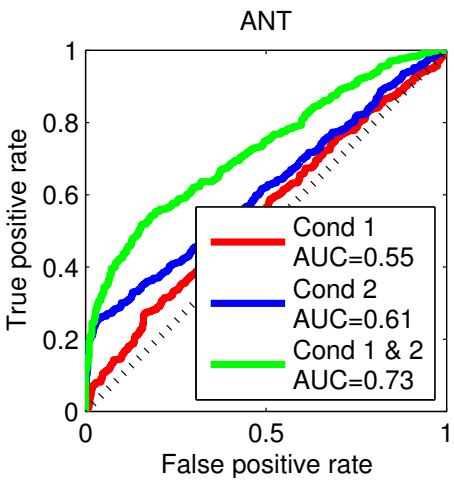


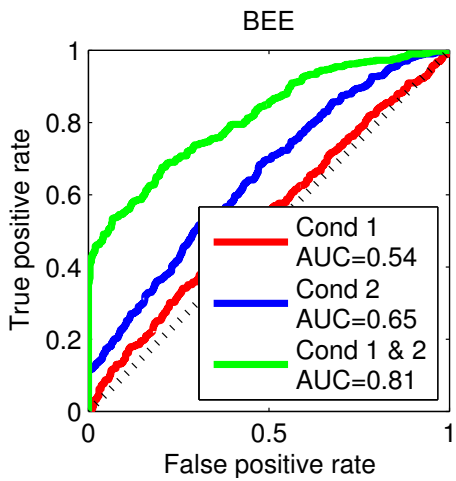


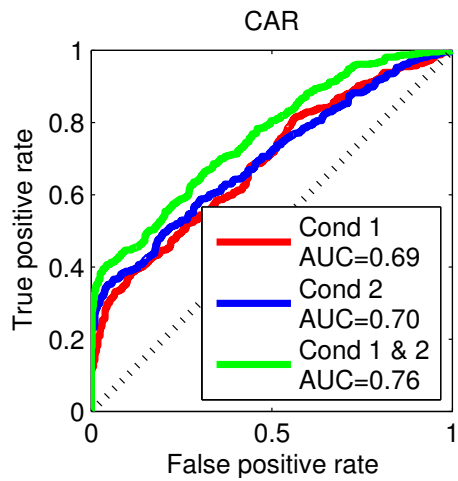




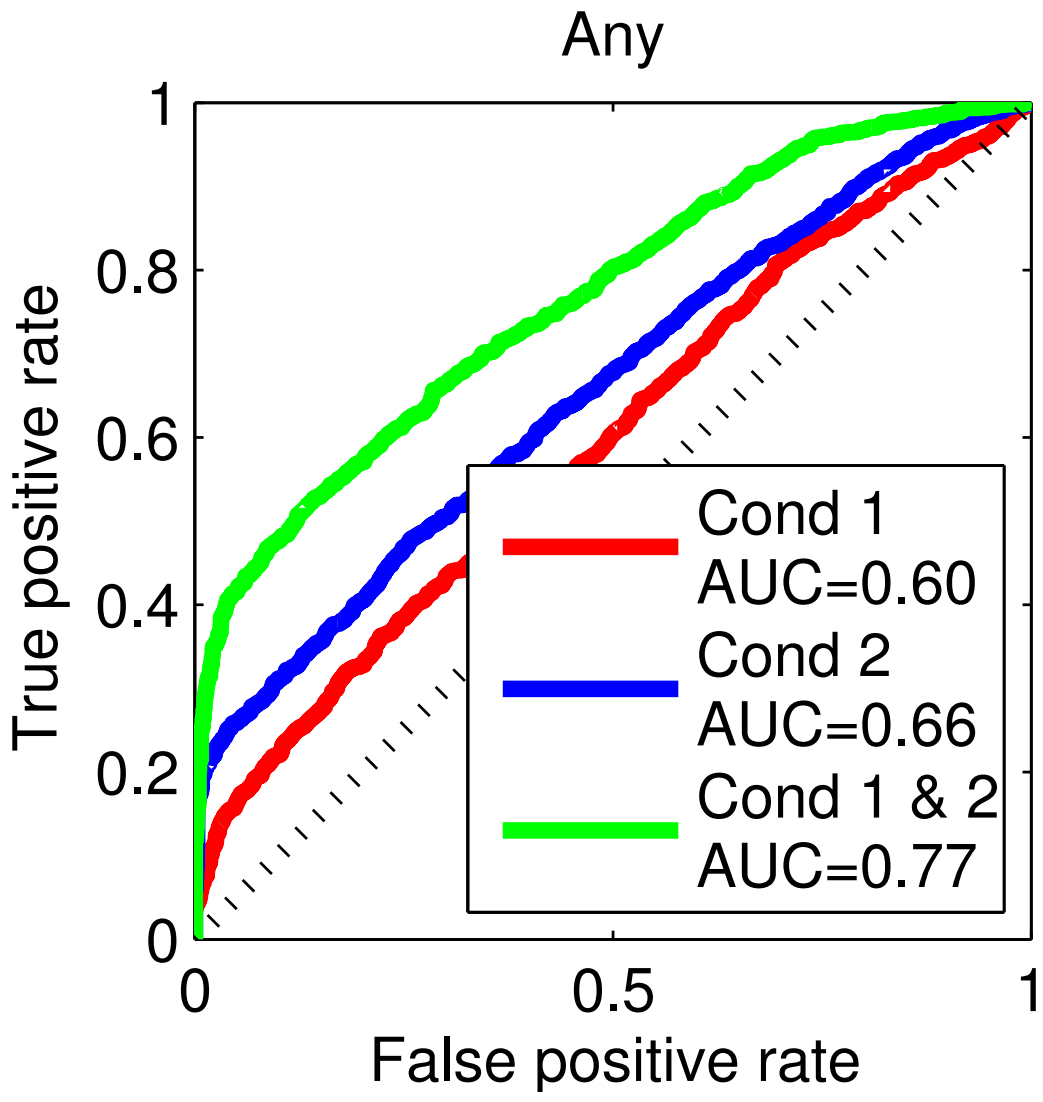






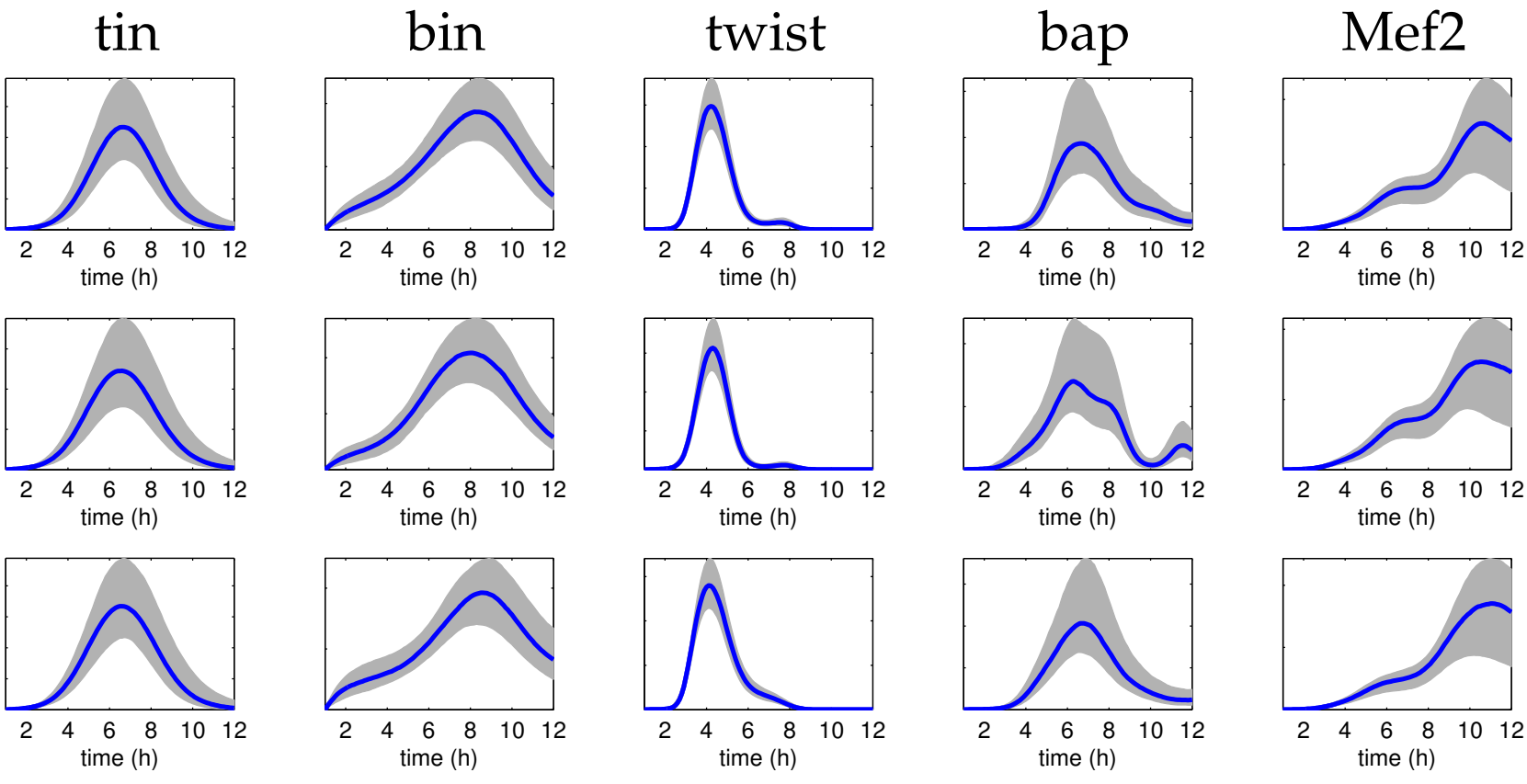


Artificial data: scanning performance for all TFs



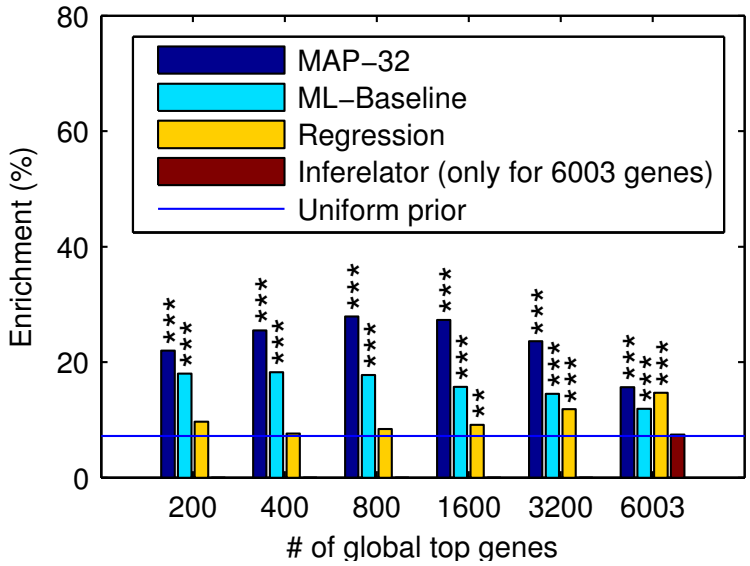
Drosophila training

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

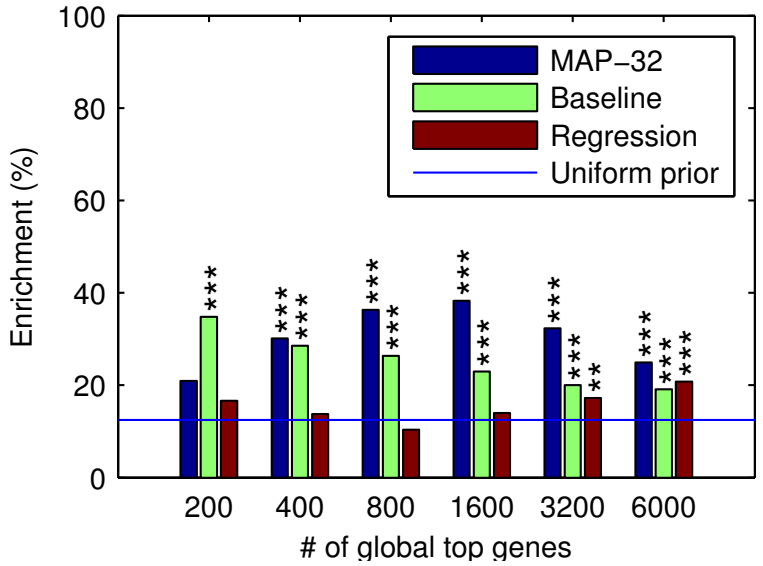


Drosophila scanning: model ranking

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

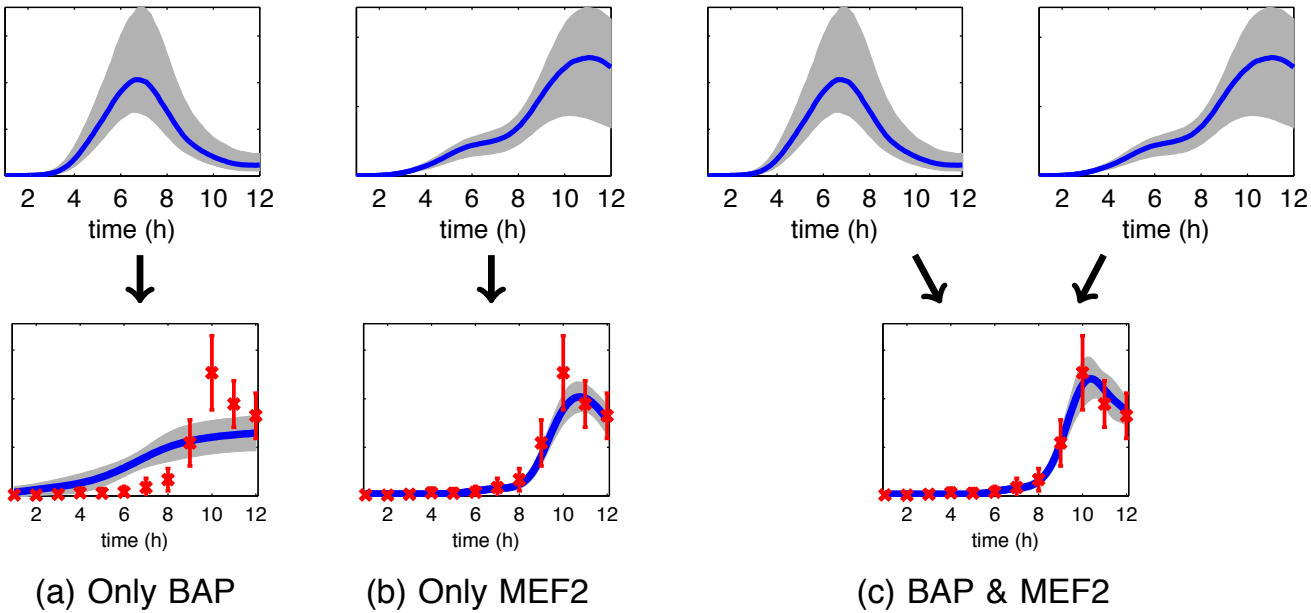


All "non-quiet" genes



All targets with in situ evidence

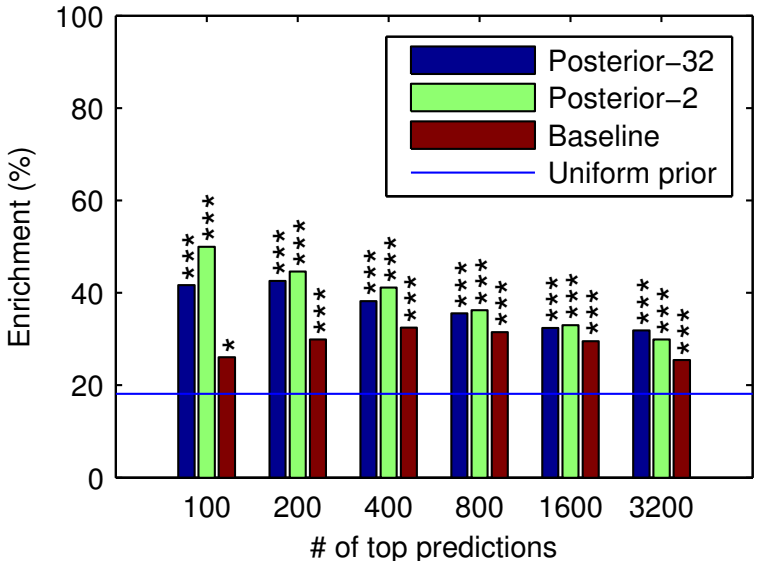
Coregulated Target Example



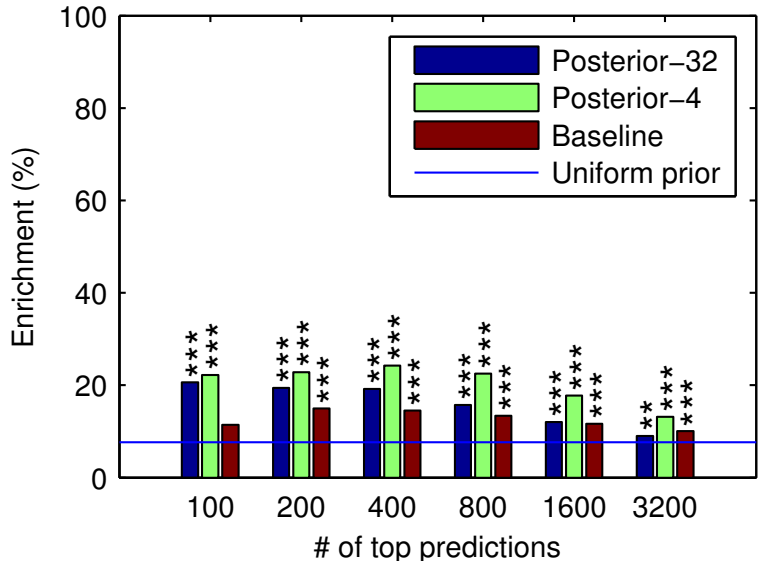
A highly ranked putative joint target of BAP amd MEF2. The candidate gene is confirmed as a joint target by independent ChIP-chip studies Zinzen et al. (2009).

Drosophila scanning: link ranking

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



TF regulation



TF pair regulation

Summary and Conclusion

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

Summary and Conclusion

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

Summary and Conclusion

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

References I

M. A. Álvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In Koller et al. (2009), pages 57–64. [PDF].

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

M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *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 Teh and Titterington (2010), pages 25–32. [PDF].

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.

W. V. Baxter and K.-I. Anjyo. Latent doodle space. In *EUROGRAPHICS*, volume 25, pages 477–485, Vienna, Austria, September 4-8 2006.

References II

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998. [DOI].

E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, Cambridge, MA, 2008. MIT Press.

P. Boyle and M. Frean. Dependent Gaussian processes. In L. Saul, Y. Weiss, and L. Bouttou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 217–224, Cambridge, MA, 2005. MIT Press.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

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

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140 (3):640–651, 2009. [DOI].

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

References III

L. Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

A. Damianou, C. H. Ek, M. K. Titsias, and N. D. Lawrence. Manifold relevance determination. In J. Langford and J. Pineau, editors, *Proceedings of the International Conference in Machine Learning*, volume 29, San Francisco, CA, 2012. Morgan Kauffman. [PDF].

A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In P. Bartlett, F. Peirrera, C. Williams, and J. Lafferty, editors, *Advances in Neural Information Processing Systems*, volume 24, Cambridge, MA, 2011. MIT Press. [PDF].

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].

C. H. Ek, J. Rihan, P. Torr, G. Rogez, and N. D. Lawrence. Ambiguity modeling in latent spaces. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction (MLMI 2008)*, LNCS, pages 62–73. Springer-Verlag, 28–30 June 2008a. [PDF].

References IV

C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In A. Popescu-Belis, S. Renals, and H. Bourlard, editors, *Machine Learning for Multimodal Interaction (MLMI 2007)*, volume 4892 of *LNCS*, pages 132–143, Brno, Czech Republic, 2008b. Springer-Verlag. [PDF].

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, Cambridge, MA, 2014.

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].

Z. Ghahramani, editor. *Proceedings of the International Conference in Machine Learning*, volume 24, 2007. Omnipress. [Google Books] .

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] .

References V

K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *ACM Transactions on Graphics (SIGGRAPH 2004)*, pages 522–531, 2004.

J. D. Helterbrand and N. A. C. Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26(2):205–226, 1994.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In A. Nicholson and P. Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013. [PDF].

J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the conjugate exponential family. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, Cambridge, MA, 2012. [PDF].

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.

D. M. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

References VI

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].

A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978. [Google Books] .

A. A. Kalaitzis and N. D. Lawrence. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12(180), 2011. [DOI].

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

N. J. King and N. D. Lawrence. Fast variational inference for Gaussian Process models through KL-correction. In *ECML, Berlin, 2006*, Lecture Notes in Computer Science, pages 270–281, Berlin, 2006. Springer-Verlag. [PDF].

A. Klami and S. Kaski. Local dependent components analysis. In Ghahramani (2007). [Google Books] .

A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.

References VII

D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors. *Advances in Neural Information Processing Systems*, volume 21, Cambridge, MA, 2009. MIT Press.

J. B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–28, 1964. [DOI].

N. D. Lawrence. Gaussian process models for visualisation of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 329–336, Cambridge, MA, 2004. MIT Press.

N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 11 2005.

N. D. Lawrence. Learning for larger datasets with the Gaussian process latent variable model. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, pages 243–250, San Juan, Puerto Rico, 21-24 March 2007. Omnipress. [PDF].

N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In Ghahramani (2007), pages 481–488. [Google Books] . [PDF].

References VIII

N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In R. Greiner and D. Schuurmans, editors, *Proceedings of the International Conference in Machine Learning*, volume 21, pages 512–519. Omnipress, 2004. [PDF].

N. D. Lawrence, J. C. Platt, and M. I. Jordan. Extensions of the informative vector machine. In J. Winkler, N. D. Lawrence, and M. Niranjan, editors, *Deterministic and Statistical Methods in Machine Learning*, volume 3635 of *Lecture Notes in Artificial Intelligence*, pages 56–87. Springer-Verlag, Berlin, 2005. [Google Books] .

N. D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In W. Cohen and A. Moore, editors, *Proceedings of the International Conference in Machine Learning*, volume 23, pages 513–520. Omnipress, 2006. [Google Books] . [PDF].

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 625–632, Cambridge, MA, 2003. MIT Press.

References IX

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

G. Leen and C. Fyfe. A Gaussian process latent variable model formulation of canonical correlation analysis. Bruges (Belgium), 26-28 April 2006 2006.

T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press.

S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (SIGGRAPH 2012)*, 31(4), 2012.

D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction with radial basis function networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 543–549, Cambridge, MA, 1997. MIT Press.

C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with GaussianFace. Technical report,

D. J. C. MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research, A*, 354(1):73–80, 1995. [DOI].

References X

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, U.K., 2003. [Google Books] .

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. [Google Books] .

T. P. Minka and R. W. Picard. Learning how to learn is learning with point sets. Available on-line., 1997. [URL]. Revised 1999, available at `http://www.stat.cmu.edu/~{}minka/`.

R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society Press, 2007.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 2002.

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.

References XI

A. D. Polyanin. *Handbook of Linear Partial Differential Equations for Engineers and Scientists*. Chapman & Hall/CRC, 1 edition, 2002.

V. Priacuriu and I. D. Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011a.

V. Priacuriu and I. D. Reid. Shared shape spaces. In *IEEE International Conference on Computer Vision (ICCV)*, 2011b.

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.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

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.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. [DOI].

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969. [DOI].

References XII

B. Schölkopf, C. J. C. Burges, and V. N. Vapnik. Incorporating invariances in support vector learning machines. In *Artificial Neural Networks — ICANN'96*, volume 1112, pages 47–52, 1996.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. [DOI].

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2001.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, The University of Edinburgh, 2004.

M. Seeger and M. I. Jordan. Sparse Gaussian Process Classification With Multiple Classes. Technical Report 661, Department of Statistics, University of California at Berkeley,

M. Seeger, C. K. I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, 3–6 Jan 2003.

A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning shared latent structure for image synthesis and robotic imitation. In Weiss et al. (2006).

References XIII

G. Skolidis and G. Sanguinetti. Bayesian multitask classification with Gaussian process priors. *IEEE Transactions on Neural Networks*, 22(12):2011 – 2021, 2011.

A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In Leen et al. (2001), pages 619–625.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Weiss et al. (2006).

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, 1999. [Google Books] .

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.

Y. W. Teh and D. M. Titterington, editors. *Artificial Intelligence and Statistics*, volume 9, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000. [DOI].

References XIV

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].

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

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

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Teh and Titterington (2010), pages 844–851. [PDF].

M. K. Titsias, N. D. Lawrence, and M. Rattray. Efficient sampling for Gaussian process inference using control variables. In Koller et al. (2009), pages 1681–1688. [PDF].

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

L. R. Tucker. An inter-battery method of factor analysis. *Psychometrika*, 23(2): 111–136, 1958.

R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable model for classification. In Ghahramani (2007). [Google Books] .

References XV

R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, New York, U.S.A., 17–22 Jun. 2006. IEEE Computer Society Press.

R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *IEEE International Conference on Computer Vision (ICCV)*, pages 403–410, Bejing, China, 17–21 Oct. 2005. IEEE Computer Society Press.

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

S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In L. Getoor and T. Scheffer, editors, *Proceedings of the International Conference in Machine Learning*, volume 28, 2011.

H. Wackernagel. *Multivariate Geostatistics: An Introduction With Applications*. Springer-Verlag, 3rd edition, 2003. [Google Books] .

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In Weiss et al. (2006).

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008. ISSN 0162-8828. [DOI].

References XVI

Y. Weiss, B. Schölkopf, and J. C. Platt, editors. *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

C. K. Williams and D. Barber. Bayesian Classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (12):1342–1351, 1998.

C. K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In Leen et al. (2001), pages 682–688.

I. Woodward, M. R. Lomas, and R. A. Betts. Vegetation-climate feedbacks in a greenhouse world. *Philosophical Transactions: Biological Sciences*, 353(1365): 29–39, 1998.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 1012–1019, 2005.

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].