Bayesian Approaches to Transcription Factor Target Identification

Neil D. Lawrence and Antti Honkela

Departments of Neuro- and Computer Science, University of Sheffield, U.K.
Course in Systems Biology 9th-11th October, Bologna, Italy

10th October 2010

Outline

Outline

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Can a Biologist Fix a Radio? Lazebnik (2002)

The Case for Systems Biology

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

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

Coregulation of Gene Expression

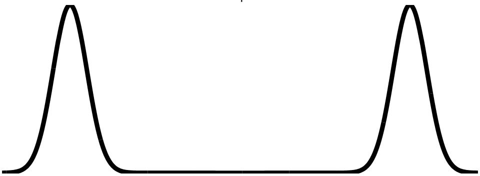
The Case for Computational Biology

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

General Approach

Broadly Speaking: Two approaches to modeling

data modeling | *mechanistic modeling*

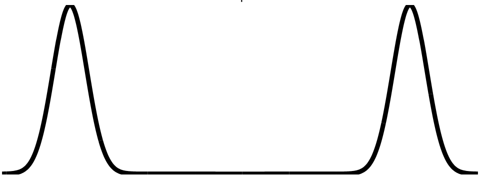


General Approach

Broadly Speaking: Two approaches to modeling

data modeling | *mechanistic modeling*

let the data "speak"



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

mechanistic modeling

impose physical laws

General Approach

Broadly Speaking: Two approaches to modeling

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

General Approach

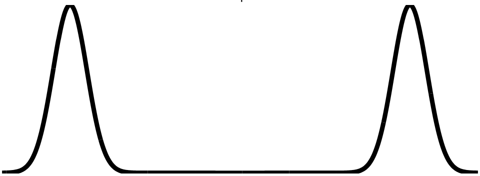
Broadly Speaking: Two approaches to modeling

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

General Approach

Broadly Speaking: Two approaches to modeling

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



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

computational models

adaptive models

mechanistic modeling

impose physical laws

systems models

differential equations

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

computational models

adaptive models

PCA, clustering

mechanistic modeling

impose physical laws

systems models

differential equations

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

computational models

adaptive models

PCA, clustering

mechanistic modeling

impose physical laws

systems models

differential equations

SDE, ODE models

A Hybrid Approach

Introduce aspects of systems biology to computational models

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

Radiation Damage in the Cell

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

p53 “Guardian of the Cell”

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

p53 DNA Damage Repair

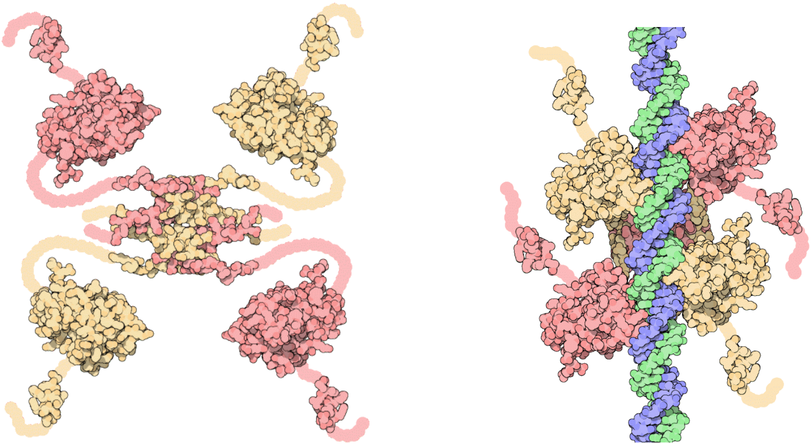


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the"Molecule of the Month" feature).

p53

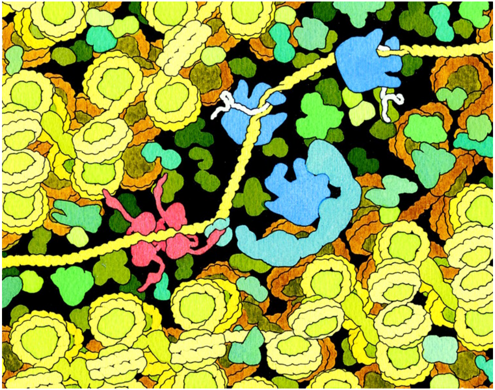


Figure: Repair of DNA damage by p53. Image from Goodsell (1999).

Some p53 Targets

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

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

hPA26/SESN1 sestrin 1 Cell Cycle arrest.

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

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

Modelling Assumption

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

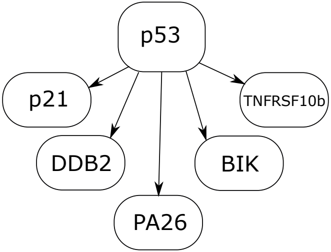


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

Standard Approach

Clustering of Gene Expression Profiles

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

Outline

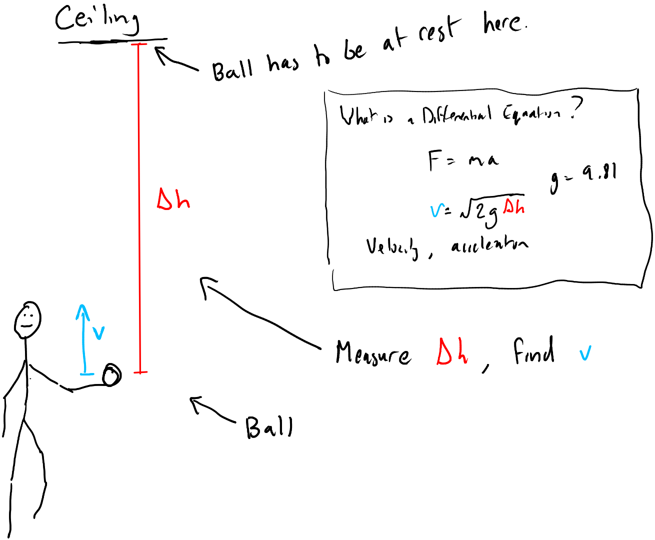
Differential Equation Overview

- What is a differential equation?
  - A way of relating quantities that are rates of each other:
    - x, is position.
    - v, velocity is rate of change of position.
    - a, acceleration is rate of change of velocity.
  - Given Newton's laws (a mechanistic model) we can use differential equations to compute, for example that:

    $$v(0) = \sqrt{2g\Delta x}$$

    where we neglect air resistance.
  - Where Δx is the height I throw a ball. $g = 9.81$ is the acceleration of the earth due to gravity. $v(0)$ is it's *initial velocity*.

Experiment


Ceiling
Ball has to be at rest here.
What is a Differential Equation?
F = ma
g = 9.81
v = √2g Δh
Velocity, acceleration
Δh
v
Measure Δh, find v
Ball

Differential Equations

- Velocity, Acceleration and Position

$$v = \frac{\mathrm{d}x}{\mathrm{d}t} \quad a = \frac{\mathrm{d}v}{\mathrm{d}t} = \frac{\mathrm{d}^2x}{\mathrm{d}t^2}$$

- All of these are functions of time. Our previous experiment found the initial condition, $v(0)$.
- The differential equation gives us the entire trajectory.

Entire Trajectory

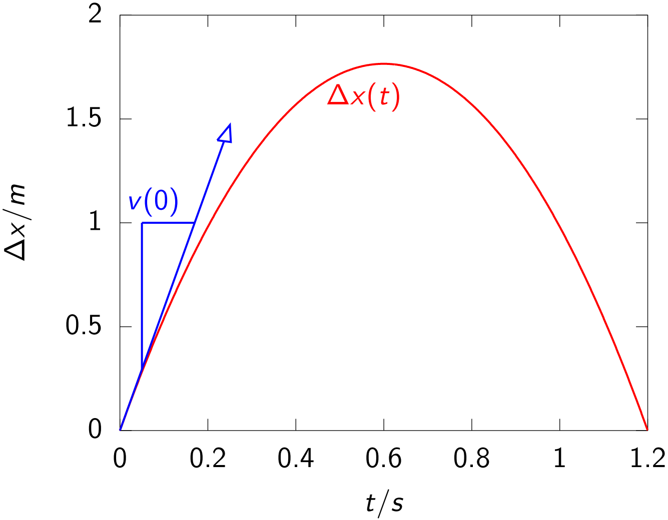


Figure: Theoretical trajectory for the ball given an initial speed of $v(0)$.

Entire Trajectory

Figure: Actual trajectory for a motorcyclist with constant forward motion. Photo by Geraint Warlow. Available under Creative Commons, http://www.flickr.com/photos/gpwarlow/850611221/.

Entire Trajectory

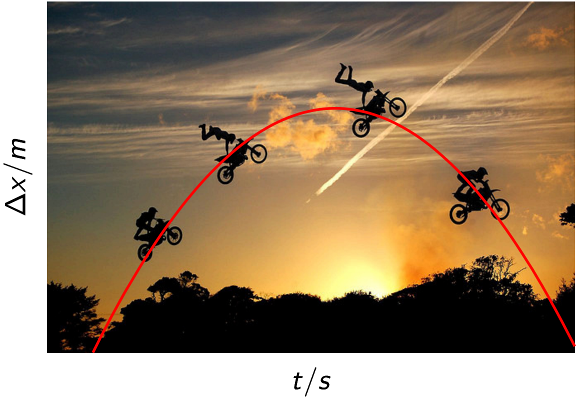


Figure: Actual trajectory for a motorcyclist with constant forward motion. Photo by Geraint Warlow. Available under Creative Commons, http://www.flickr.com/photos/gpwarlow/850611221/.

Galileo

- Empirically showed objects fell in a parabola.
- Overthrew Aristotlean view of motion.

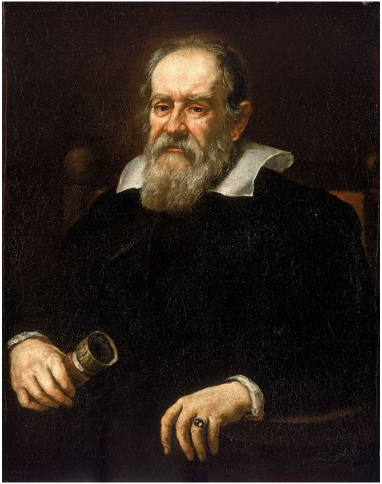


Figure: Galileo Galilei in 1636

Newton

- Developed calculus (alongside Leibniz).
- Laid the foundations of mechanistic modelling with description of gravity.

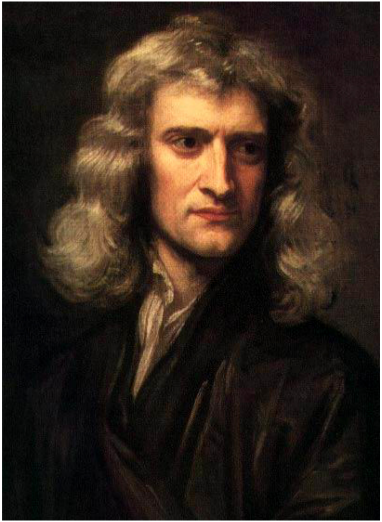


Figure: Isaac Newton in 1689

Biological Systems

- In biological systems:
  1. $x(t)$ is now concentration of gene j, $m_j(t)$.
  2. $v(t)$ is now rate of production of gene j, $\frac{\mathrm{d}m_j(t)}{\mathrm{d}t}$.

Biological Systems

- Instead of masses of planets and force of gravity, we now have:
  1. concentration of governing TF, $p(t)$,
  2. decay rate of an mRNA, d_j,
  3. sensitivity to governing TF, s_j,
  4. base rate of transcription, b_j.

Mathematical Model

- Differential equation model of system.

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

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

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

Outline

Fitting Observations to Data

- Back to example:
$$v(0) = \sqrt{2g\delta x}$$
- Make observation of δx.
- Compute $v(0)$.
- But what if I give you two observations of δx?
- Were there two different values $v(0)$?

Theory of Error

- This was a problem also for celestial mechanics.
- If you have more observations than unknowns, which are the right observations?
- Both Laplace and Gauss worked on this.

Figure: Pierre Simon Laplace 1749–1827

Theory of Error

- This was a problem also for celestial mechanics.
- If you have more observations than unknowns, which are the right observations?
- Both Laplace and Gauss worked on this.

Figure: Carl Friedrich Gauss 1777–1855

Theory of Error: Generative Model of System

- Back to example:

$$v(0) = \sqrt{2g\Delta x}$$

Theory of Error: Generative Model of System

- Back to example:

$$v(0) = \sqrt{2g\Delta x}$$
$$\Delta x = \frac{v(0)^2}{2g}$$

Theory of Error: Generative Model of System

- Back to example:

$$v(0) = \sqrt{2g\Delta x}$$

$$\Delta x = \frac{v(0)^2}{2g}$$

- This rearrangement reflects the fact that height is a result of velocity not the other way around.

Theory of Error: Generative Model of System

- Back to example:

$$v(0) = \sqrt{2g\Delta x}$$
$$\Delta x = \frac{v(0)^2}{2g}$$

- This rearrangement reflects the fact that height is a result of velocity not the other way around.
- Now add errors ...

$$\Delta x_i = \frac{v(0)^2}{2g} + \epsilon_i$$

where ϵ_i represents the error in the ith measurement of height.

- Now we have a single velocity, but need to deal with all these errors.

$$\Delta x_i = \frac{v(0)^2}{2g} + \epsilon_i$$

- Need to introduce a probability distribution for errors.

$$p(x_i | v(0))$$

Noise Model

- Relates observation to actual value.
- Idea: we observe a corrupted version of the truth.
- For Laplace and Gauss a corrupted version of a planets actual position.
- For us a corrupted version of the balls maximum height.
- The object that defines this relationship is a *noise model*.

Gaussian Density

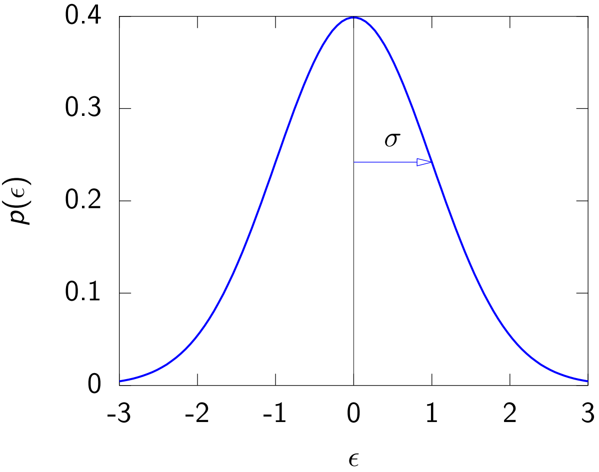


Figure: Gaussian density. $p(\epsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$.

Gaussian Density

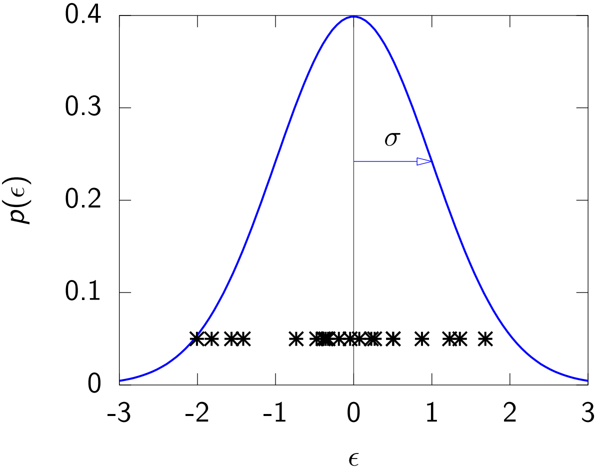


Figure: Gaussian density. $p(\epsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$. 20 Samples from the Density.

The Likelihood

- This model of the error gives us a probabilistic relationship between the velocity and the position.
- Because the position is

$$x_i = \frac{v(0)^2}{2g} + \epsilon_i.$$

- We know that

$$\epsilon_i = x_i - \frac{v(0)^2}{2g}.$$

The Likelihood II

- This allows us to go from

$$p(\epsilon_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\epsilon_i)^2}{2\sigma^2}\right)$$

- To the *likelihood*:

$$p(x_i|v(0)^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x_i - \frac{v(0)^2}{2g}\right)^2}{2\sigma^2}\right)$$

- Which can also be written

$$p(x_i|v(0)^2) = \mathcal{N}\left(x_i|\frac{v(0)^2}{2g}, \sigma^2\right)$$

or

$$x_i|v(0)^2 \sim \mathcal{N}\left(\frac{v(0)^2}{2g}, \sigma^2\right)$$

Maximum Likelihood

- Maximize the probability of the observations:

$$L(v(0)) = \log \prod_{i=1}^{n} p(x_i | v(0)^2)$$

gives

$$2gv(0)^2 = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$v(0) = \sqrt{\frac{1}{2gn} \sum_{i=1}^{n} x_i}$$

Bayesian Updates

- Bayesian approach is slightly different.
- Consider product rule of probability:

$$p(x, v(0)^2) = p(x|v(0)^2)p(v(0)^2)$$
$$p(v(0)^2, x) = p(v(0)^2|x)p(x) = p(x, v(0)^2)$$

- Reorganize to obtain Bayes' rule:

$$p(v(0)^2|x) = \frac{p(x|v(0)^2)p(v(0)^2)}{p(x)}$$

Bayesian Updates

- Bayesian approach is slightly different.
- Consider product rule of probability:

$$
\begin{aligned}
p(x, v(0)^2) &= p(x|v(0)^2)p(v(0)^2) \\
p(v(0)^2, x) &= p(v(0)^2|x)p(x) = p(x, v(0)^2)
\end{aligned}
$$

- Reorganize to obtain Bayes' rule:

$$
\overbrace{p(v(0)^2|x)}^{\text{posterior}} = \frac{\overbrace{p(x|v(0)^2)}^{\text{likelihood}}\overbrace{p(v(0)^2)}^{\text{prior}}}{\underbrace{p(x)}_{\text{marginal likelihood}}}
$$

Simple Bayesian Inference

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

- Four components:
  1. Prior distribution: represents "belief" about parameter (velocity squared) before seeing data heights.
  2. Likelihood: gives relation between parameter (velocity squared) and data (heights).
  3. Posterior distribution: represents updated belief about parameters after data is observed.
  4. Marginal likelihood: represents assessment of the quality of the model. Ratios of marginal likelihoods are known as Bayes factors.

Example System: Update our Velocity Belief

- Our initial belief about robot position is given by $p(v(0)^2)$ this is the prior.
- Our belief about height readings given the squared velocity is $p(x_i|v(0)^2)$.
- We combine this likelihood with our prior belief to get the posterior: $p(v(0)^2|x_i)$.
- For several position observations $\{x_i\}_{i=1}^n$ we can apply the formula iteratively.

Gaussian Noise

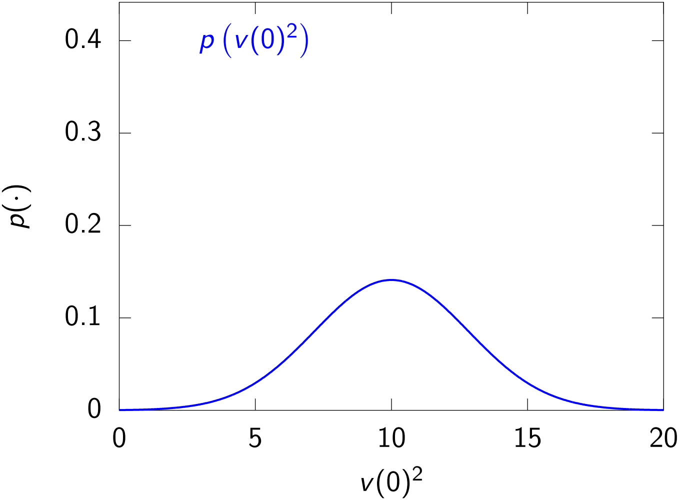


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

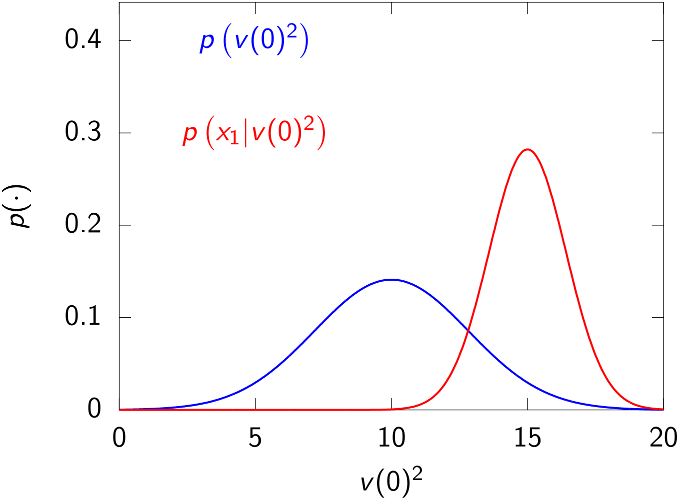


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

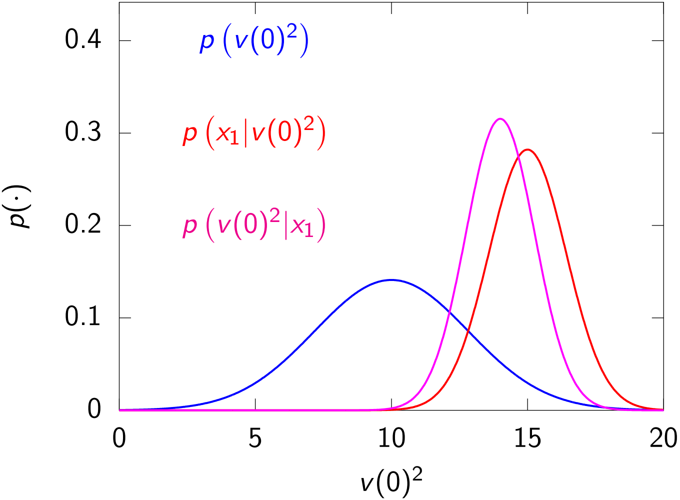


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

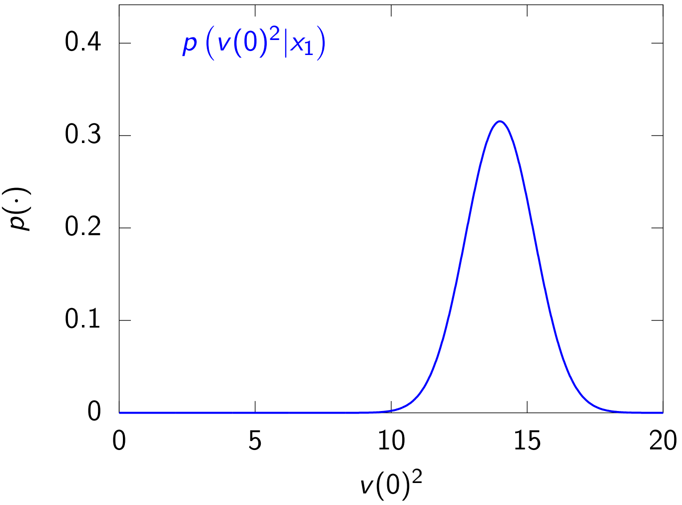


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

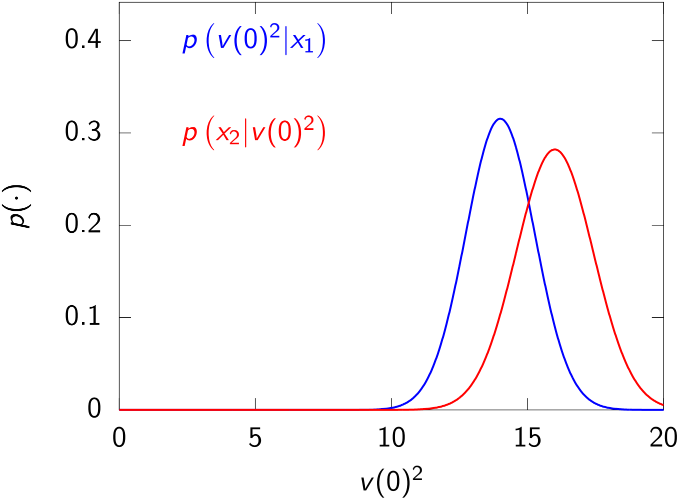


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

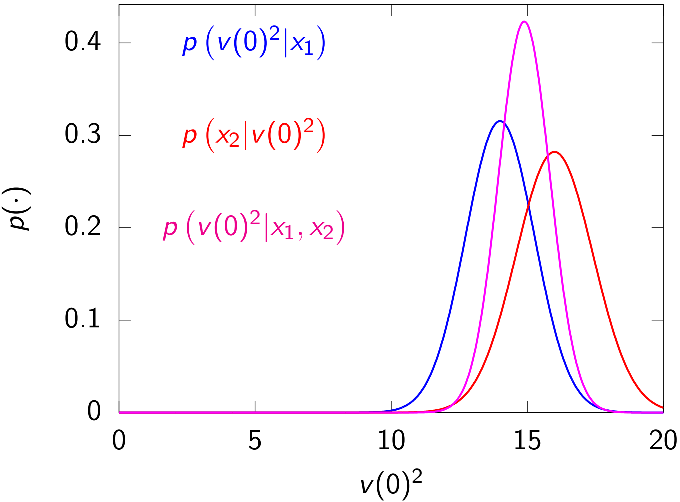


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

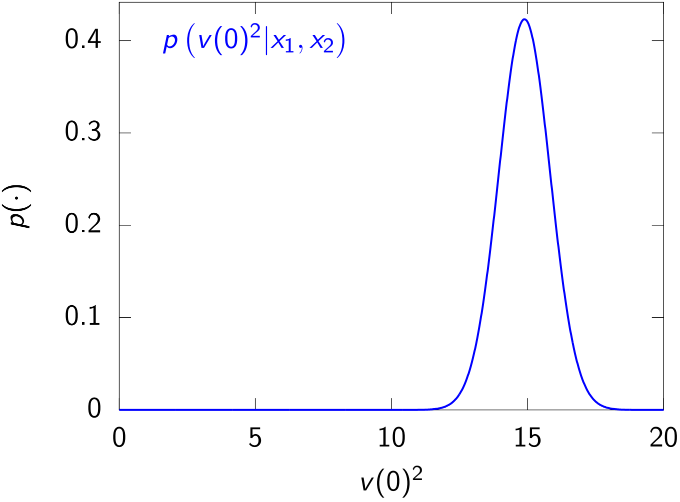


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Gaussian Noise

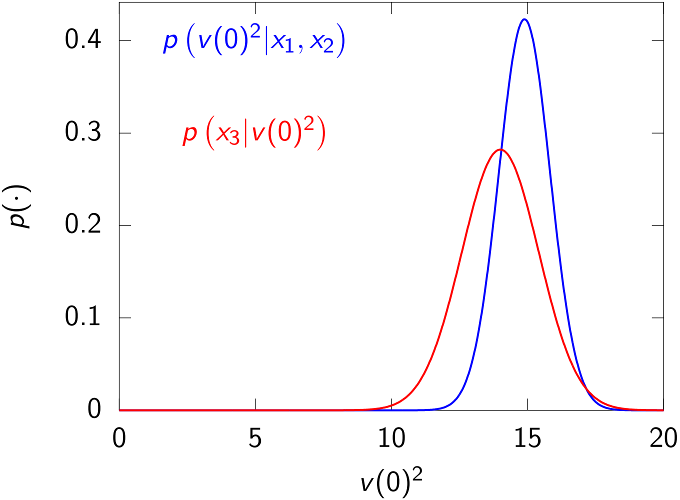


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

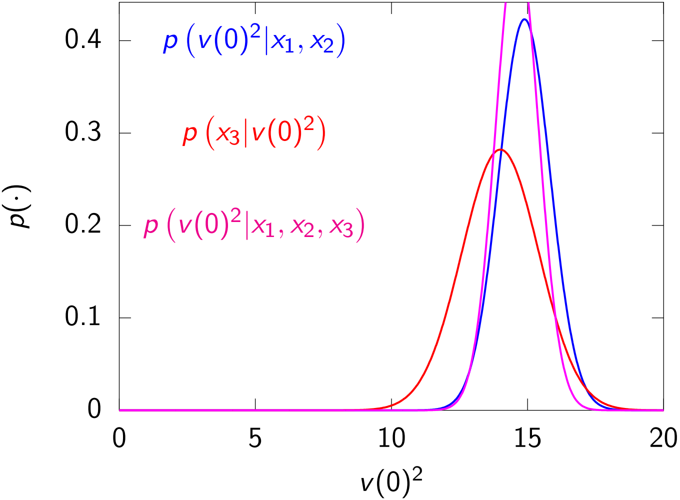


Figure: A Gaussian prior combines with a Gaussian likelihood for a Gaussian posterior.

Posterior Distribution Progression

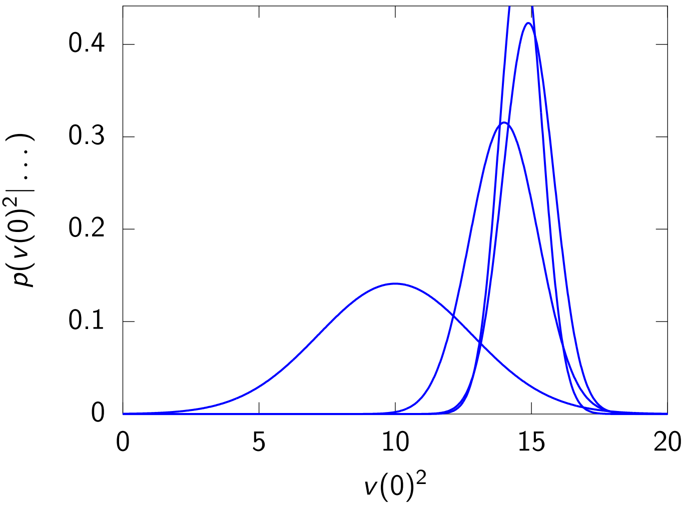


Figure: Progression of our belief about the squared velocity.

Outline

Mathematical Model

- Differential equation model of system.

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

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

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

Mathematical Model

- Differential equation model of system.

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

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

Mathematical Model

- Differential equation model of system.

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

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

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

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

Mathematical Model

- Differential equation model of system.

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

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

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

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

Mathematical Model

- Differential equation model of system.

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

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

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

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

Mathematical Model

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

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

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

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

Mathematical Model

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

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

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

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

Mathematical Model

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

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

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

Mathematical Model

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

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

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

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

Method

Open Access

Ranked prediction of p53 targets using hidden variable dynamic modeling

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

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

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

Published: 31 March 2006

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

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

Response of p53

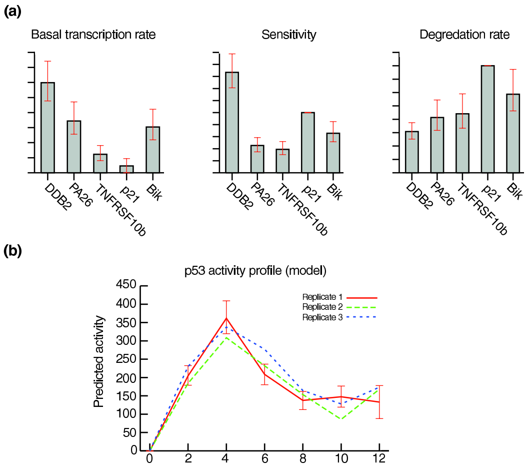


Figure: Results from Barenco et al. (2006). Top is parameter estimates. Bottom is inferred profile.

Respose to p53 ...

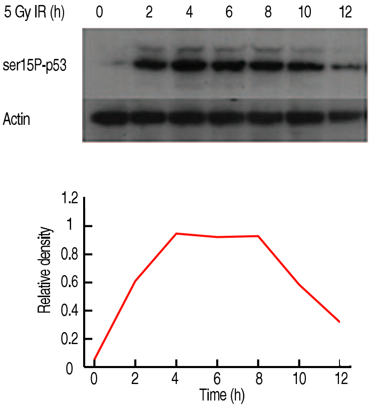


Figure: Results from Barenco et al. (2006). Activity profile of p53 was measured by Western blot to determine the levels of ser-15 phosphorylated p53 (ser15P-p53).

Outline

Two Dimensional Gaussian

- Probability distributions can be higher dimensional.
- Consider height, h/m and weight, w/kg.
- Could sample height from a distribution:

$$p(h) \sim \mathcal{N}\left(1.7, 0.0225\right)$$

- And similarly weight:

$$p(w) \sim \mathcal{N}\left(75, 36\right)$$

Height and Weight Models

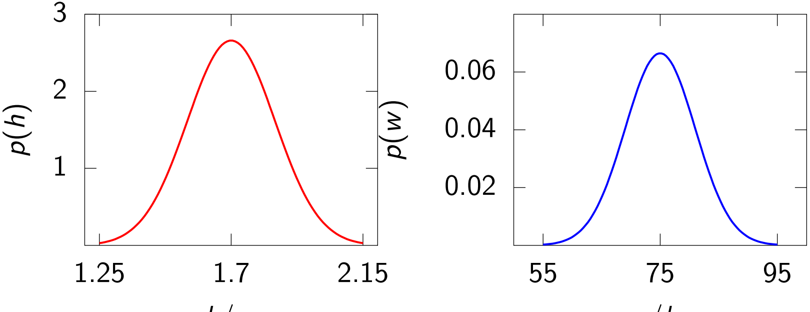
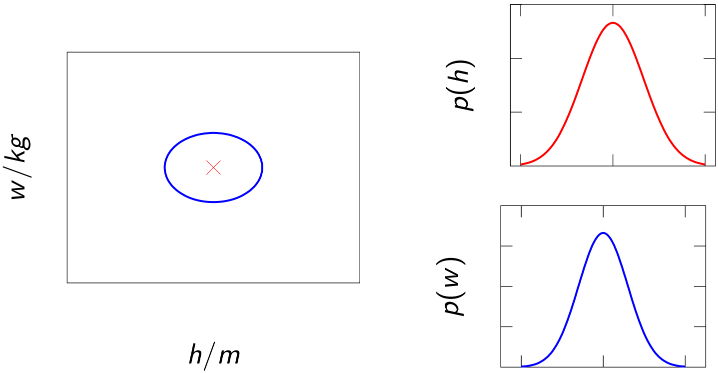
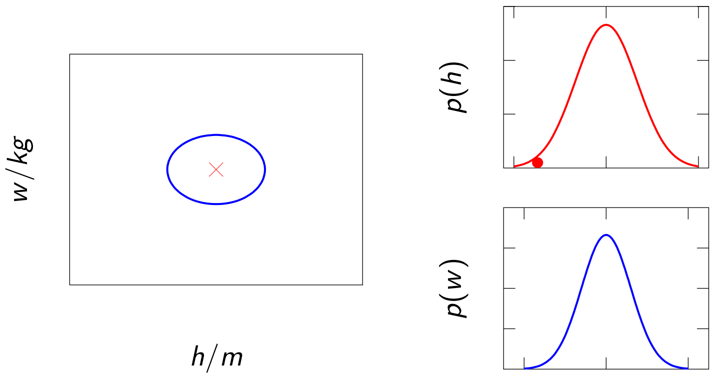


Figure: Gaussian distributions for height and weight.

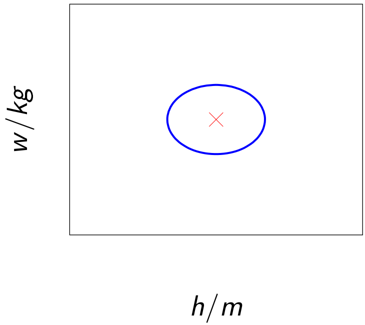
Sampling Two Dimensional Variables

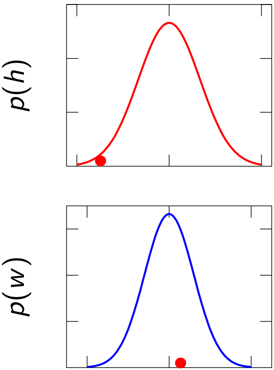


Sampling Two Dimensional Variables

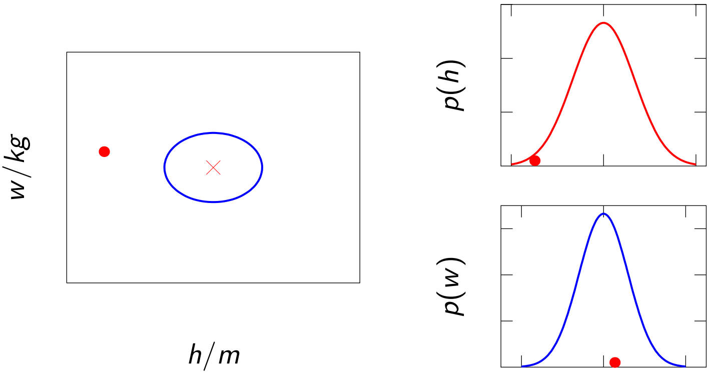


Sampling Two Dimensional Variables

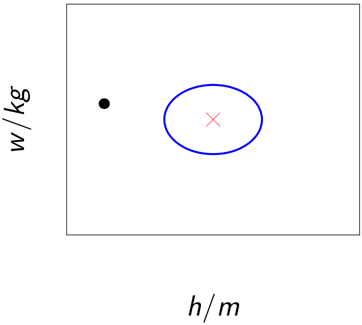


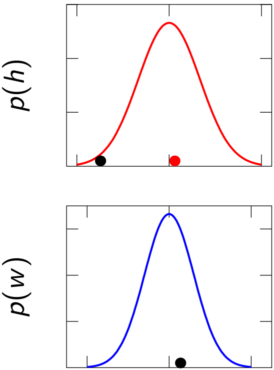


Sampling Two Dimensional Variables

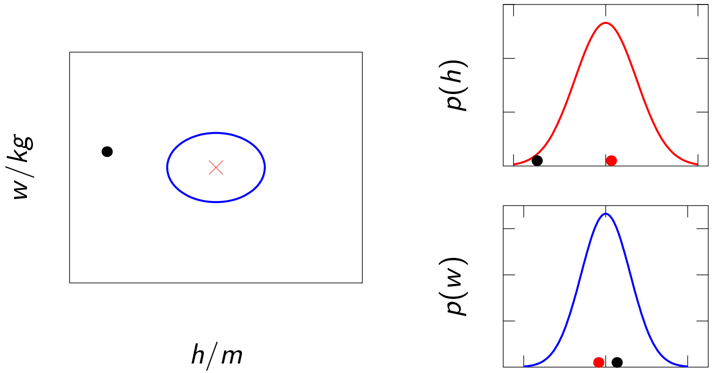


Sampling Two Dimensional Variables

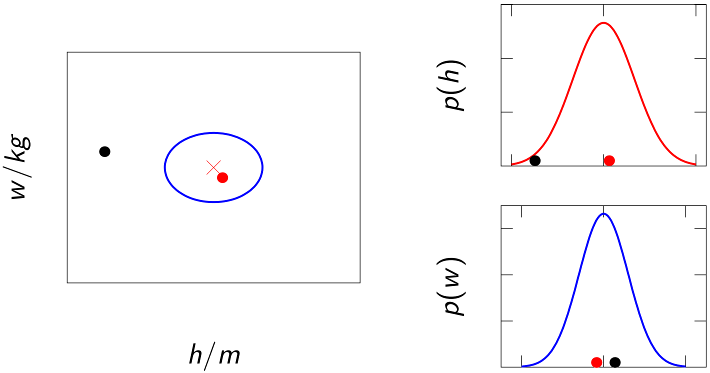




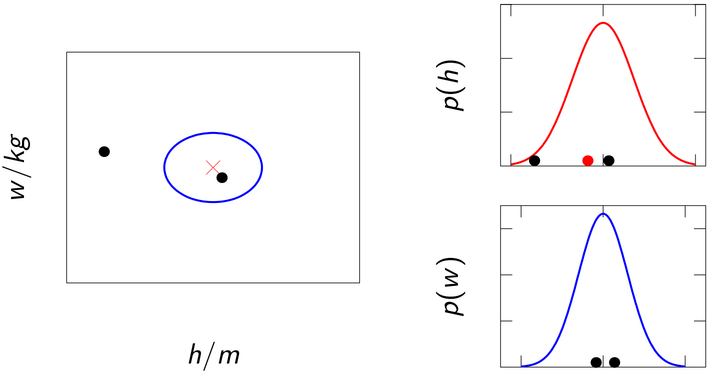
Sampling Two Dimensional Variables



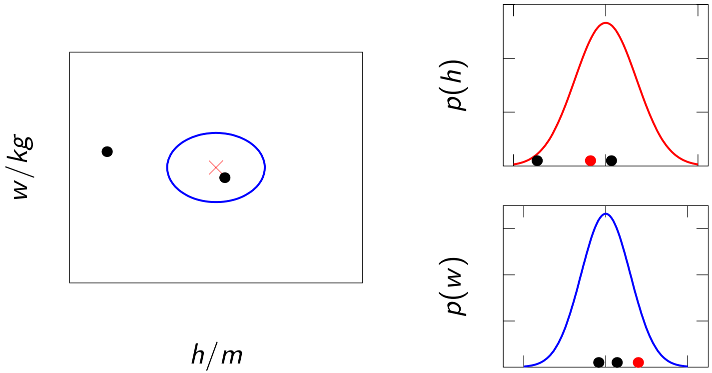
Sampling Two Dimensional Variables



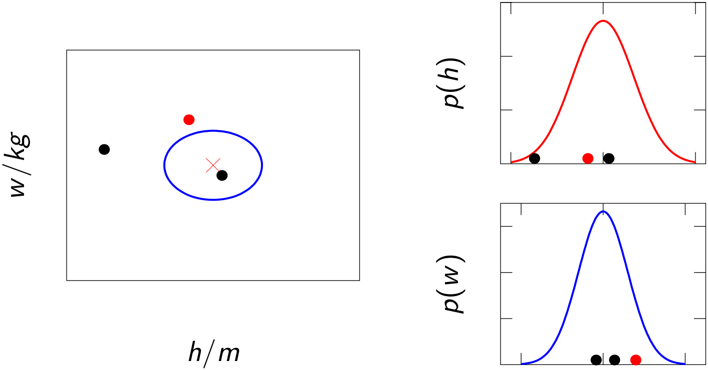
Sampling Two Dimensional Variables



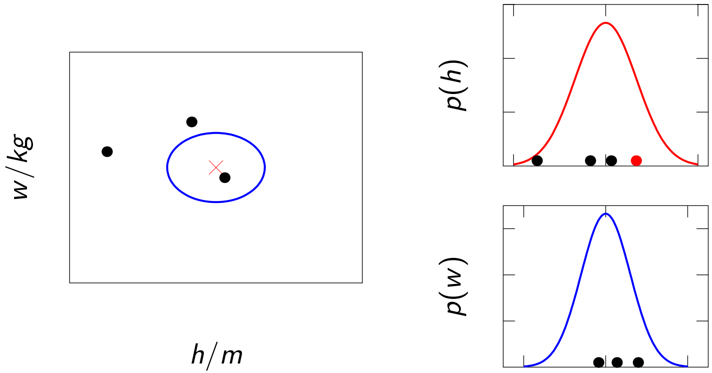
Sampling Two Dimensional Variables



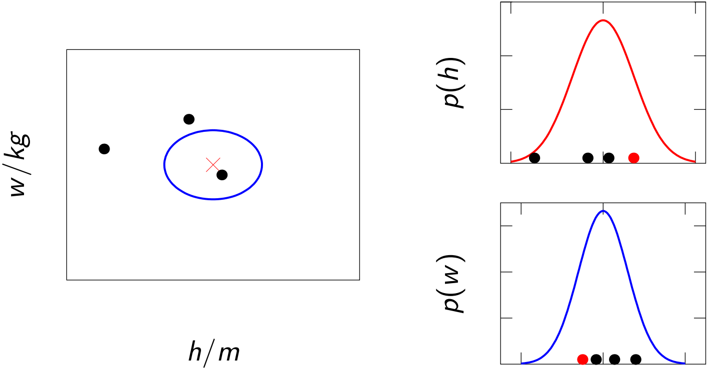
Sampling Two Dimensional Variables



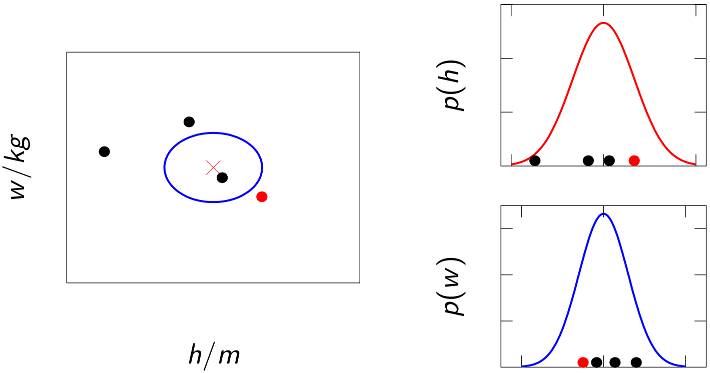
Sampling Two Dimensional Variables



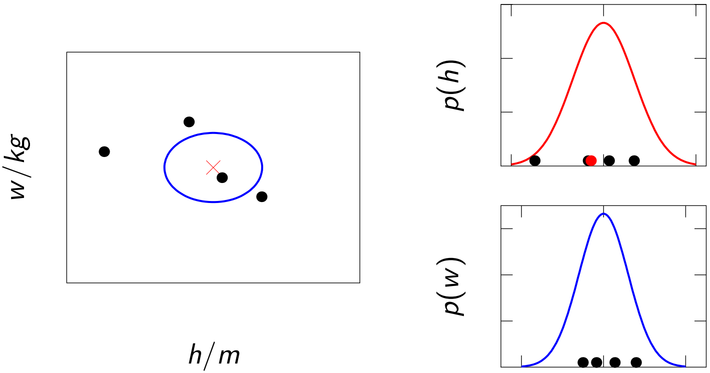
Sampling Two Dimensional Variables



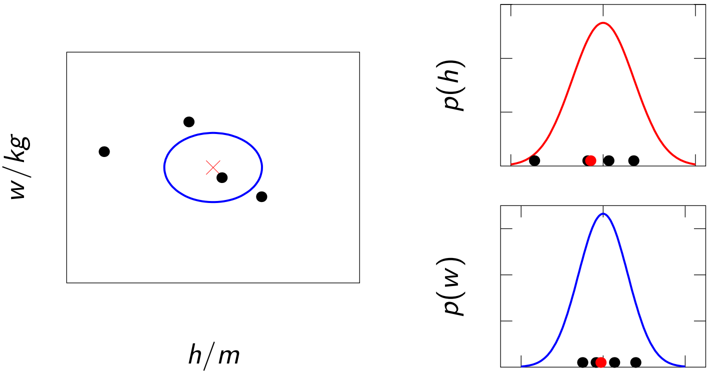
Sampling Two Dimensional Variables



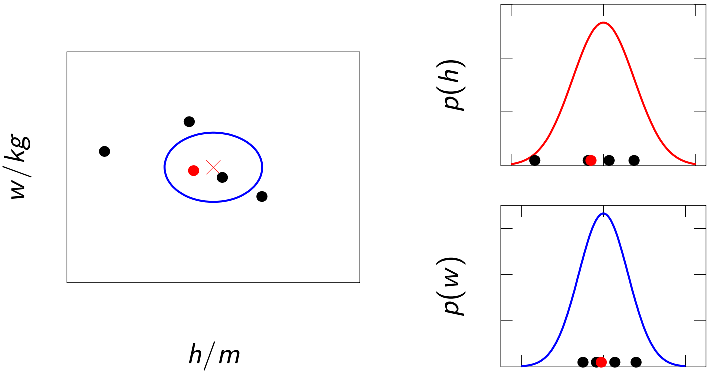
Sampling Two Dimensional Variables



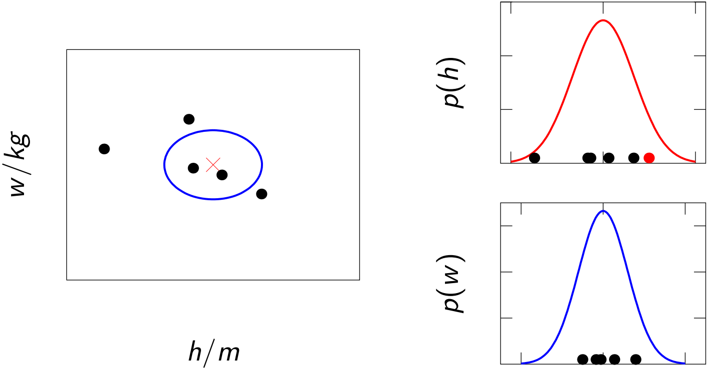
Sampling Two Dimensional Variables



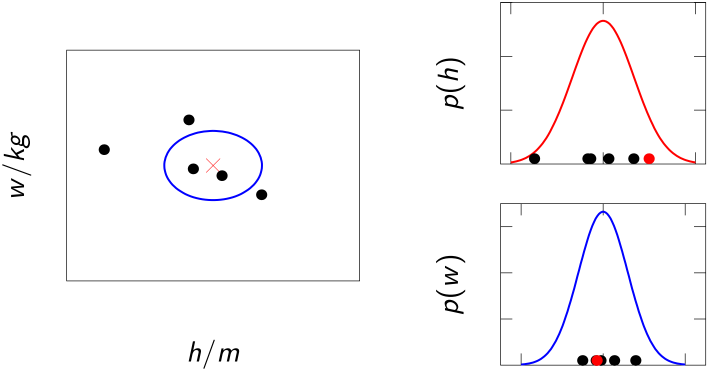
Sampling Two Dimensional Variables



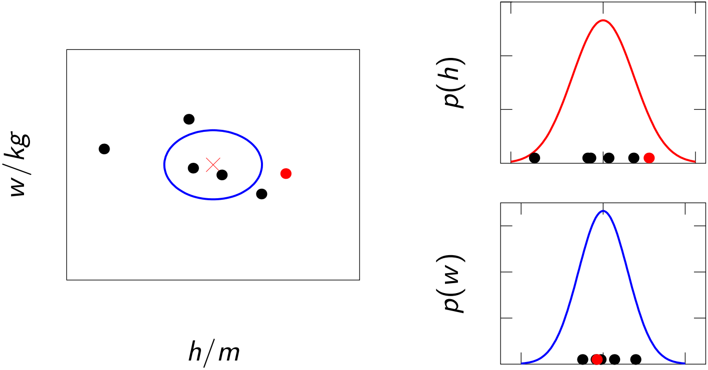
Sampling Two Dimensional Variables



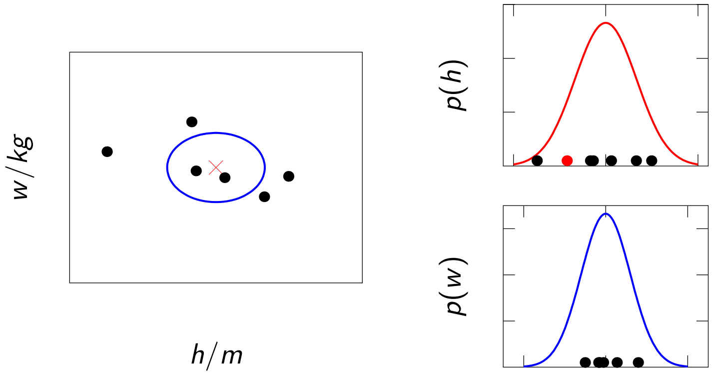
Sampling Two Dimensional Variables



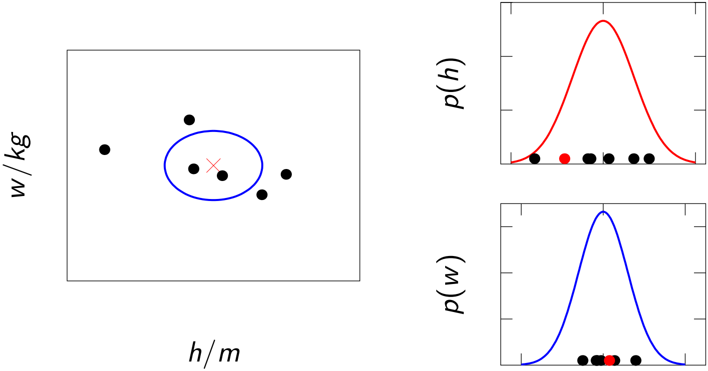
Sampling Two Dimensional Variables



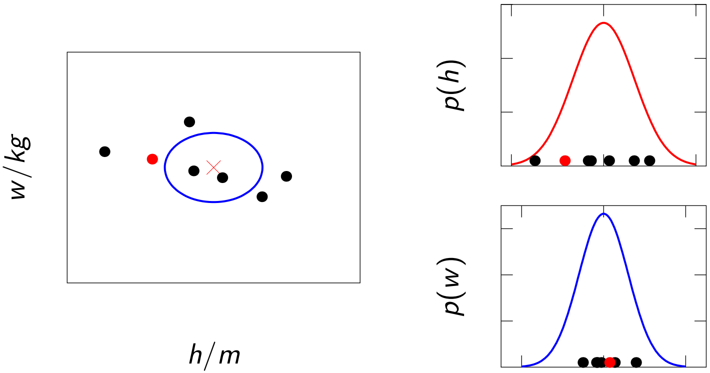
Sampling Two Dimensional Variables



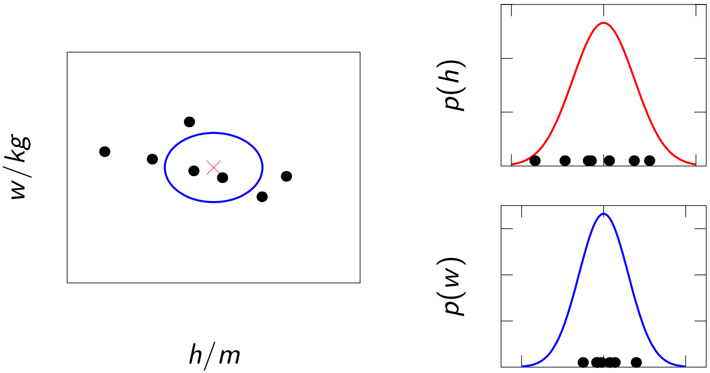
Sampling Two Dimensional Variables



Sampling Two Dimensional Variables



Sampling Two Dimensional Variables



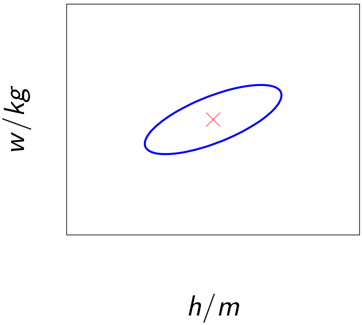
Independence Assumption

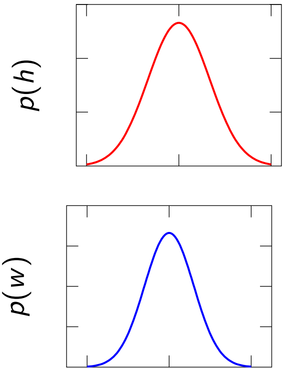
- This assumes height and weight are independent.

$$p(h, w) = p(h)p(w)$$

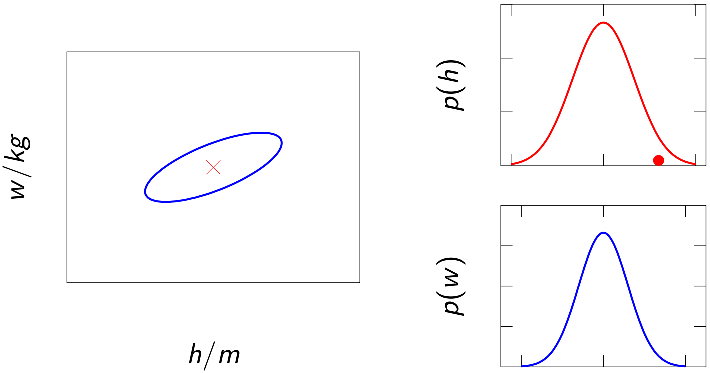
- In reality they are dependent (body mass index) $= \frac{w}{h^2}$.

Sampling Two Dimensional Variables

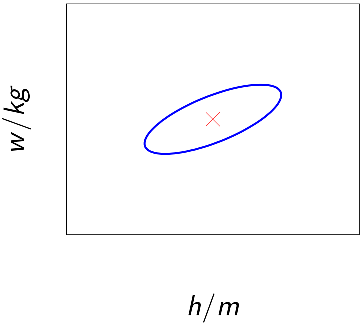


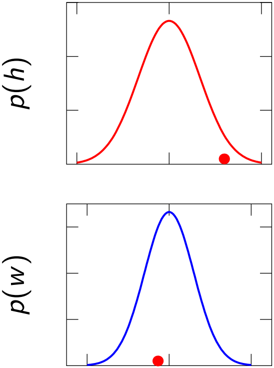


Sampling Two Dimensional Variables

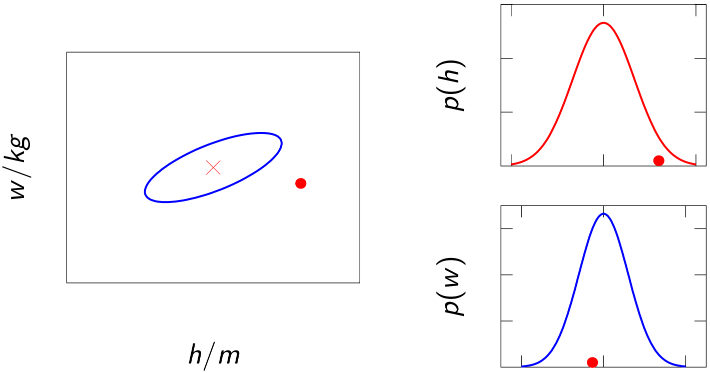


Sampling Two Dimensional Variables

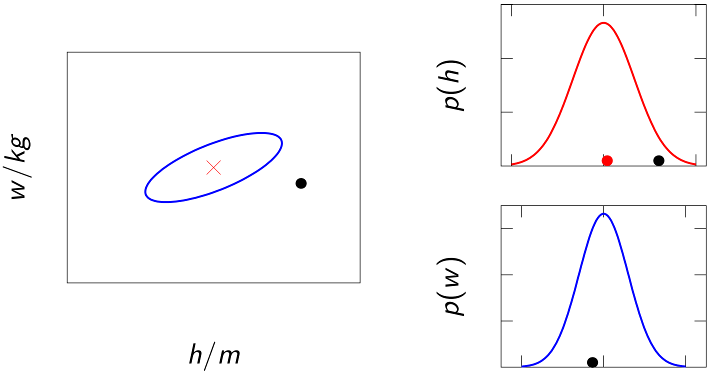




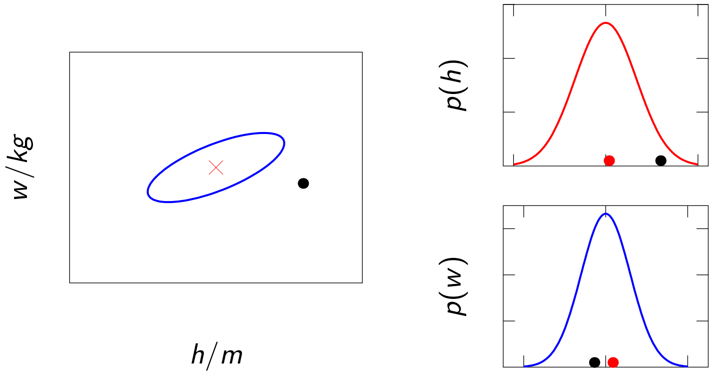
Sampling Two Dimensional Variables



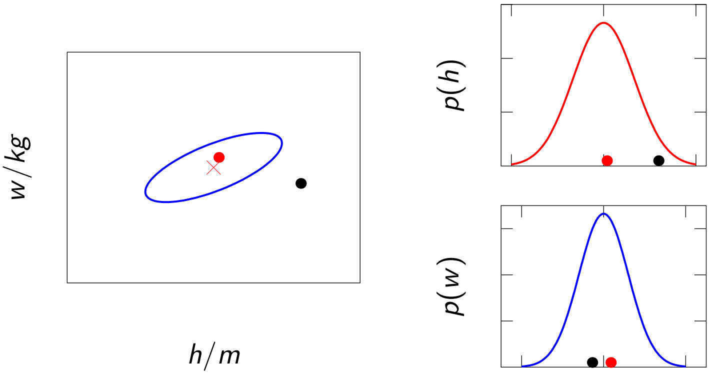
Sampling Two Dimensional Variables



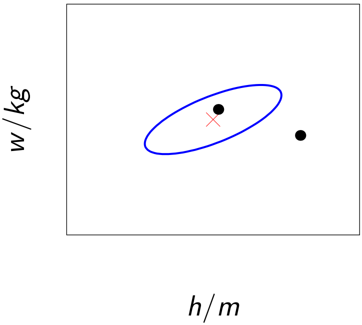
Sampling Two Dimensional Variables

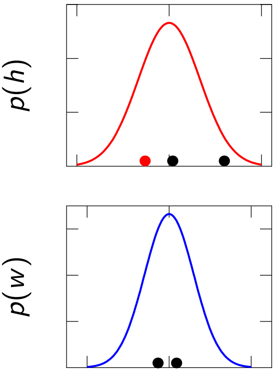


Sampling Two Dimensional Variables

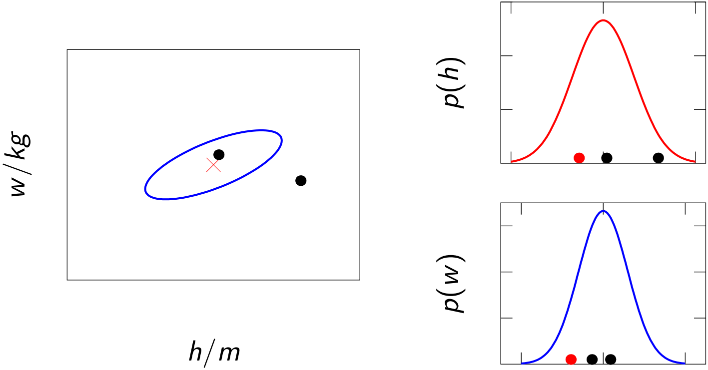


Sampling Two Dimensional Variables

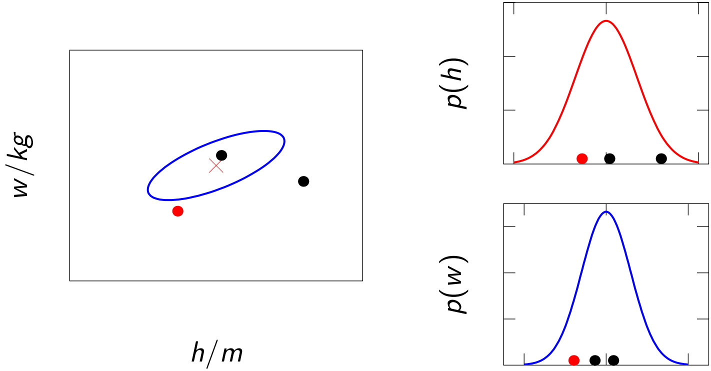




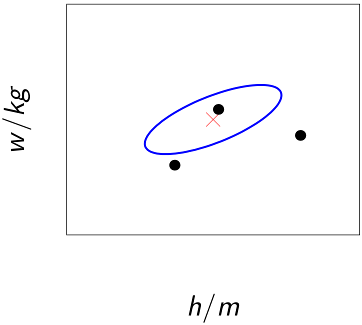
Sampling Two Dimensional Variables

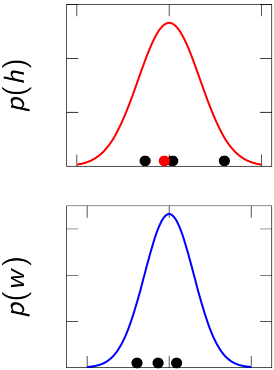


Sampling Two Dimensional Variables

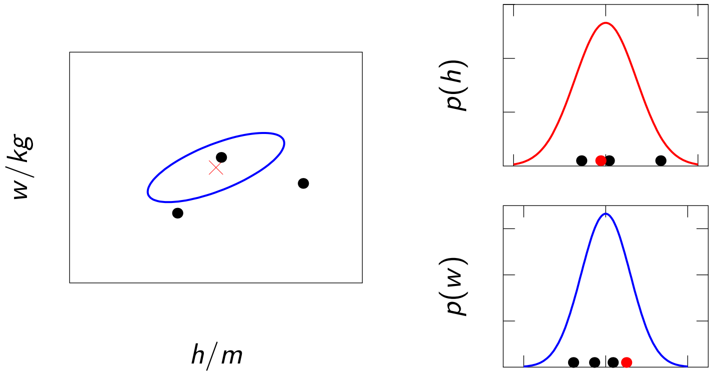


Sampling Two Dimensional Variables

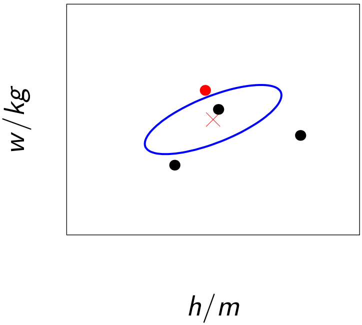


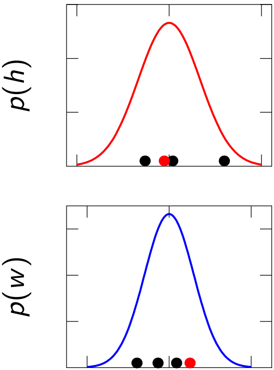


Sampling Two Dimensional Variables

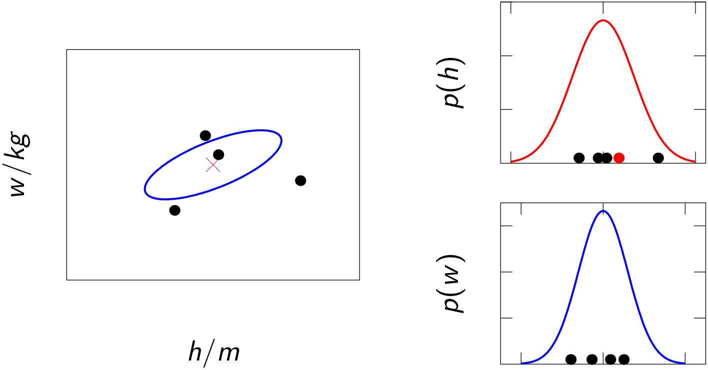


Sampling Two Dimensional Variables

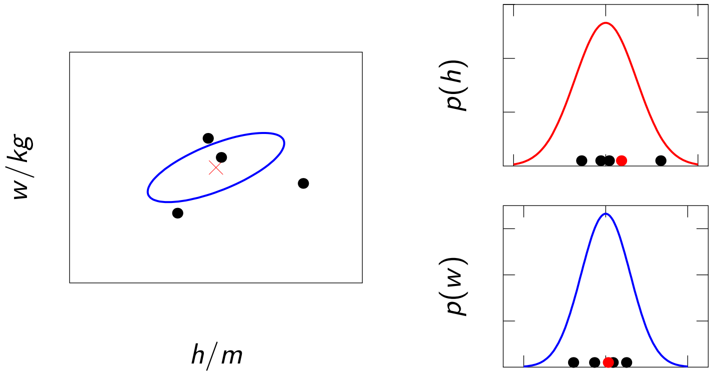




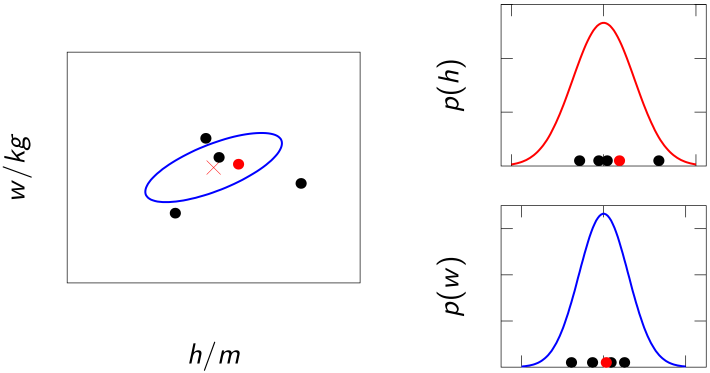
Sampling Two Dimensional Variables



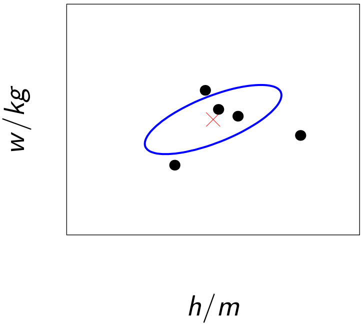
Sampling Two Dimensional Variables

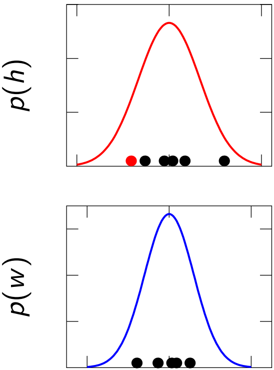


Sampling Two Dimensional Variables

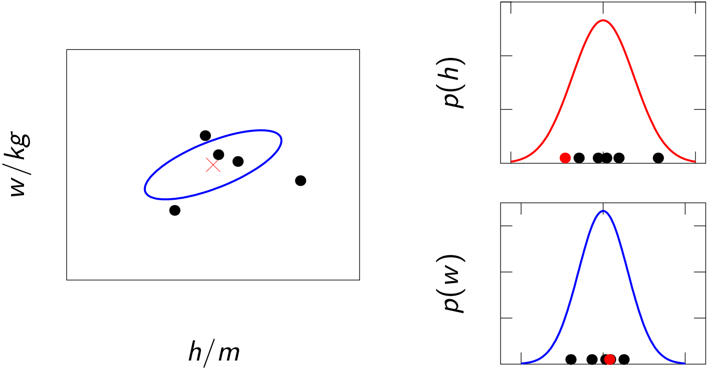


Sampling Two Dimensional Variables

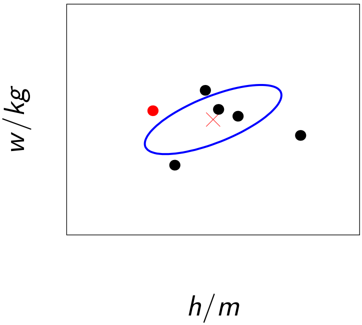


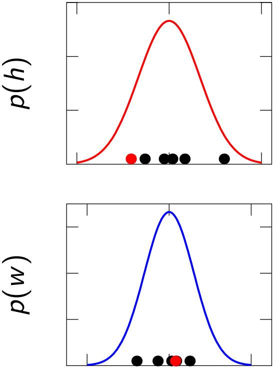


Sampling Two Dimensional Variables

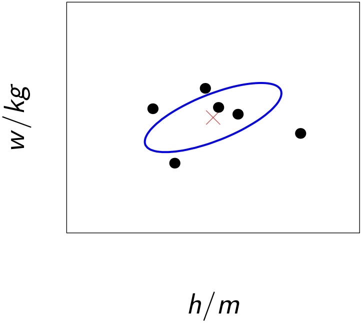


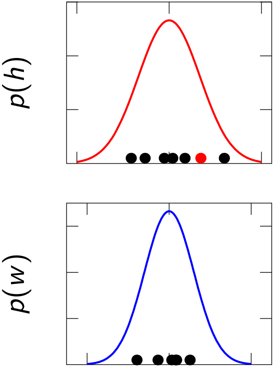
Sampling Two Dimensional Variables



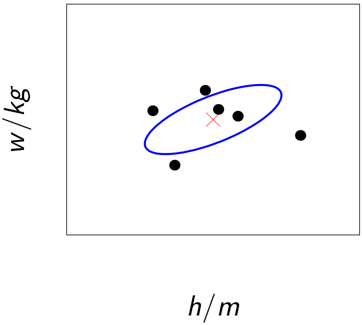


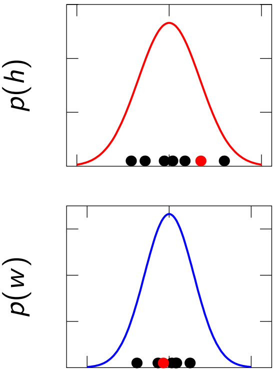
Sampling Two Dimensional Variables



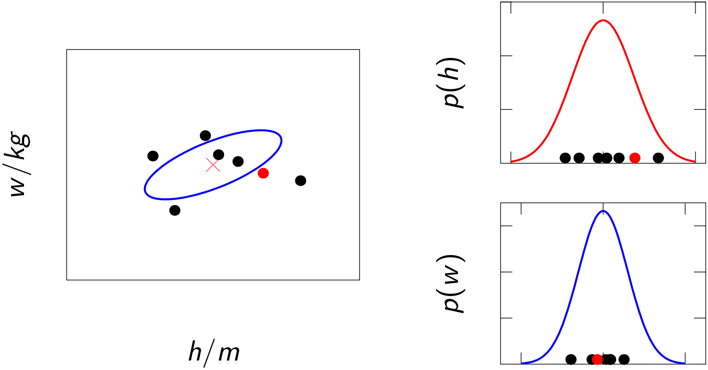


Sampling Two Dimensional Variables

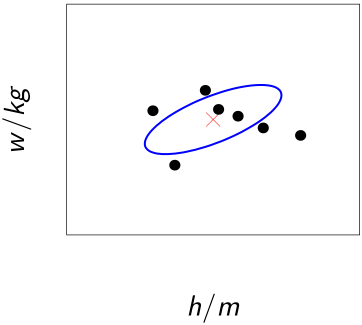


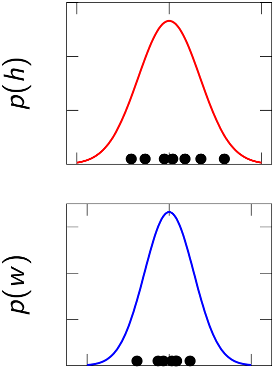


Sampling Two Dimensional Variables



Sampling Two Dimensional Variables





Correlated Gaussian

- Second Gaussian correlated.
- Form from original Gaussian by elongating one direction and rotating.
- For rotation matrix $\mathbf{R}$ and scaling matrix

$$\mathbf{L} = \begin{bmatrix} \ell_1 & 0 \\ 0 & \ell_2 \end{bmatrix}$$

this gives a covariance matrix:

$$\mathbf{K} = \mathbf{R}\mathbf{L}^2\mathbf{R}^\top$$

Gaussian Distribution

Zero mean Gaussian distribution

- A multi-variate Gaussian distribution is defined by a mean and a covariance matrix.

$$\mathcal{N}\left(\mathbf{p}|\mu, \mathbf{K}\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\mathbf{K}\right|^{\frac{1}{2}}} \exp\left(-\frac{\left(\mathbf{p}-\mu\right)^{\top} \mathbf{K}^{-1} \left(\mathbf{p}-\mu\right)}{2}\right).$$

- We will consider the special case where the mean is zero,

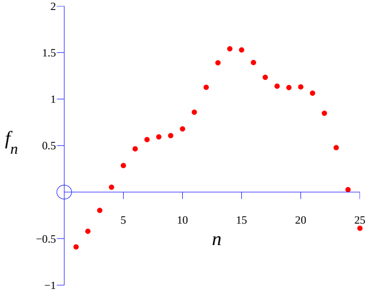
$$\mathcal{N}\left(\mathbf{p}|\mathbf{0}, \mathbf{K}\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\mathbf{K}\right|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{p}^{\top}\mathbf{K}^{-1}\mathbf{p}}{2}\right).$$

Sampling a Function

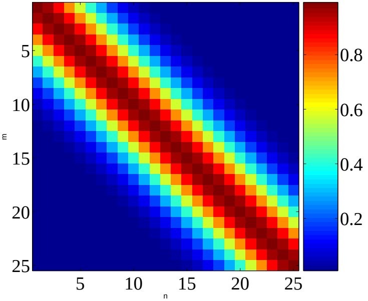
Multi-variate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{p} = [p_1, p_2 \ldots p_{25}]$.
- We will plot these points against their index.

Gaussian Distribution Sample



(a) A 25 dimensional correlated random variable (values ploted against index)



(b) colormap showing correlations between dimensions

Figure: A sample from a 25 dimensional Gaussian distribution.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_i and p_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_i and p_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_i and p_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points p_i and p_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Prediction of p_2 from p_1

```
demGpCov2D([1 2])
```

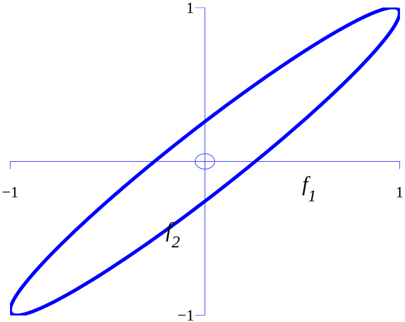


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of p_2 from p_1

```
demGpCov2D([1 2])
```

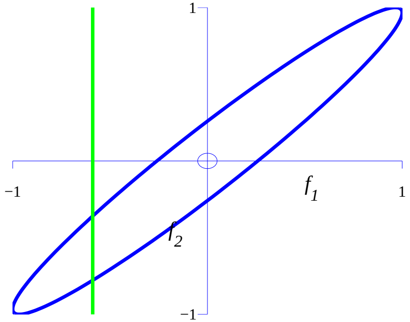


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of p_2 from p_1

```
demGpCov2D([1 2])
```

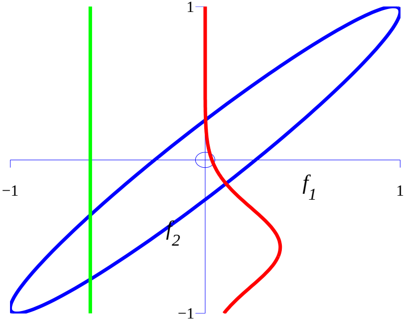


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of p_5 from p_1

```
demGpCov2D([1 5])
```

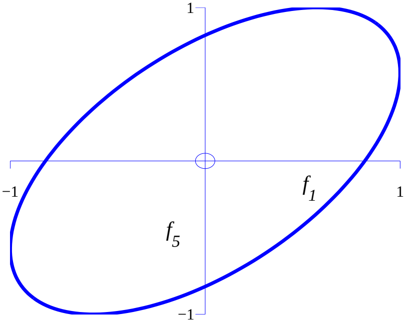


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of p_5 from p_1

```
demGpCov2D([1 5])
```

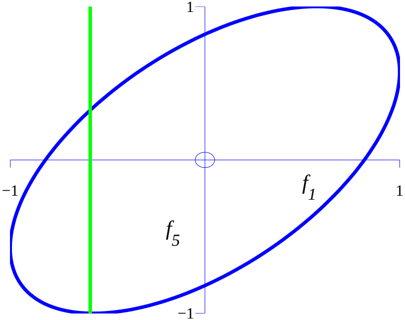


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of p_5 from p_1

```
demGpCov2D([1 5])
```

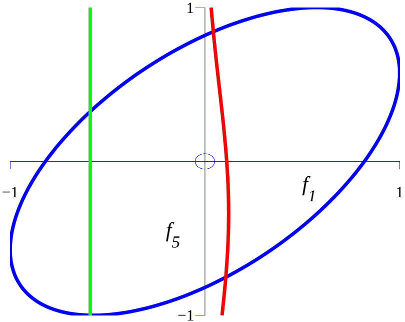


Figure: Covariance for $\begin{bmatrix} p_1 \\ p_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

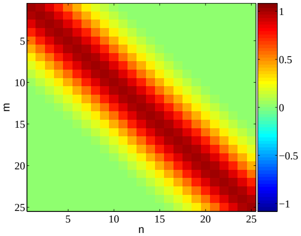
Covariance Functions

Where did this covariance matrix come from?

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

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

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



Covariance Samples

`demCovFuncSample`

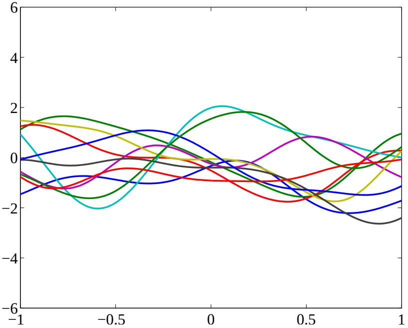


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

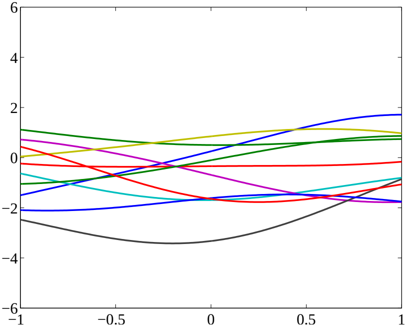


Figure: Exponentiated quadratic kernel with $\ell = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

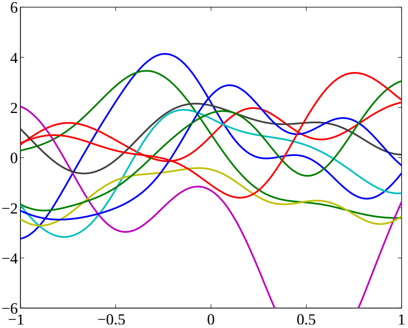


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

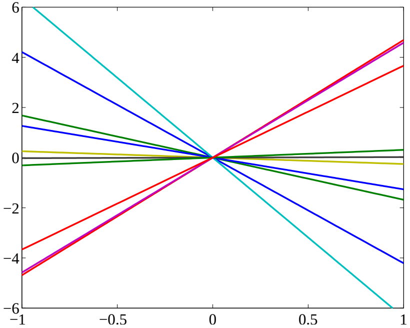


Figure: Linear covariance function, $\alpha = 16$.

Covariance Samples

`demCovFuncSample`

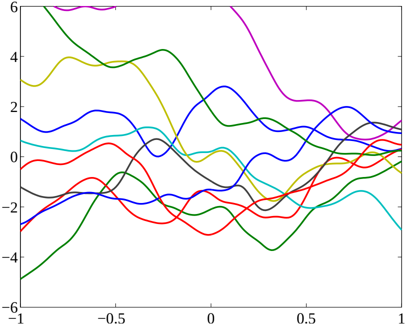


Figure: MLP covariance function, $\sigma_w^2 = 100$, $\sigma_b^2 = 100$, $\alpha = 8$.

Covariance Samples

`demCovFuncSample`

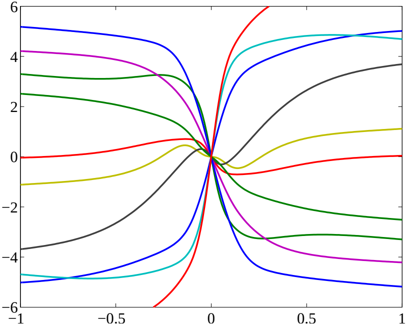


Figure: MLP covariance function, $\sigma_w^2 = 100$, $\sigma_b^2 = 0$, $\alpha = 8$.

Covariance Samples

`demCovFuncSample`

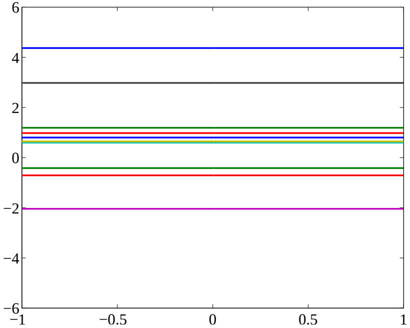


Figure: Bias term, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

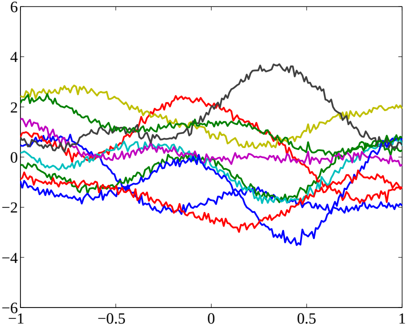


Figure: Exponentiated quadratic $\ell = 0.3$, $\alpha = 1$ plus bias term with $\alpha = 1$ plus white noise with $\alpha = 0.01$.

Covariance Samples

`demCovFuncSample`

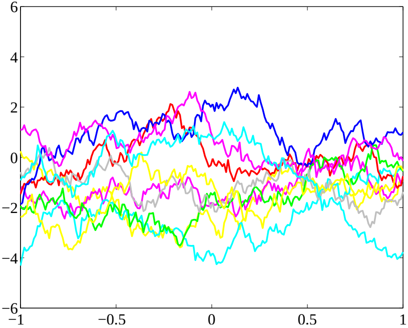


Figure: Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function $\ell = 1$, $\alpha = 4$.

Gaussian Process Interpolation

`demInterpolation`

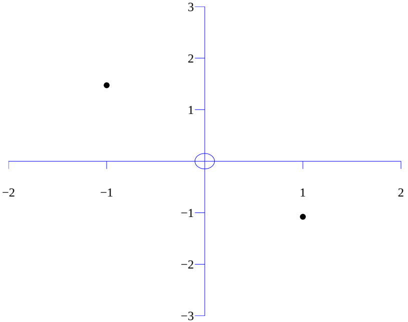


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

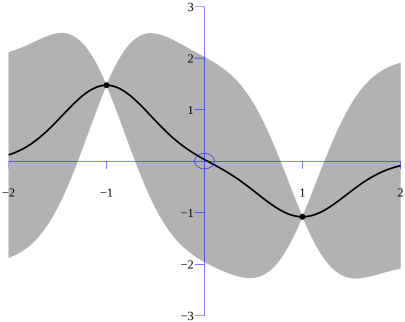


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

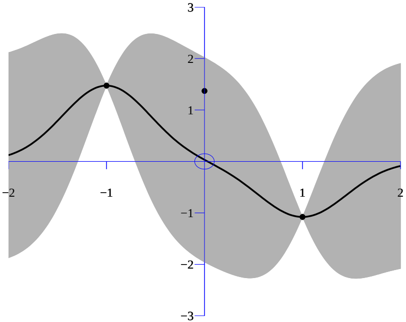


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

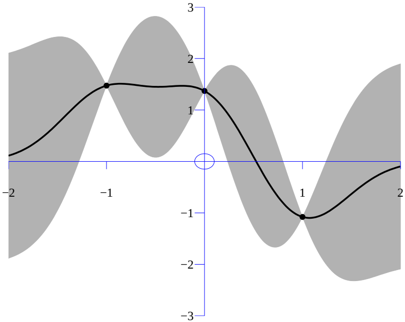


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

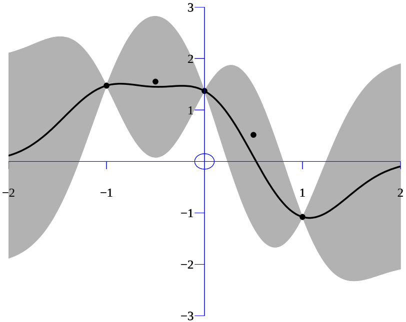


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

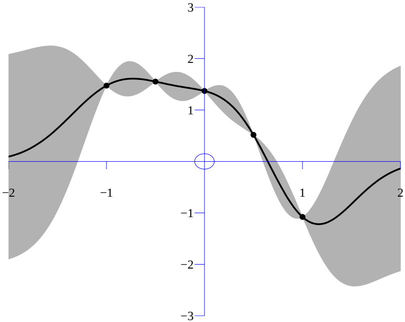


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

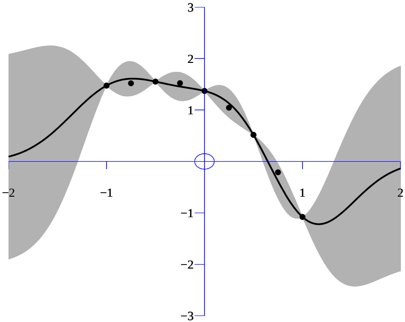


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

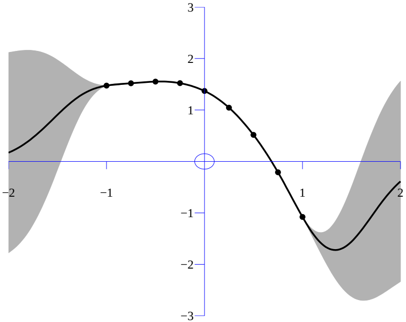


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Noise Models

Graph of a GP

- Relates input variables, $\mathbf{t}$, to vector, $\mathbf{m}$, through $\mathbf{p}$ given kernel parameters $\boldsymbol{\theta}$.
- Plate notation indicates independence of $m_i|p_i$.
- Noise model, $p\left(m_i|p_i\right)$ can take several forms.
- Simplest is Gaussian noise.

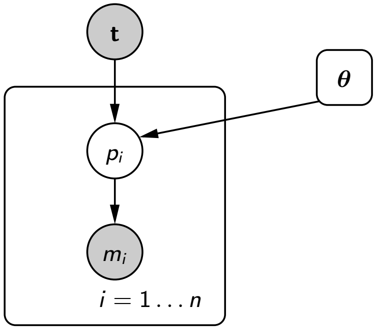


Figure: The Gaussian process depicted graphically.

Gaussian Noise

- Gaussian noise model,

$$p\left(m_i|p_i\right) = \mathcal{N}\left(m_i|p_i, \sigma^2\right)$$

where σ^2 is the variance of the noise.

- Equivalent to a covariance function of the form

$$k(t_i, t_j) = \delta_{i,j}\sigma^2$$

where $\delta_{i,j}$ is the Kronecker delta function.

- Additive nature of Gaussians means we can simply add this term to existing covariance matrices.

Gaussian Process Regression

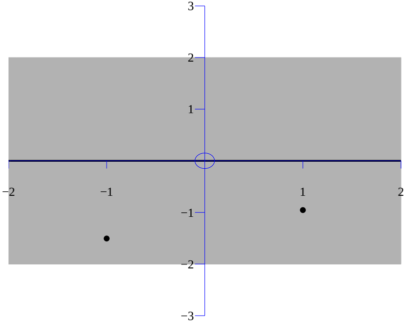


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

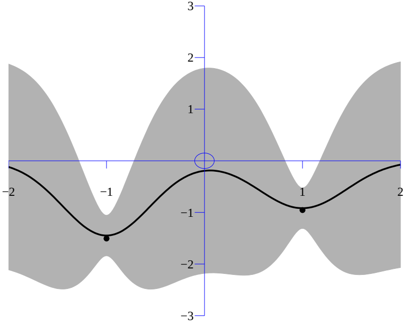


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

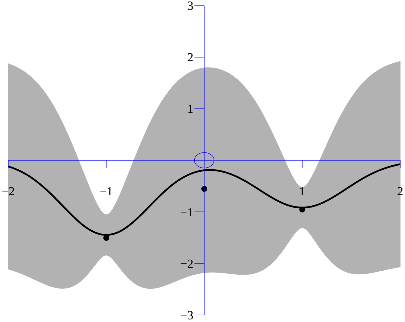


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

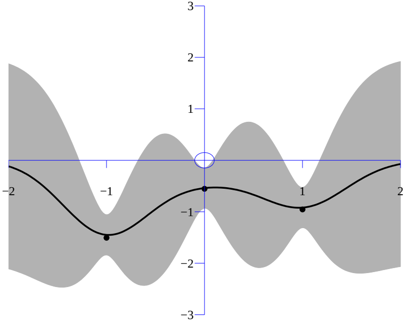


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

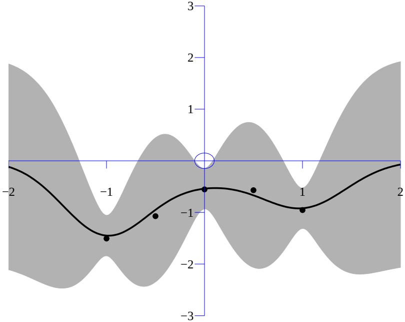


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

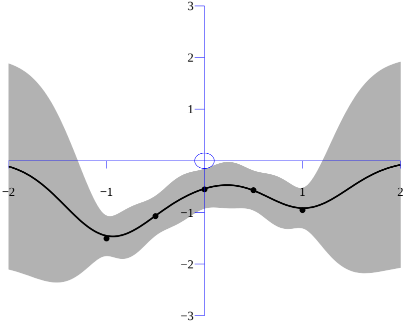


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

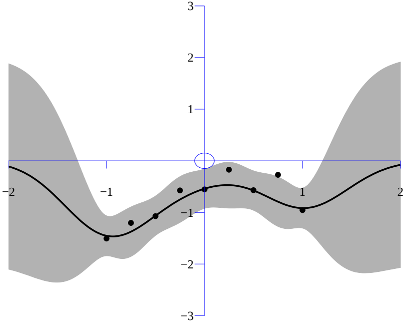


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

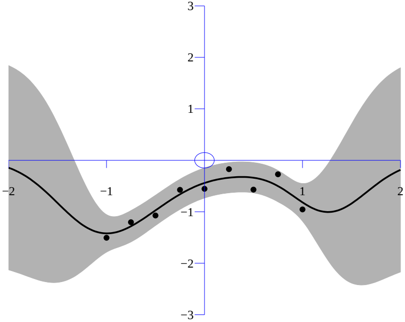
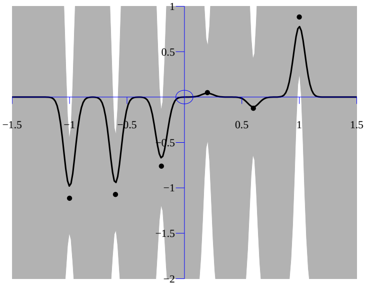
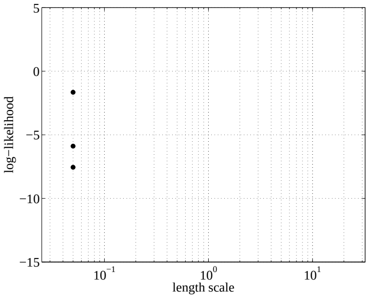


Figure: Examples include WiFi localization, C14 callibration curve.

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

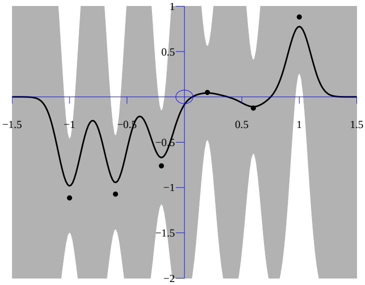


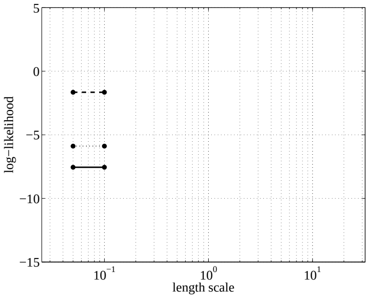


$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0}, \mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top\mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

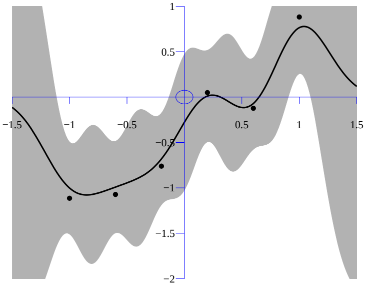


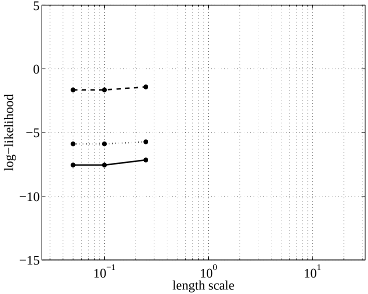


$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0}, \mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top\mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

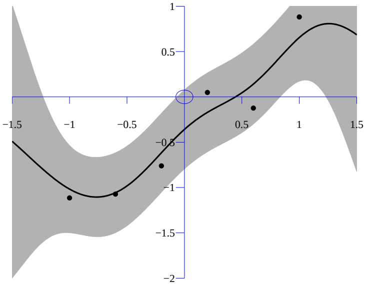


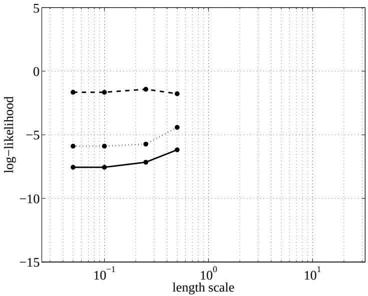


$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0}, \mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^{\top}\mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

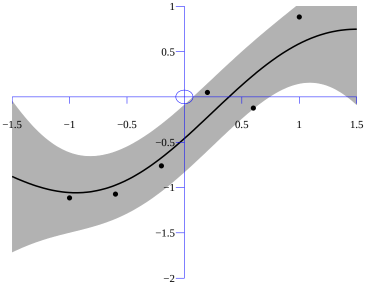


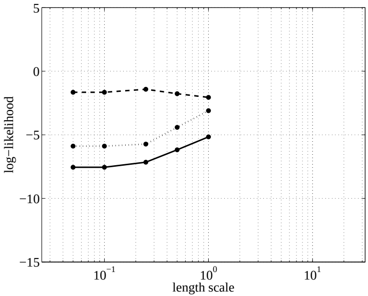


$$\log \mathcal{N}(\mathbf{m}|\mathbf{0}, \mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

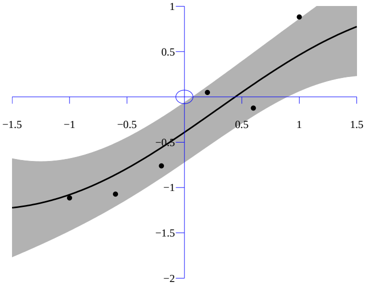


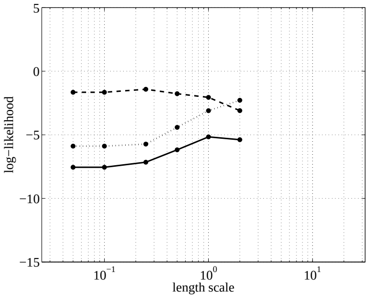


$$\log \mathcal{N}(\mathbf{m}|\mathbf{0}, \mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

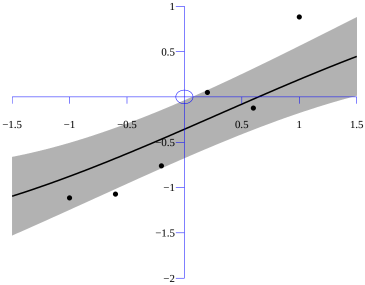


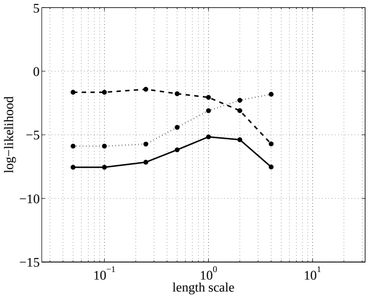


$$\log \mathcal{N}(\mathbf{m}|\mathbf{0}, \mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

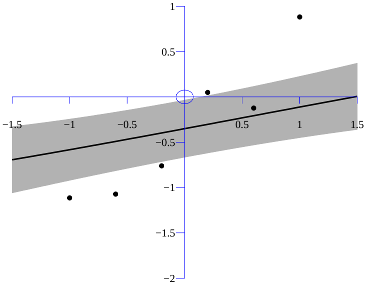


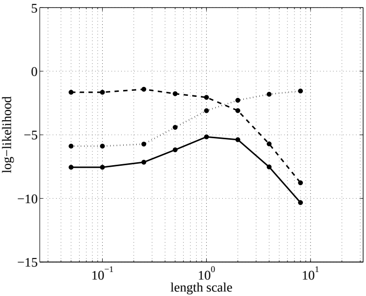


$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0},\mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

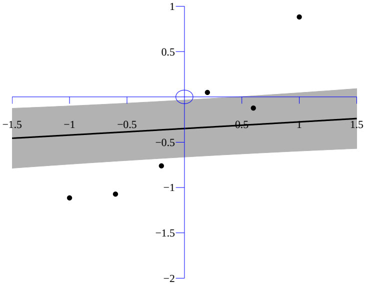


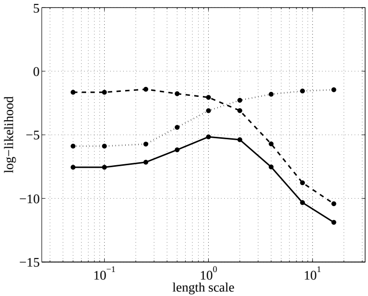


$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0}, \mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?





$$\log \mathcal{N}\left(\mathbf{m}|\mathbf{0}, \mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{m}^\top \mathbf{K}^{-1}\mathbf{m}}{2}$$

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

Example: Transcriptional Regulation

- First Order Differential Equation

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

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

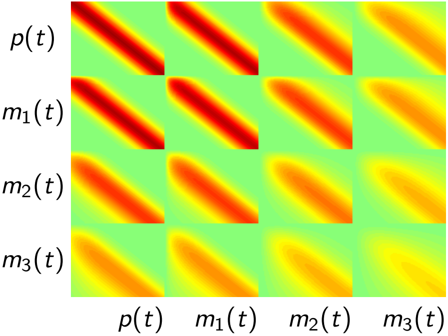
Covariance for Transcription Model

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



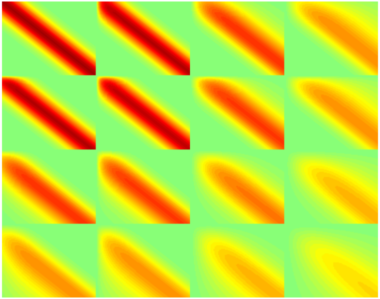
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



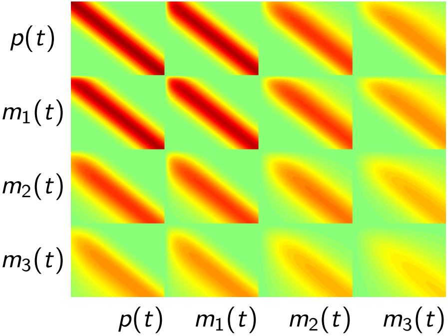
Covariance for Transcription Model

RBF covariance function for $p(t)$

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

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

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



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

- `simSample`

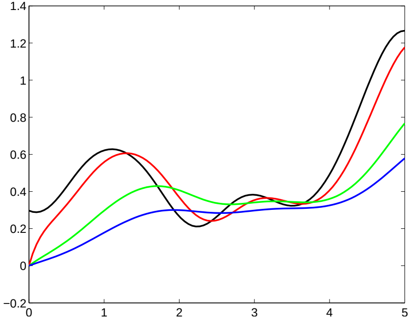


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

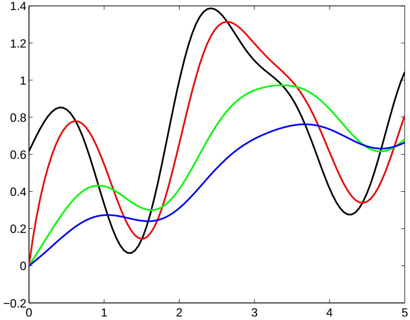


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

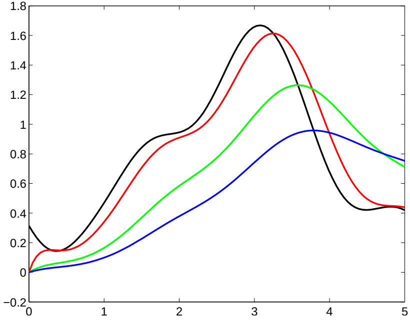


Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

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

- `simSample`

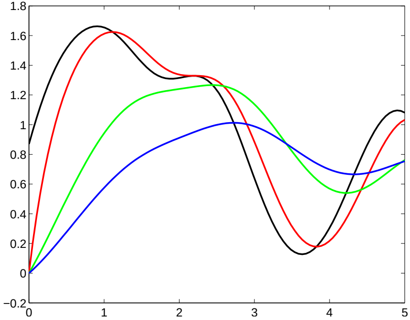
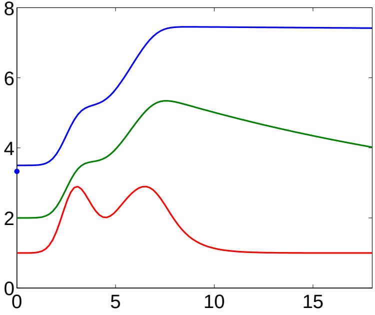


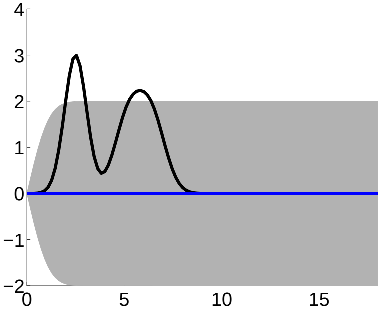
Figure: Joint samples from the ODE covariance, *black*: $p(t)$, *red*: $m_1(t)$ (high decay/sensitivity), *green*: $m_2(t)$ (medium decay/sensitivity) and *blue*: $m_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



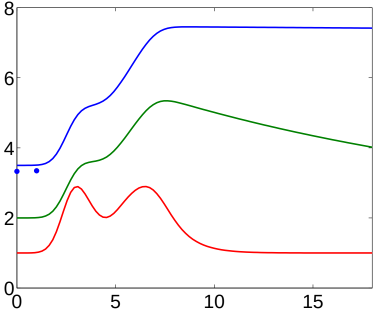
True "gene profiles" and noisy observations.



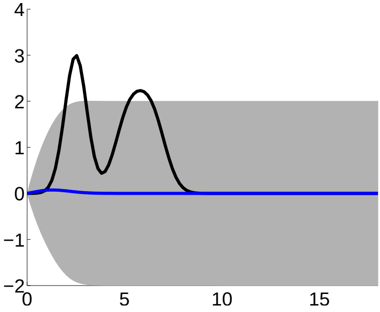
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



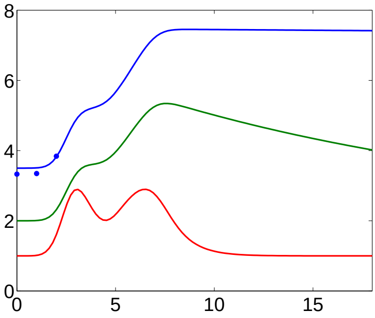
True "gene profiles" and noisy observations.



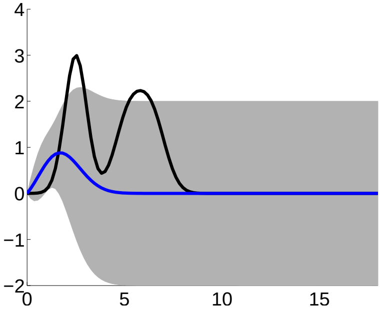
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



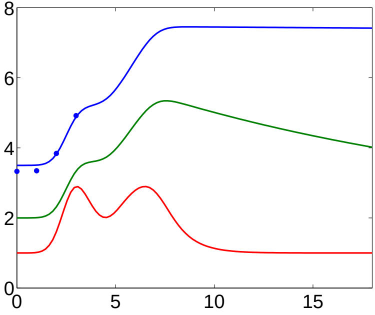
True "gene profiles" and noisy observations.



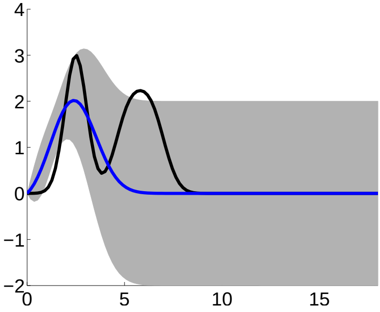
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



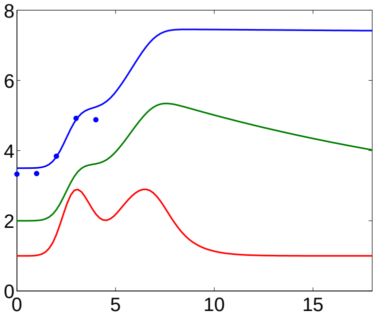
True "gene profiles" and noisy observations.



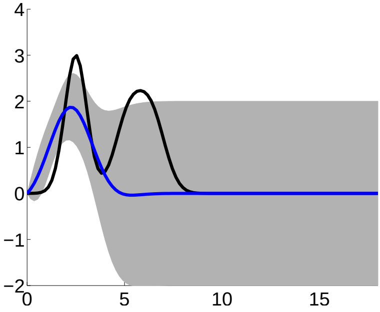
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



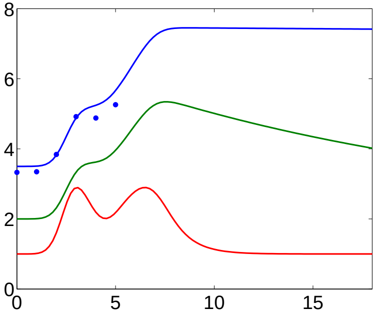
True “gene profiles” and noisy observations.



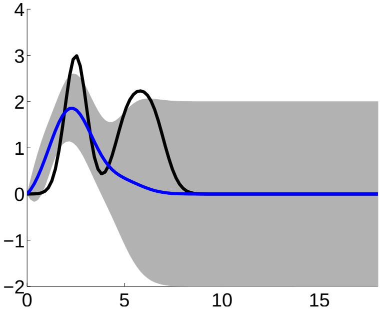
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



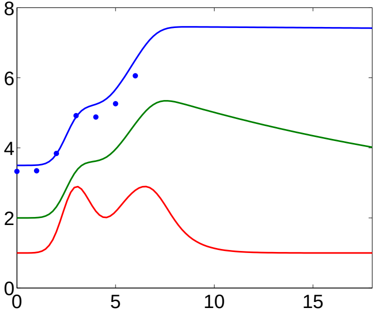
True "gene profiles" and noisy observations.



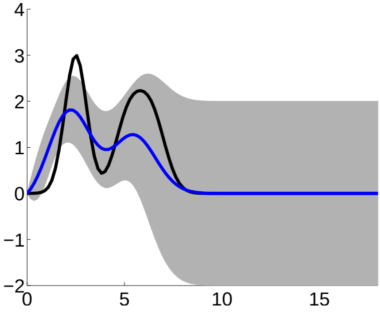
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



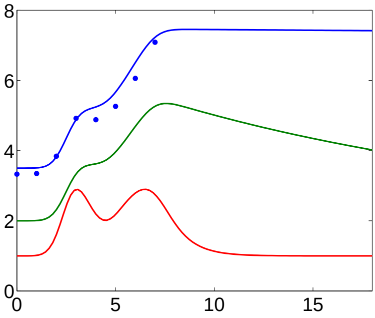
True "gene profiles" and noisy observations.



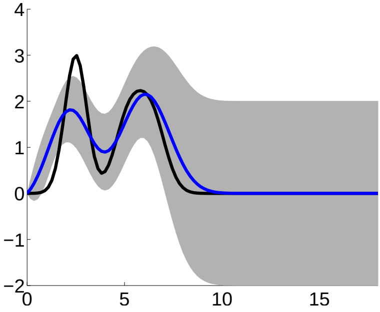
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



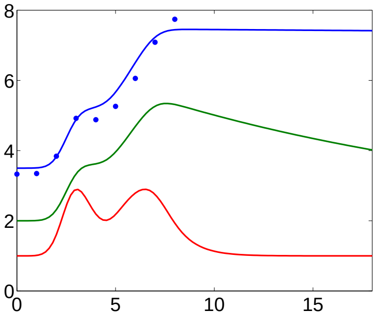
True "gene profiles" and noisy observations.



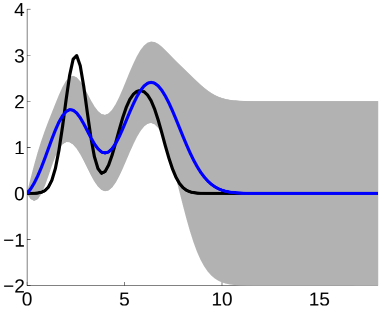
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



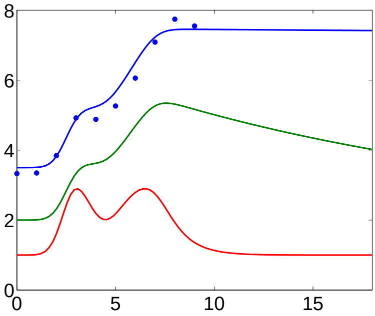
True "gene profiles" and noisy observations.



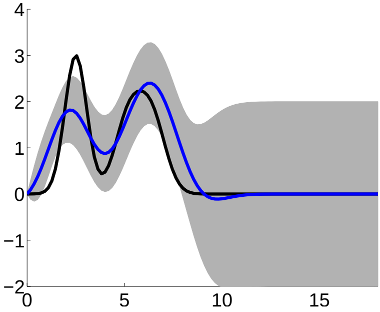
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



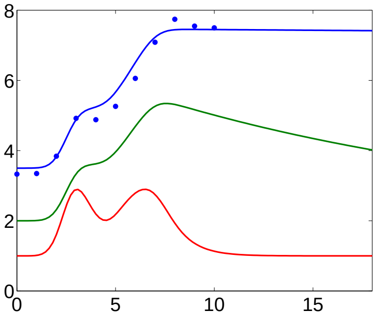
True "gene profiles" and noisy observations.



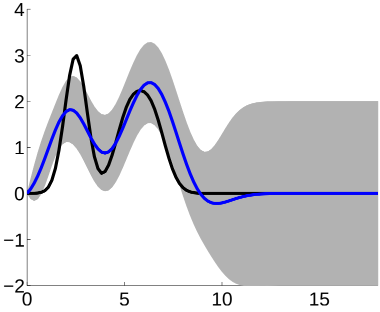
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



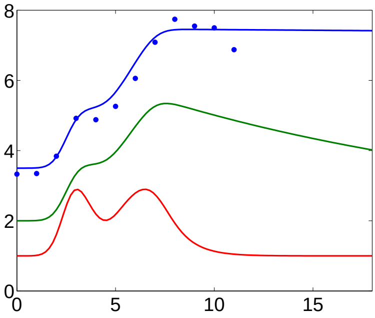
True "gene profiles" and noisy observations.



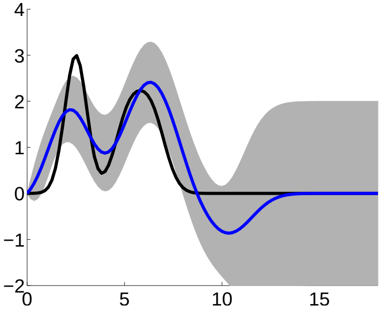
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



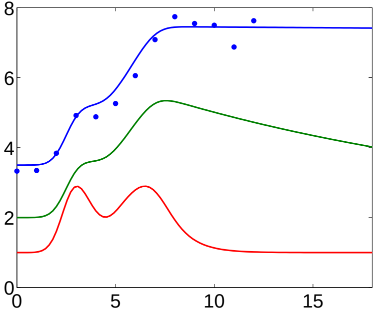
True "gene profiles" and noisy observations.



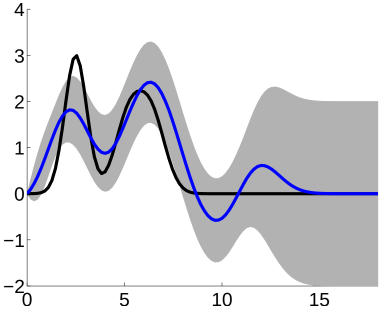
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



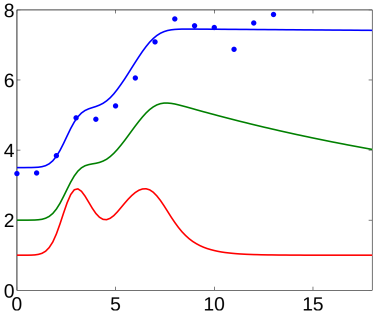
True "gene profiles" and noisy observations.



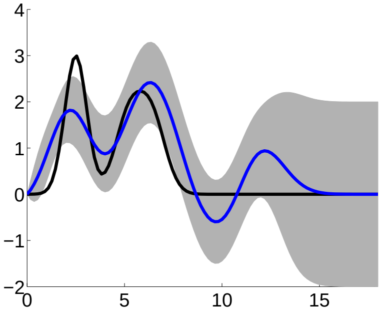
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



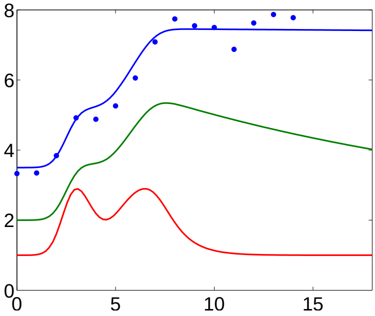
True "gene profiles" and noisy observations.



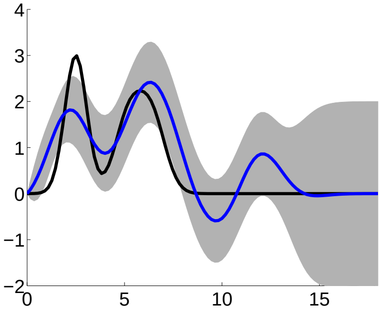
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



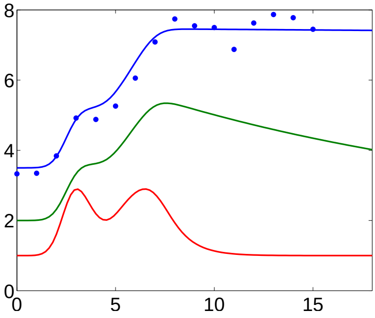
True "gene profiles" and noisy observations.



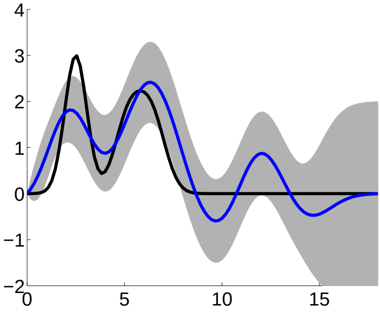
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



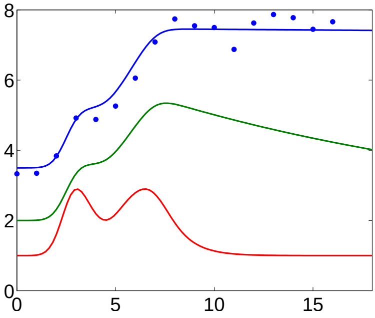
True “gene profiles” and noisy observations.



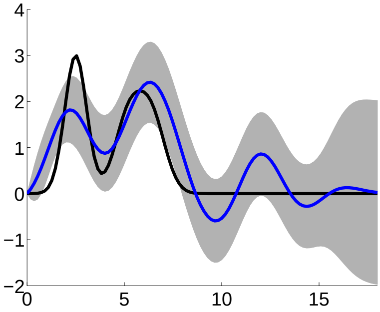
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



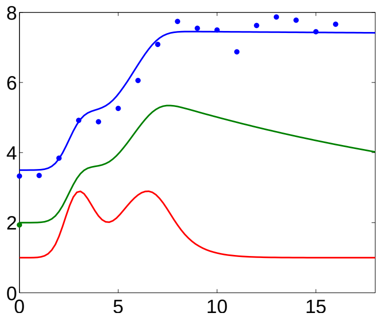
True "gene profiles" and noisy observations.



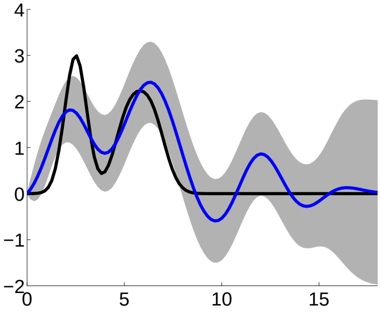
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



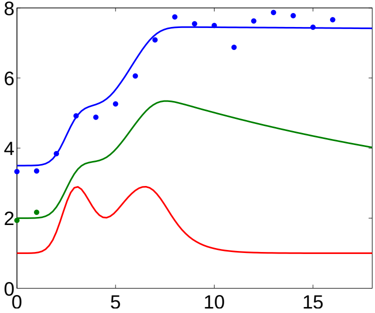
True "gene profiles" and noisy observations.



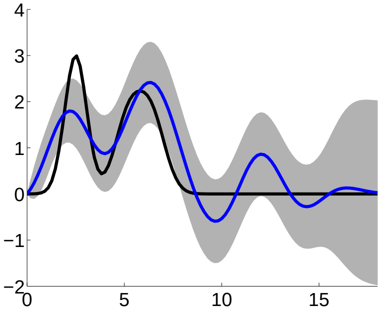
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



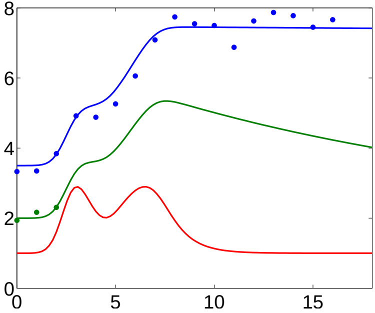
True "gene profiles" and noisy observations.



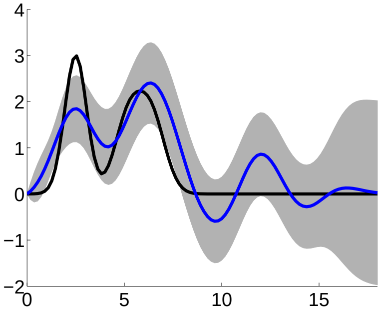
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



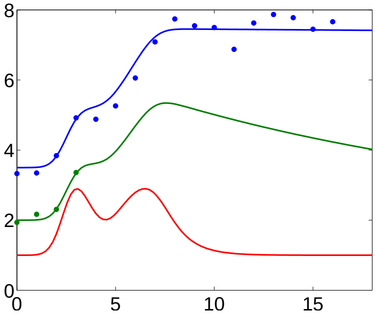
True "gene profiles" and noisy observations.



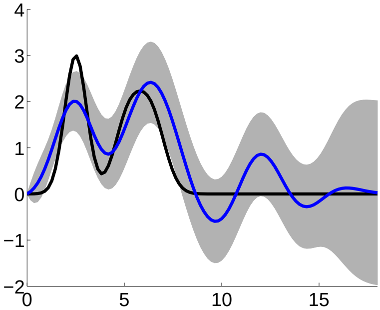
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



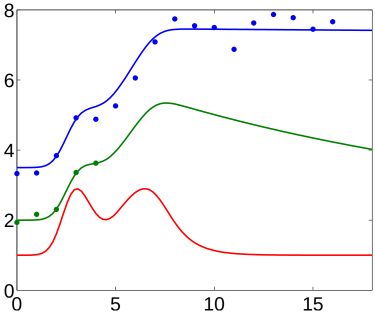
True "gene profiles" and noisy observations.



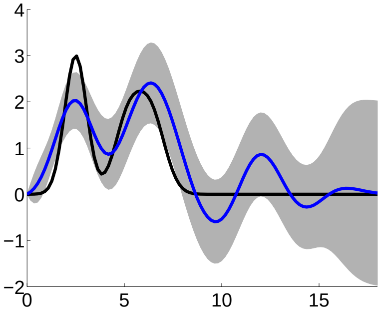
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



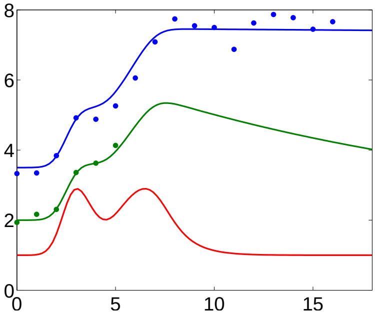
True "gene profiles" and noisy observations.



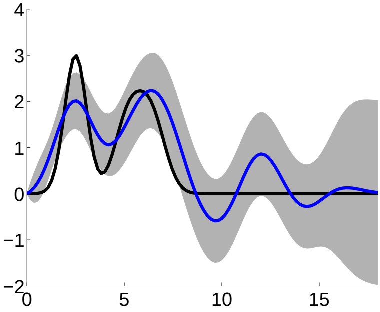
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



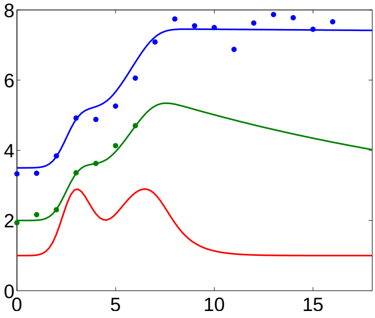
True "gene profiles" and noisy observations.



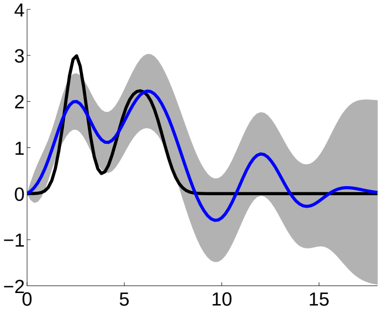
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



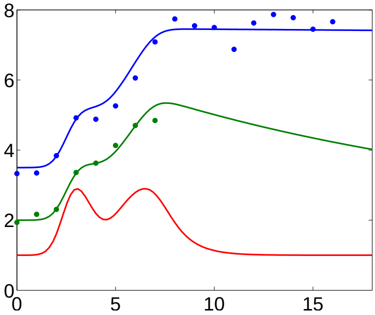
True "gene profiles" and noisy observations.



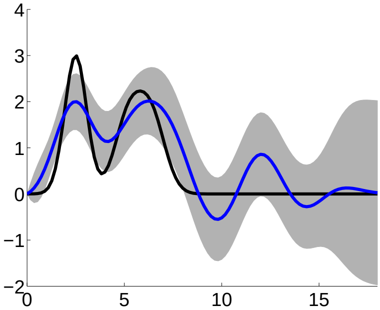
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



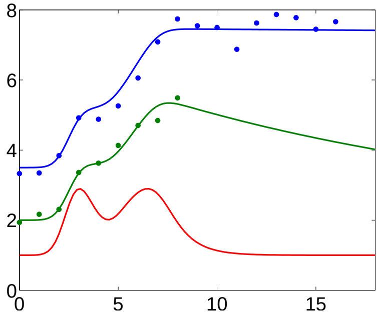
True "gene profiles" and noisy observations.



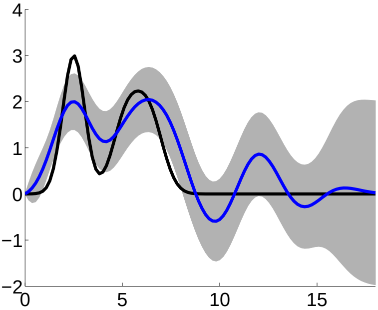
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



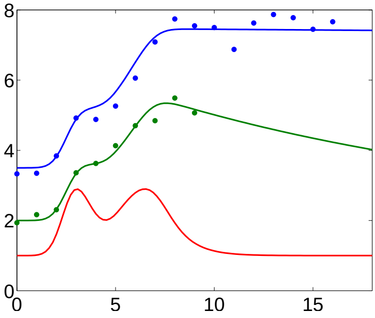
True “gene profiles” and noisy observations.



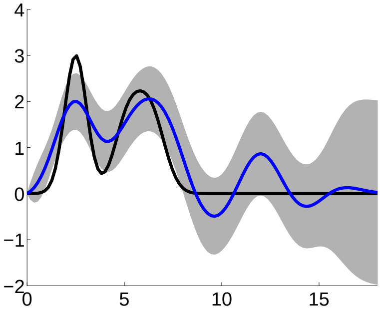
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



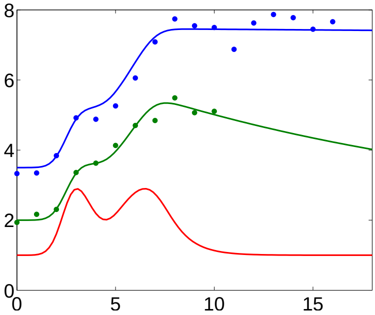
True "gene profiles" and noisy observations.



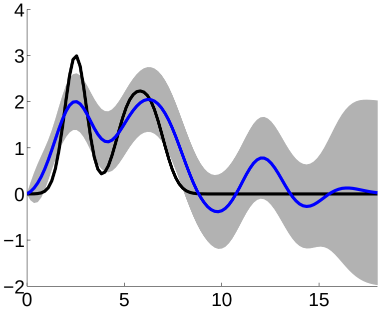
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



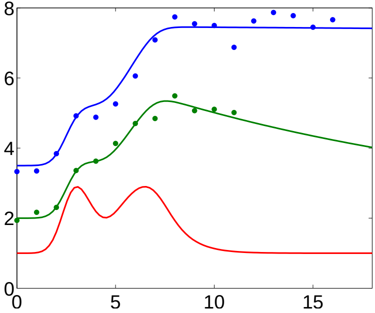
True "gene profiles" and noisy observations.



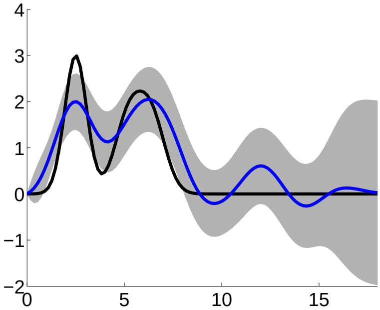
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



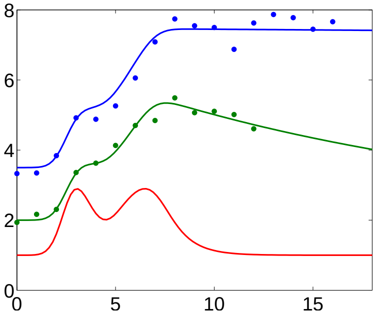
True "gene profiles" and noisy observations.



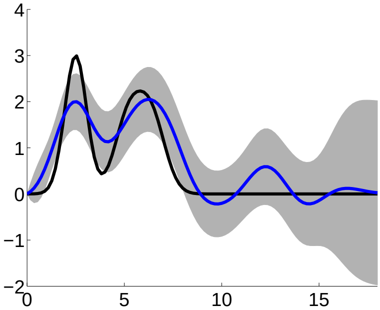
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



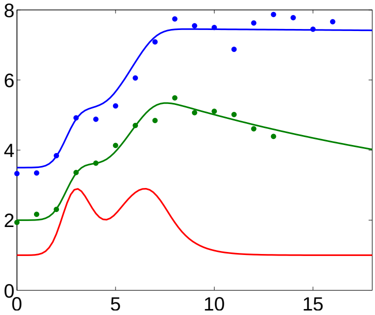
True "gene profiles" and noisy observations.



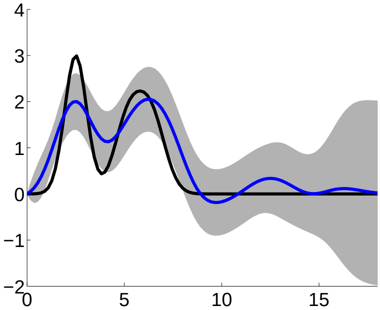
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



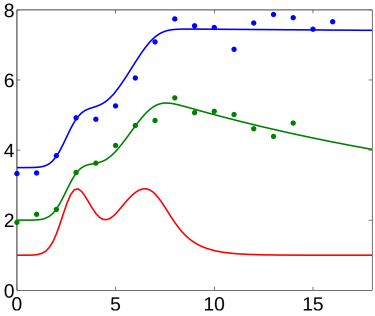
True "gene profiles" and noisy observations.



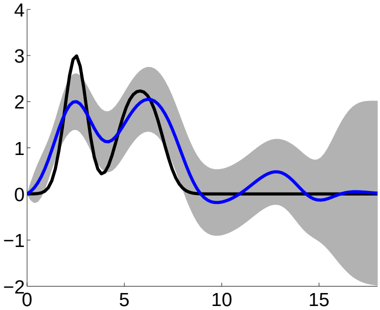
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



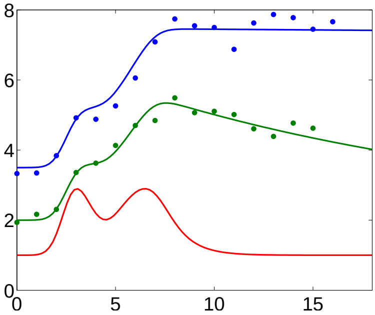
True "gene profiles" and noisy observations.



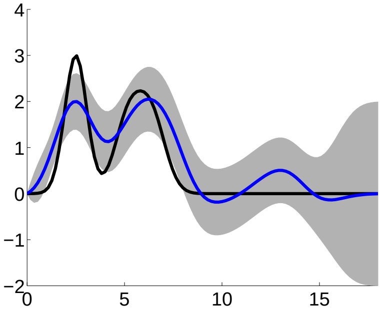
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



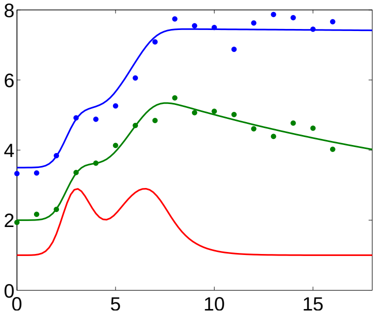
True "gene profiles" and noisy observations.



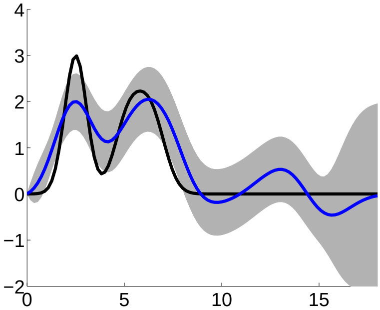
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



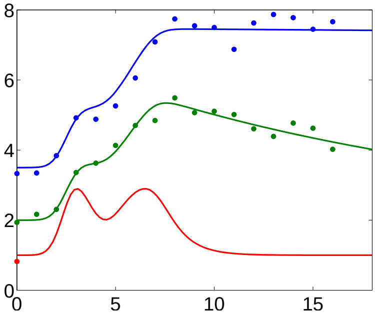
True "gene profiles" and noisy observations.



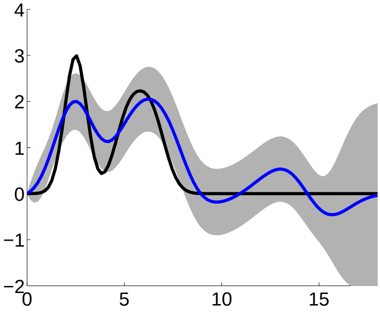
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



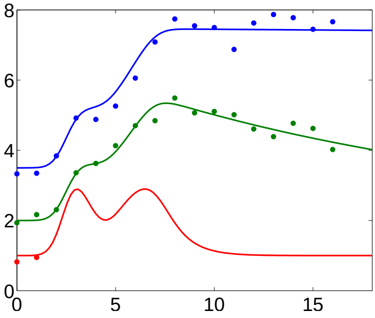
True "gene profiles" and noisy observations.



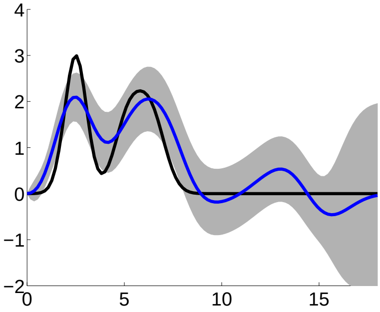
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



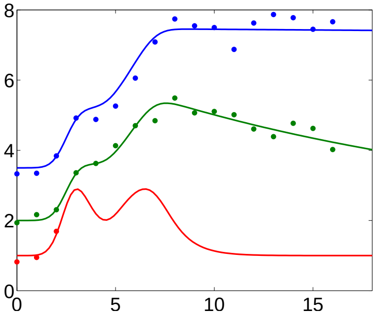
True "gene profiles" and noisy observations.



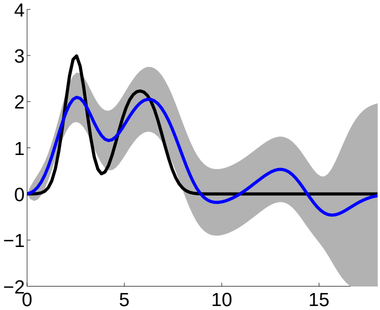
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



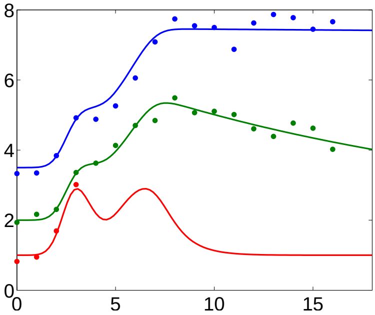
True "gene profiles" and noisy observations.



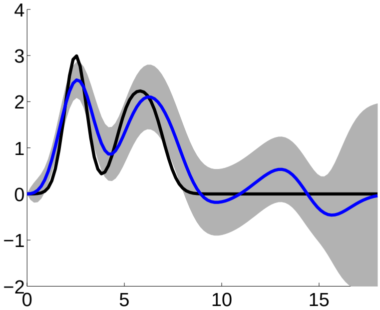
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



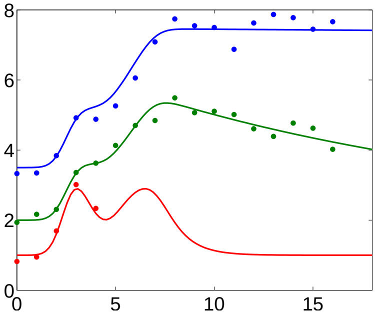
True “gene profiles” and noisy observations.



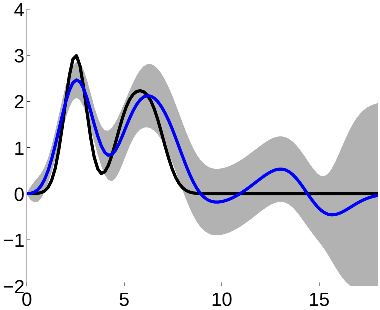
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



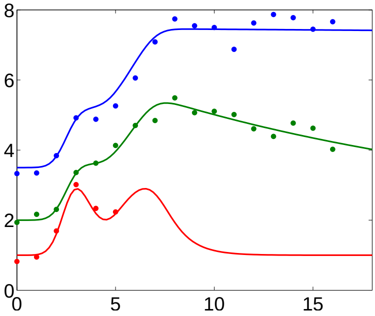
True “gene profiles” and noisy observations.



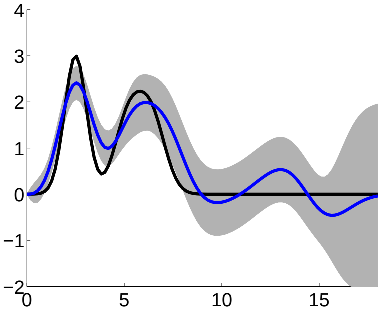
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



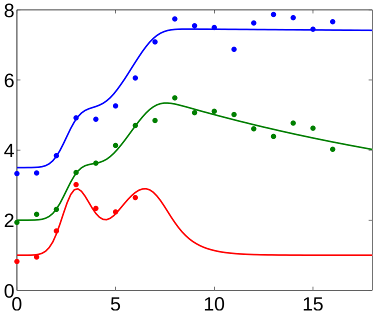
True "gene profiles" and noisy observations.



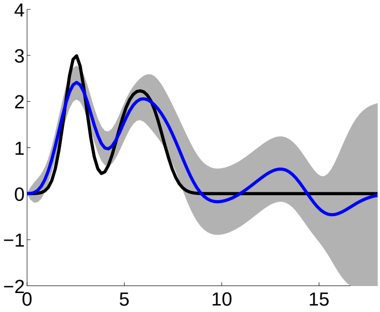
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



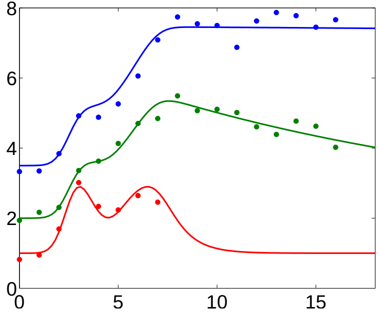
True "gene profiles" and noisy observations.



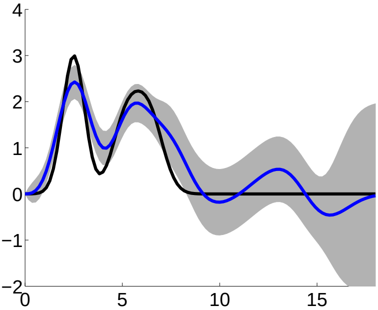
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



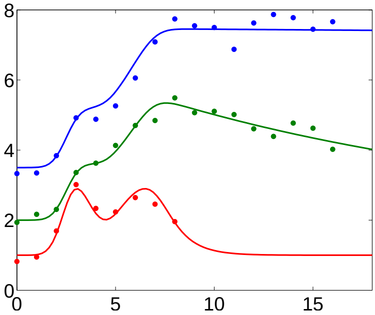
True “gene profiles” and noisy observations.



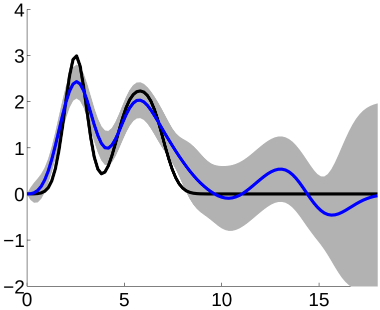
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



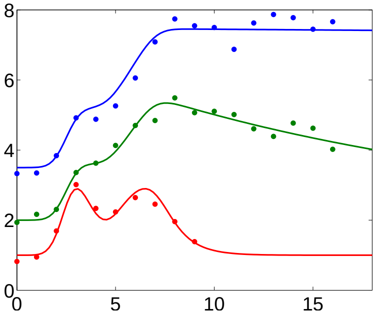
True "gene profiles" and noisy observations.



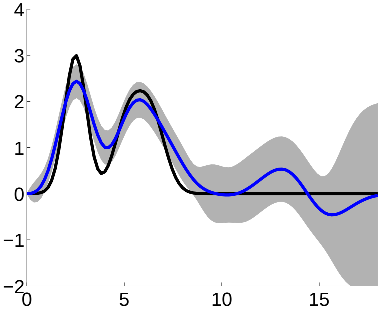
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



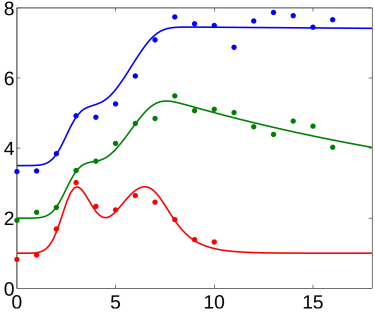
True "gene profiles" and noisy observations.



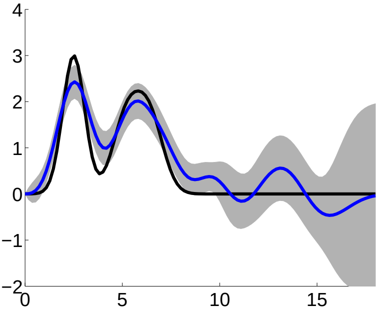
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



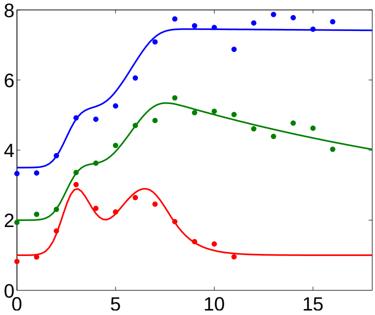
True “gene profiles” and noisy observations.



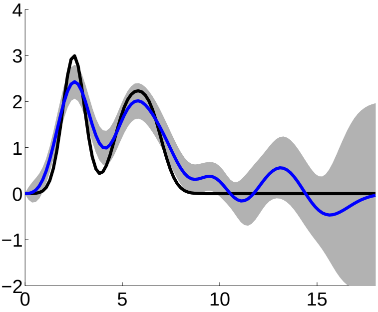
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



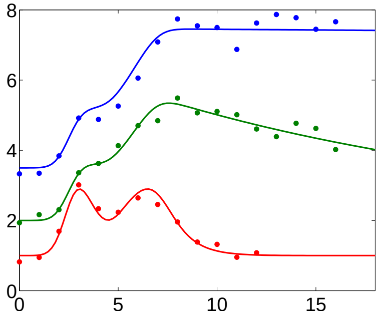
True “gene profiles” and noisy observations.



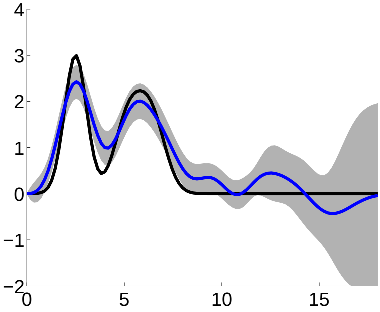
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



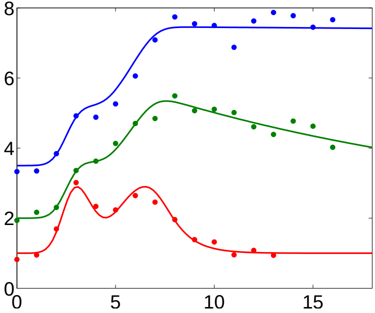
True "gene profiles" and noisy observations.



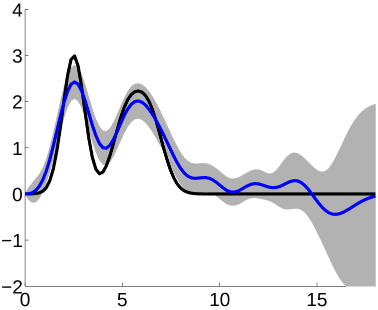
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



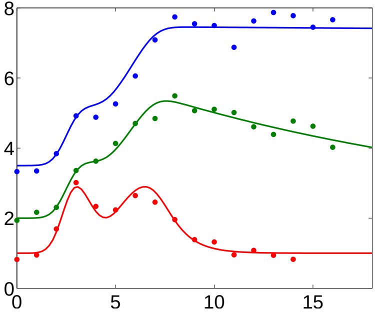
True "gene profiles" and noisy observations.



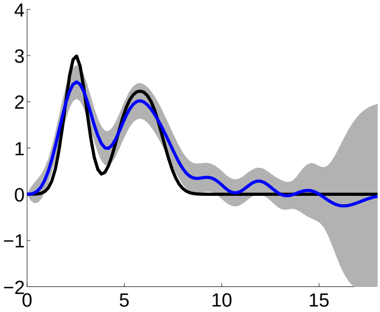
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



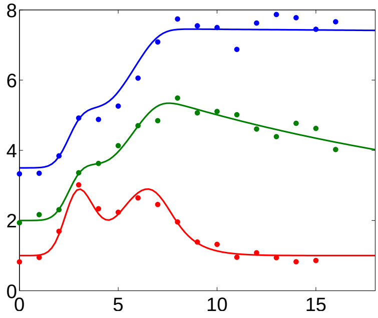
True "gene profiles" and noisy observations.



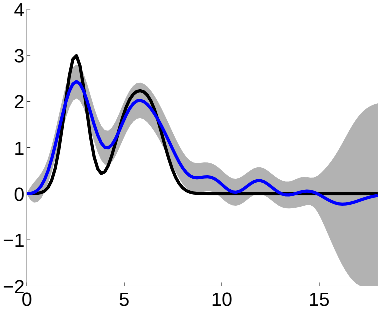
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



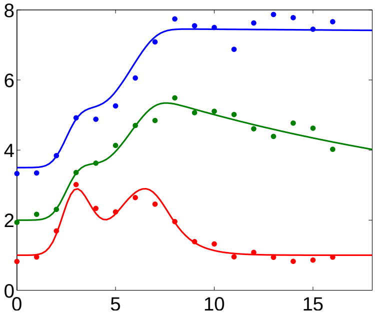
True "gene profiles" and noisy observations.



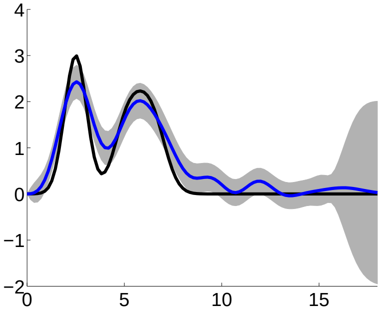
Inferred transcription factor activity.

Artificial Example: Inferring $p(t)$

Inferring TF activity from artificially sampled genes.



True "gene profiles" and noisy observations.



Inferred transcription factor activity.

p53 Results with GP

BIOINFORMATICS

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

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

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

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

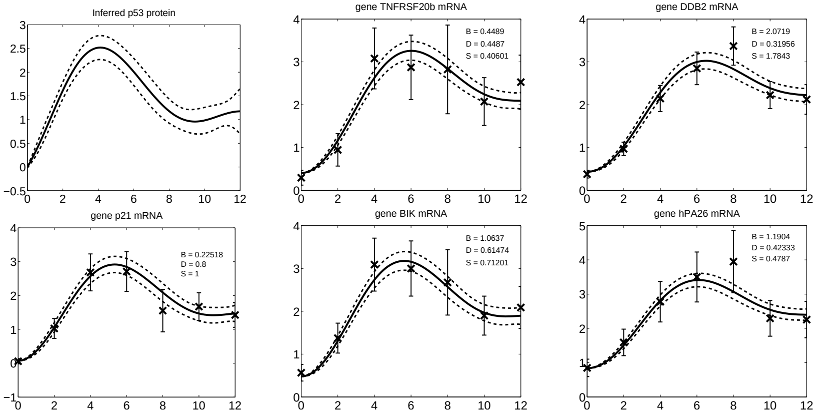
ABSTRACT

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

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

p53 Results with GP

(Gao et al., 2008)

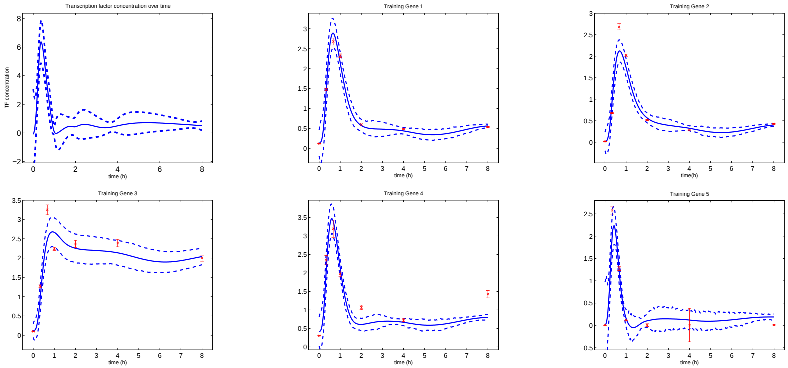


Ranking with ERK Signalling

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

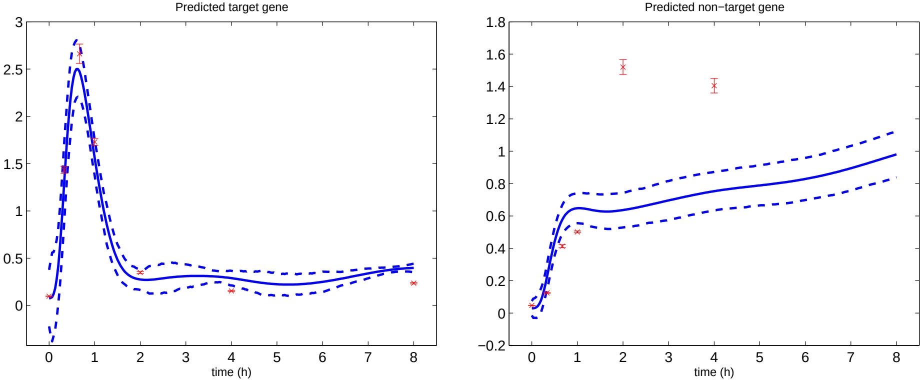
Elk-1 (MLP covariance)

Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



Outline

Cascaded Differential Equations

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

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

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

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

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

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

Cascaded Differential Equations

(Honkela et al., 2010)

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

Drosophila *Mesoderm* Development

Collaboration with Furlong Lab in EMBL Heidelberg.

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

Cascaded Differential Equations

(Honkela et al., 2010)

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

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

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

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

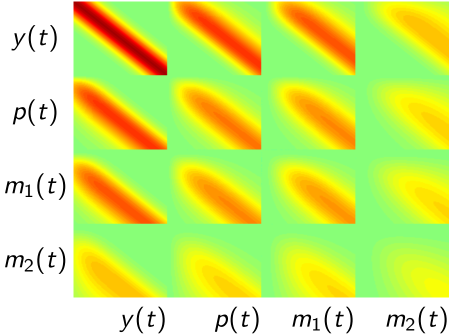
Covariance for Translation/Transcription Model

RBF covariance function for $y(t)$

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

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

δ	d_1	s_1	d_2	s_2
1	5	5	0.5	0.5



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

- `disimSample`

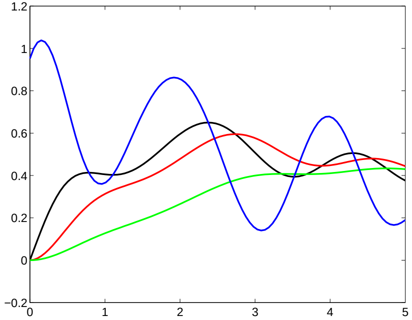


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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

- `disimSample`

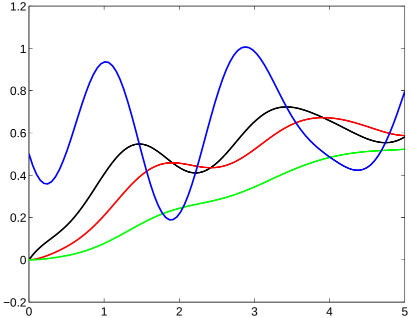


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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

- `disimSample`

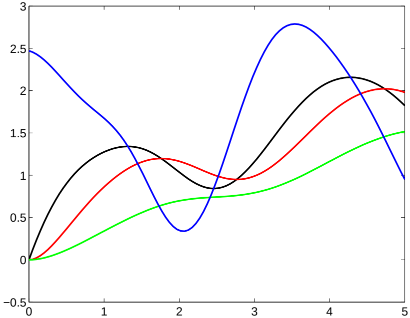


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

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

- `disimSample`

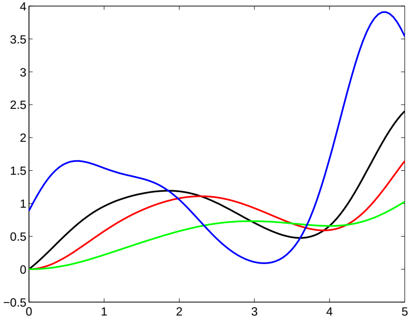


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $p(t)$ (TF concentration), *red*: $m_1(t)$ (high decay target) and *green*: $m_2(t)$ (low decay target)

.

Twist Results

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

Results for Twi using the Cascade model

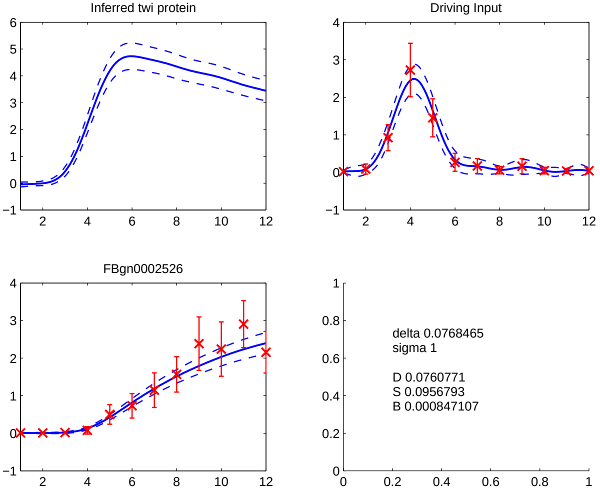


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

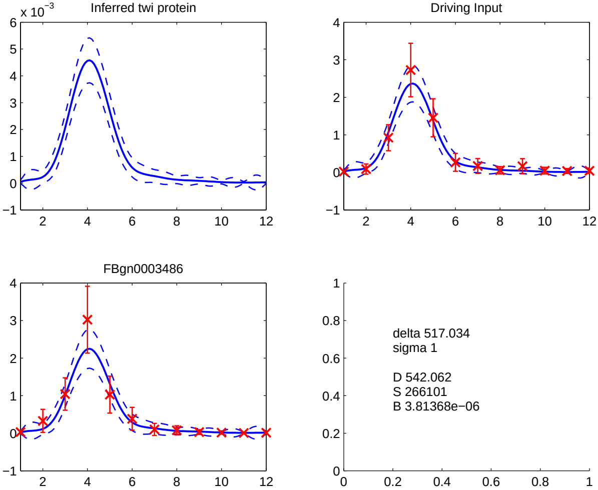


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

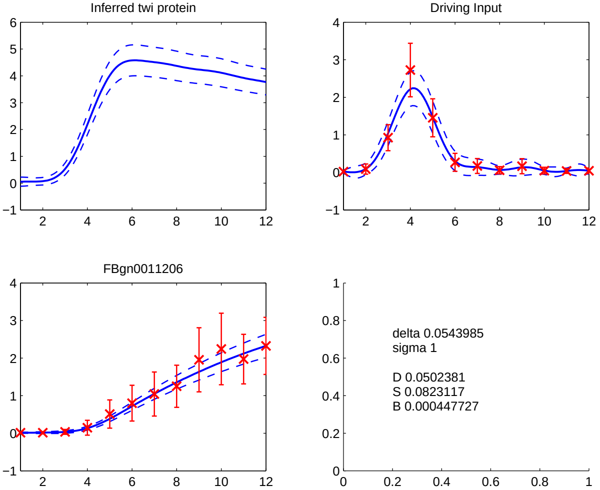


Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

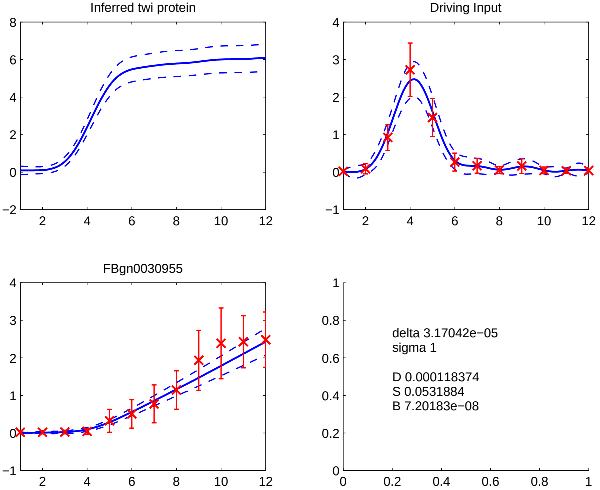


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

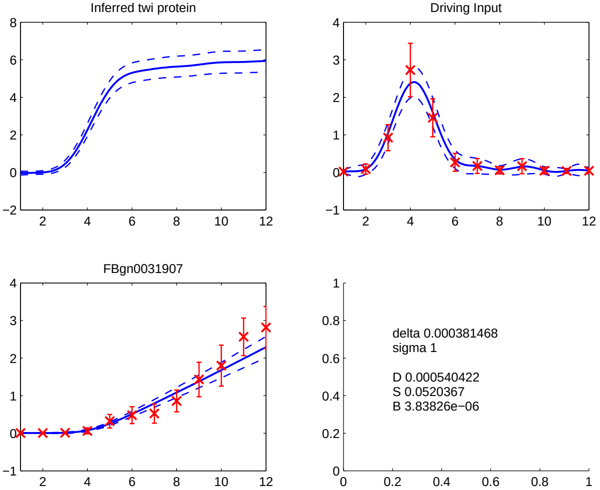


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

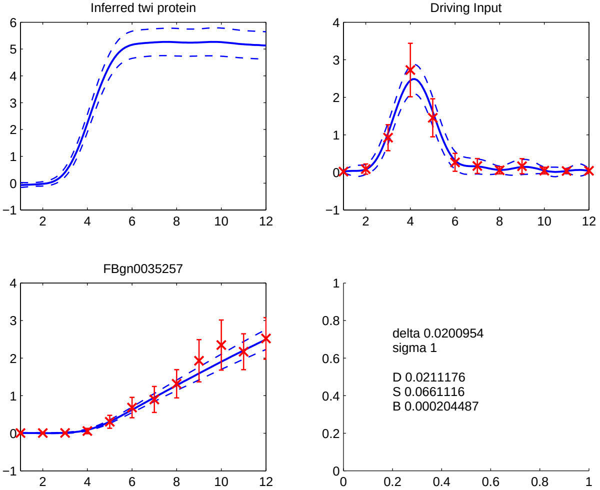


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

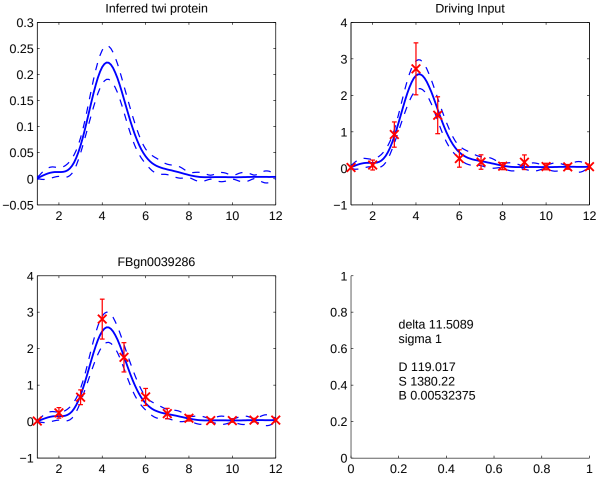
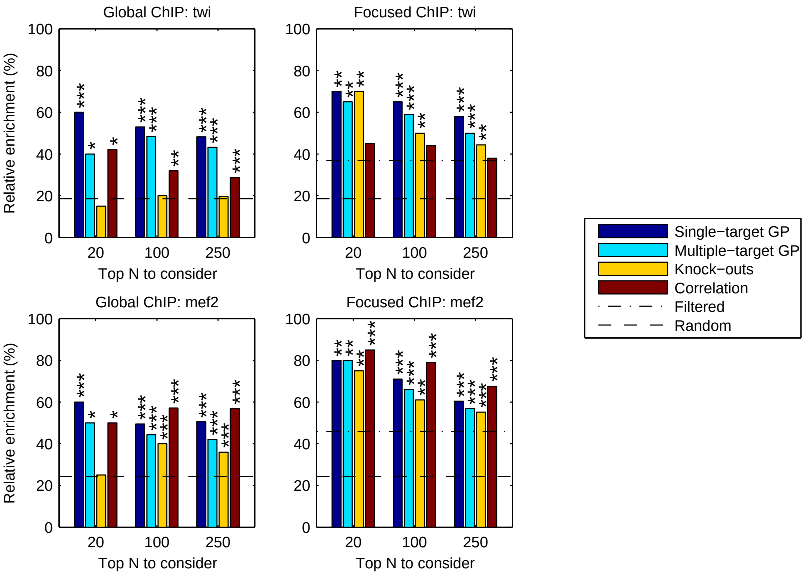


Figure: Model for flybase gene identity FBgn0039286.

Evaluation methods

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

Results



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

Summary

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

Outline

Discussion and Future Work

- Integration of probabilistic inference and mechanistic models for ranking by likelihood.
- Applications in modeling gene expression.
- Cascade model introduces model of translation.
- Challenges:
  - Non linear response and non linear differential equations.
  - Scaling up to larger systems.
  - Stochastic differential equations.

Acknowledgements

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

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

tigre — Transcription factor Inference through Gaussian process Reconstruction of Expression

Now available in Bioconductor and the basis of your Lab Session.

References I

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

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

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

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

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

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

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

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