Gaussian Process Modelling of Transcription Factor Networks using Markov Chain Monte Carlo

Michalis K. Titsias, Neil Lawrence and Magnus Rattray
School of Computer Science
University of Manchester

26th March 2008

Outline

- A sampling algorithm for Gaussian Process Models
- Transcription Factor protein inference
- Conclusions

Gaussian Processes

- A Gaussian process (GP) is a distribution over real-valued functions $f(\mathbf{x})$. It is defined by
  - a mean function

$$\mu(\mathbf{x}) = E(f(\mathbf{x}))$$

  - and a covariance or kernel function

$$k(\mathbf{x}_n, \mathbf{x}_m) = E(f(\mathbf{x}_n)f(\mathbf{x}_m))$$

    E.g. this can be the RBF (or squared exponential) kernel

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha \exp\left(-\frac{||\mathbf{x}_n - \mathbf{x}_m||^2}{2\ell^2}\right)$$

  What does it mean a distribution over functions?

Gaussian Processes

- In reality we only need to evaluate a function in a set of inputs $(\mathbf{x}_i)_{i=1}^N$:

$$f_i = f(\mathbf{x}_i)$$

- A Gaussian process reduces to a multivariate Gaussian distribution over $\mathbf{f} = (f_i)_{i=1}^N$

$$p(\mathbf{f}) = N(\mathbf{x}|\mathbf{0}, K) = \frac{1}{(2\pi)^{\frac{N}{2}}|K|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{f}^T K^{-1}\mathbf{f}}{2}\right)$$

  where the covariance K is defined by the kernel function

- $p(\mathbf{f})$ is a conditional distribution (we condition on the inputs $(\mathbf{x}_i)_{i=1}^N$)

Gaussian Processes for Bayesian learning

Many problems involve inference over some unobserved/latent functions

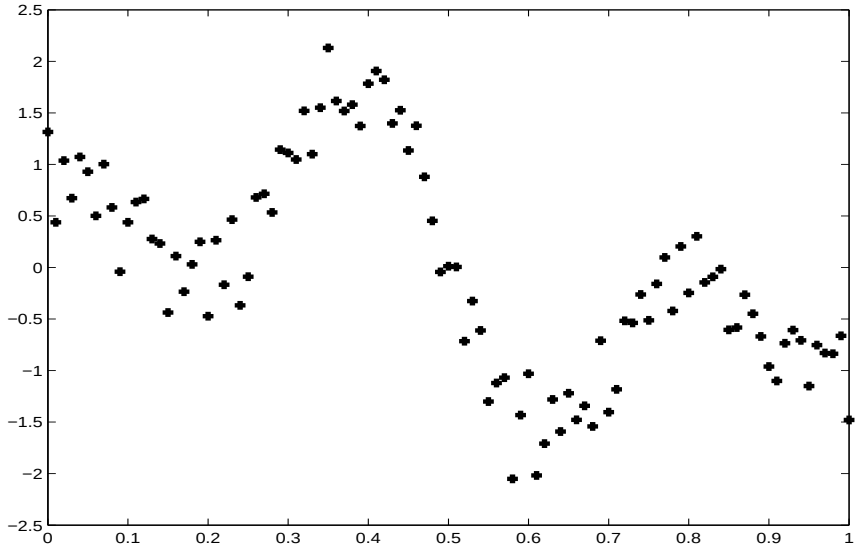
- A Gaussian process can place a prior over functions
- Bayesian inference:
  - Observe data $\mathbf{y} = (y_i)_{i=1}^N$ (associated with inputs $(\mathbf{x}_i)_{i=1}^N$)
  - Assume a likelihood model $p(\mathbf{y}|\mathbf{f})$
  - A GP prior $p(\mathbf{f})$ for the latent function $\mathbf{f}$
  - and apply Bayes rule

$$\begin{aligned} p(\mathbf{f}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{f}) \times p(\mathbf{f}) \\ \text{Posterior} &\propto \text{Likelihood} \times \text{Prior} \end{aligned}$$

- For regression, where the likelihood is Gaussian, this computation is analytically obtained

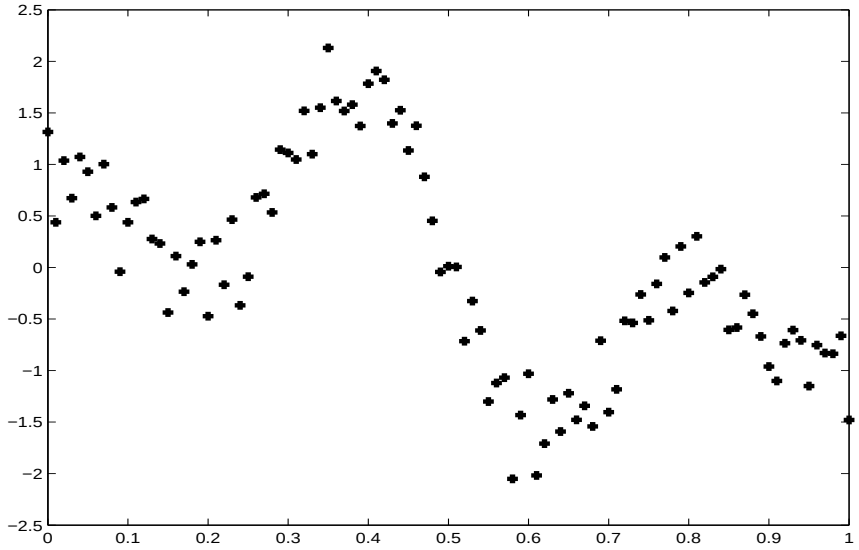
Gaussian Processes for Bayesian Regression

- Data

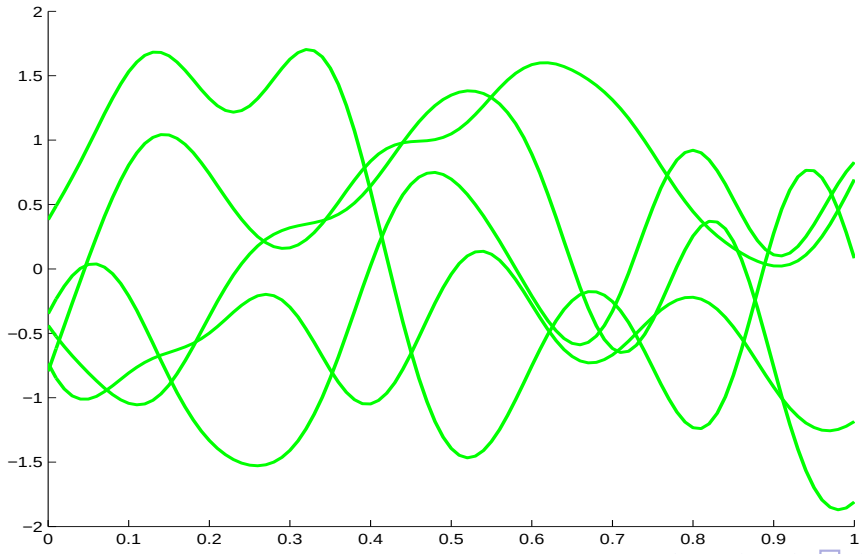


Gaussian Processes for Bayesian Regression

- Data

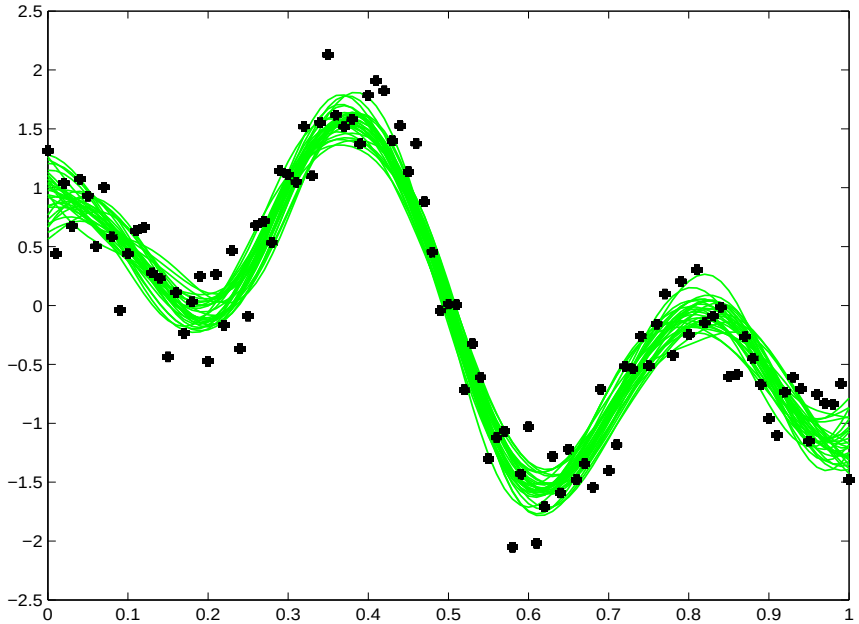


- Gaussian process prior (RBF kernel)



Gaussian Processes for Bayesian Regression

- Posterior



Gaussian Processes for non-Gaussian Likelihoods

- If the likelihood $p(\mathbf{y}|\mathbf{f})$ is non-Gaussian (nonlinear model w.r.t. $\mathbf{f}$ or non Gaussian noise) computations become intractable
- Examples of such likelihoods arise in:
  - Classification problems
  - Non-linear differential equations
- Approximations need to be considered
- MCMC offers a general framework for inference
  - It is applied independently from the form of the likelihood
  - Gives exact inference in the limit of many samples
  - Can be used to validate deterministic approximations

MCMC for Gaussian Processes

The Metropolis-Hastings algorithm

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

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

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

Sampling using control points

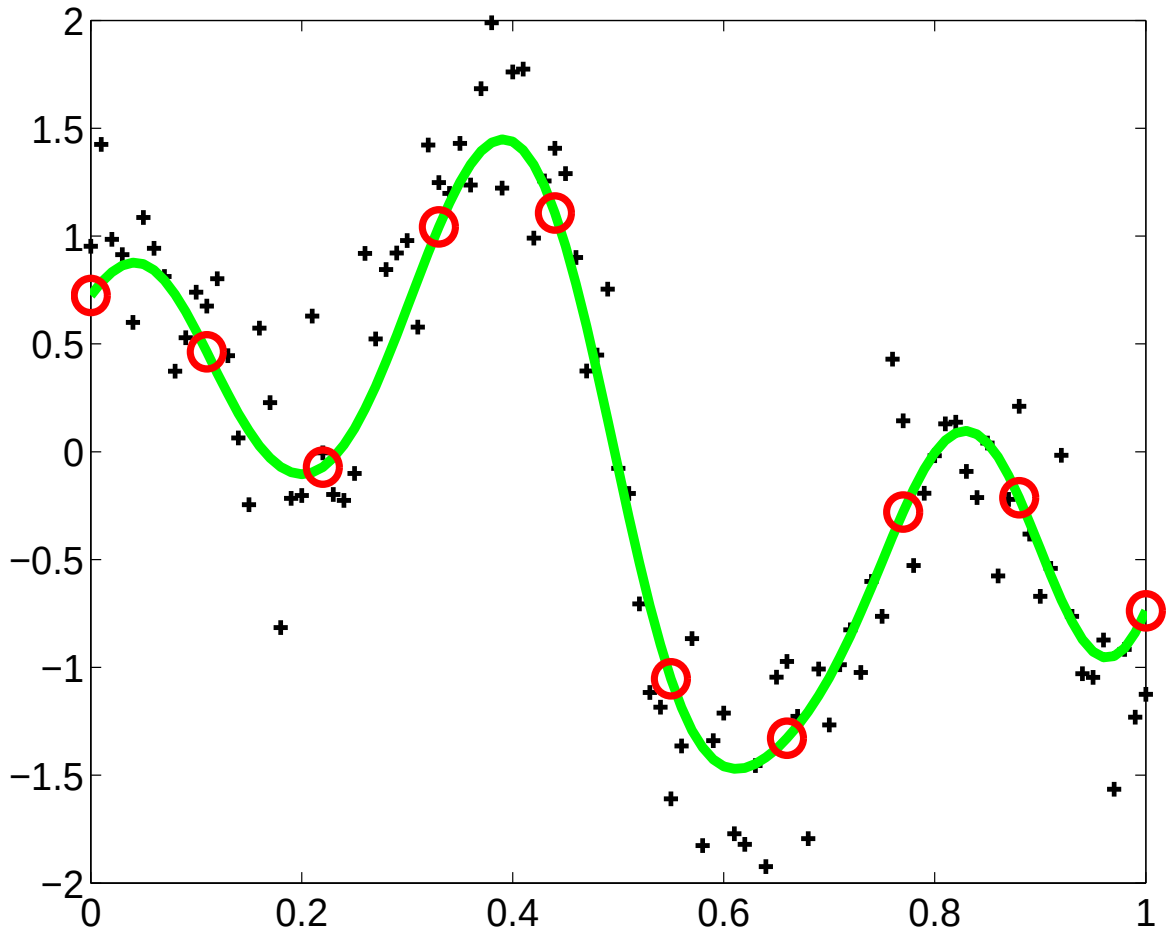
- Separate the points in $\mathbf{f}$ into two groups:
  - few control points $\mathbf{f}_c$
  - and the large majority of the remaining points $\mathbf{f}_\rho = \mathbf{f} \setminus \mathbf{f}_c$
- Sample the control points $\mathbf{f}_c$ using a proposal $q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$
- Sample the remaining points $\mathbf{f}_\rho$ using the conditional GP prior $p(\mathbf{f}_\rho^{(t+1)}|\mathbf{f}_c^{(t+1)})$
- The whole proposal is

$$Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)}) = p(\mathbf{f}_\rho^{(t+1)}|\mathbf{f}_c^{(t+1)})q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$$

- Its like sampling from the prior $p(\mathbf{f})$ but imposing random walk behaviour through the control points

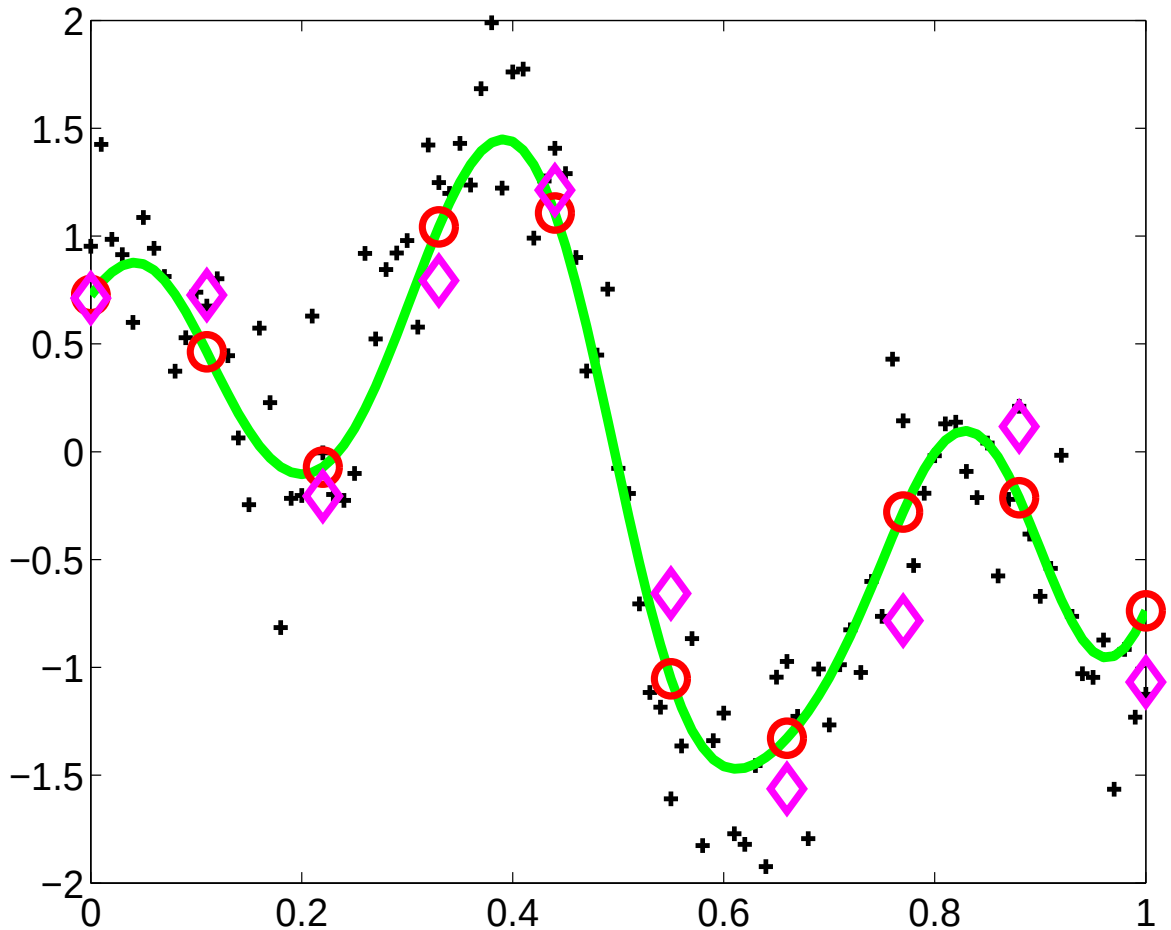
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



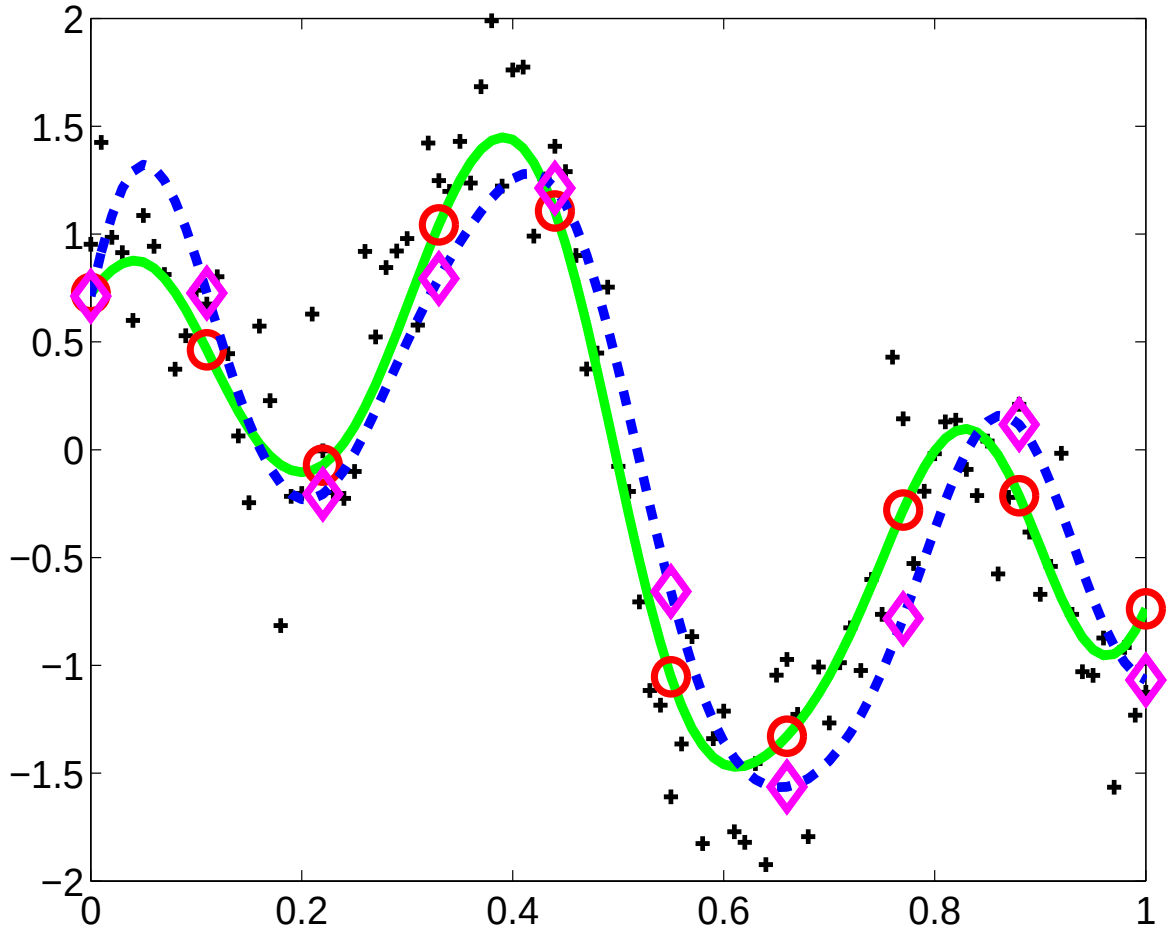
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



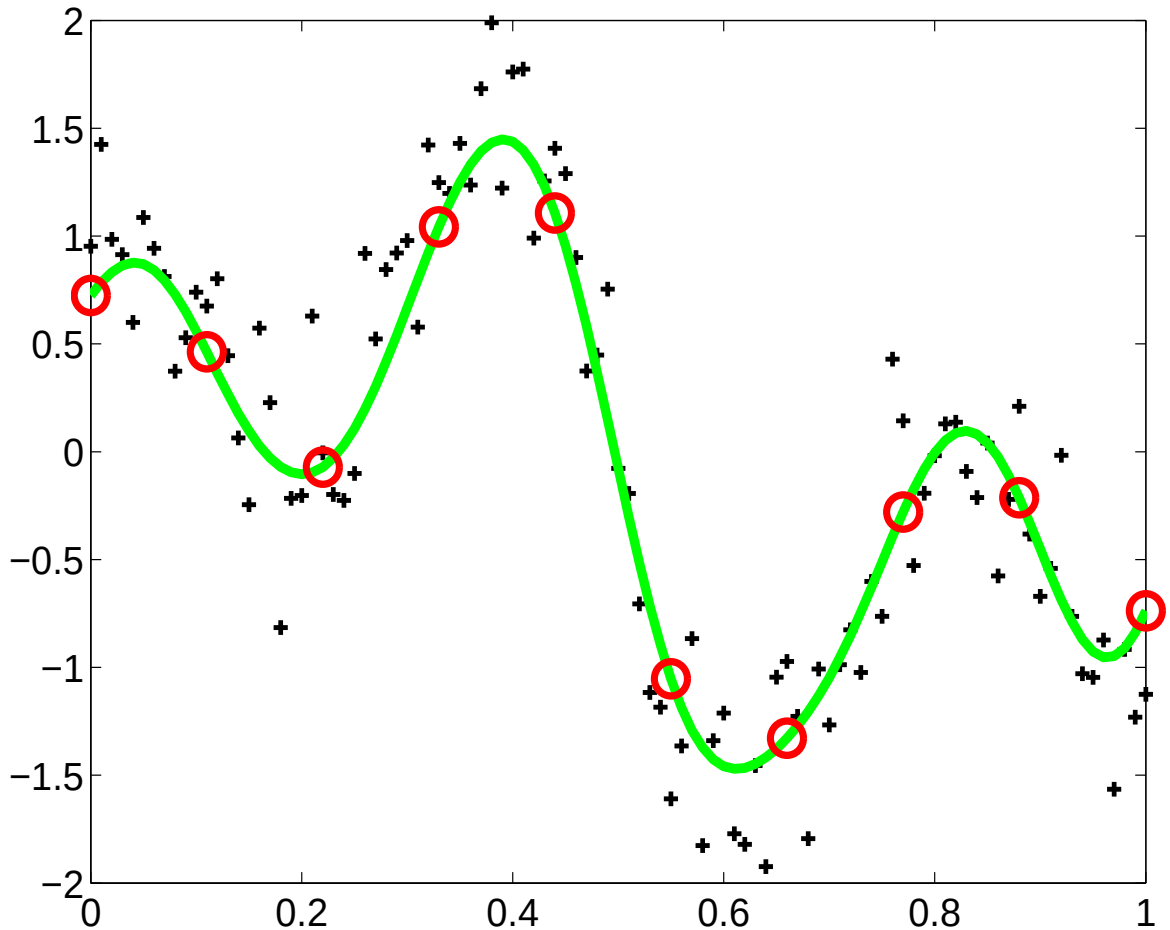
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



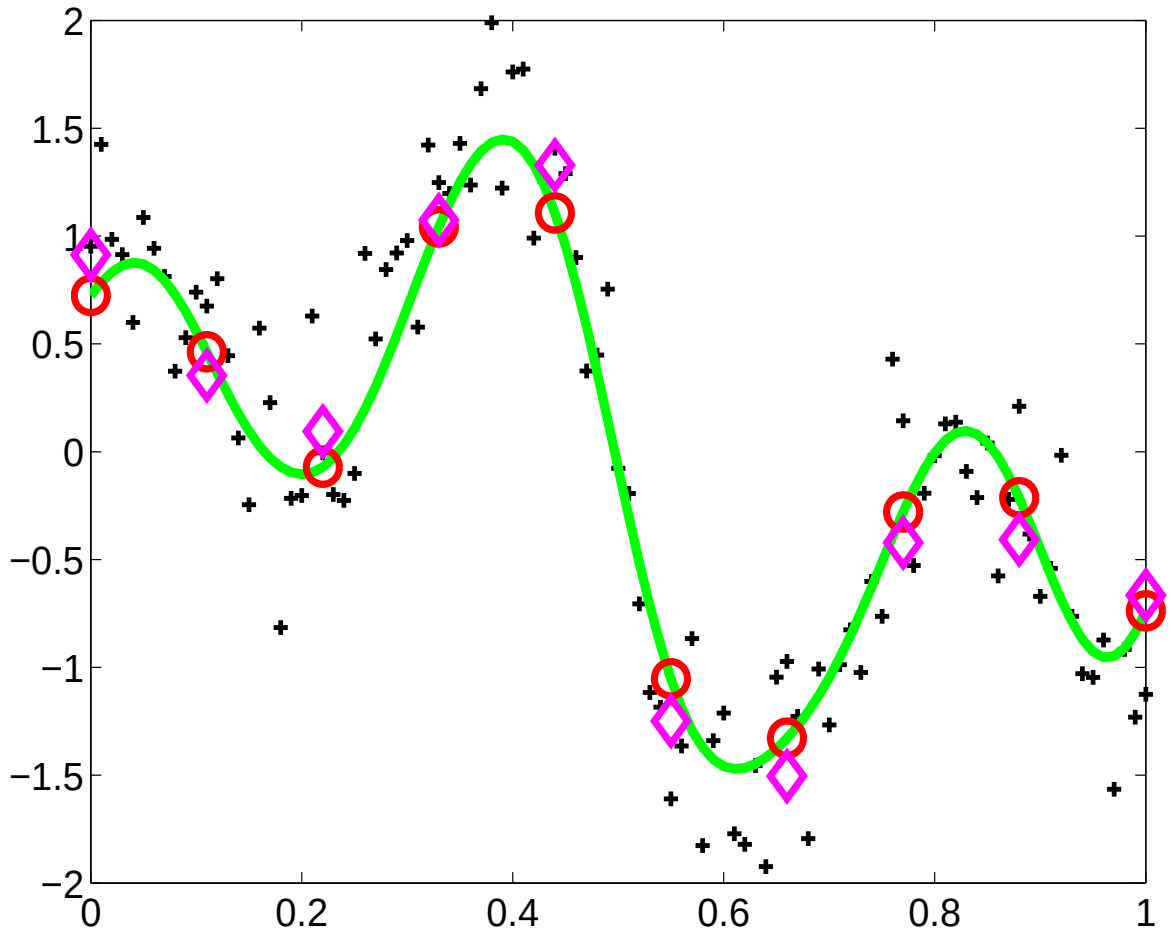
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



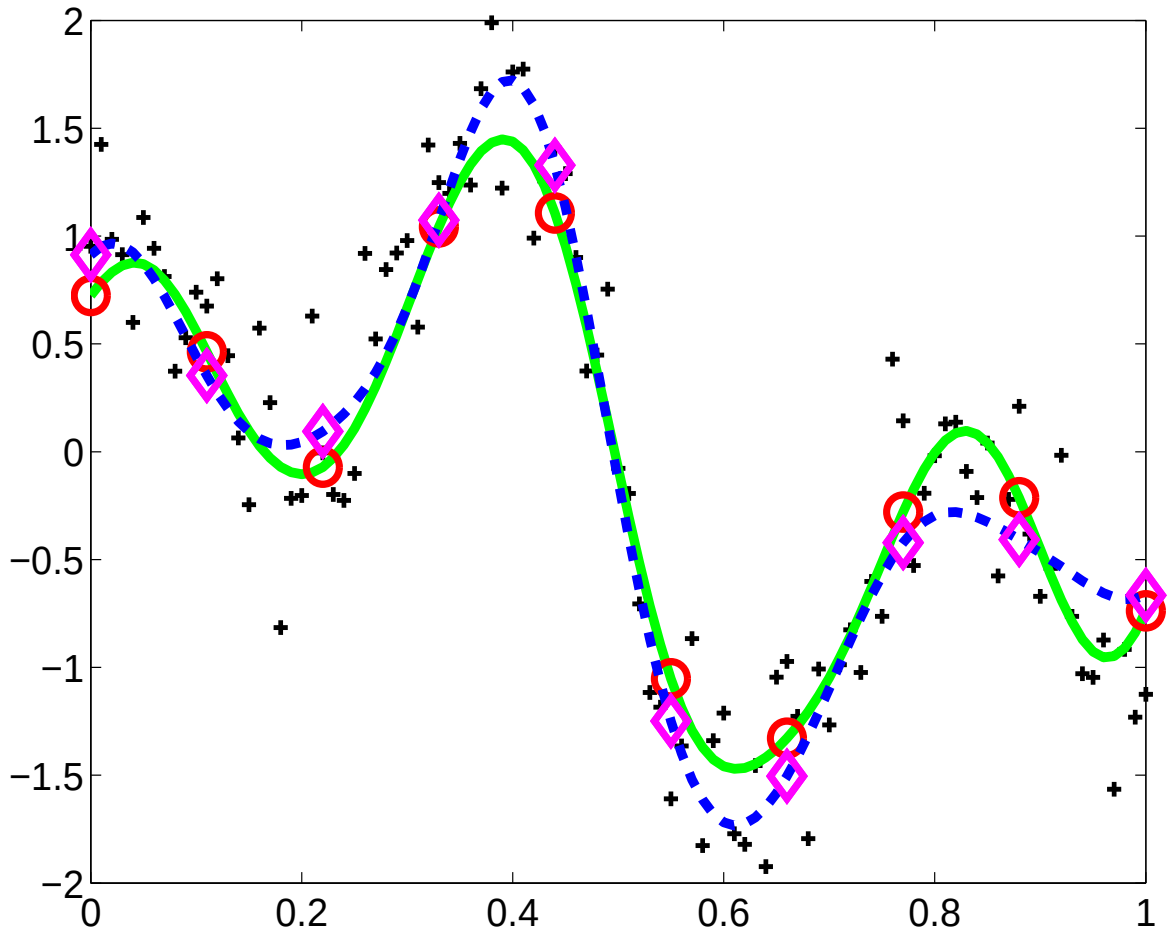
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



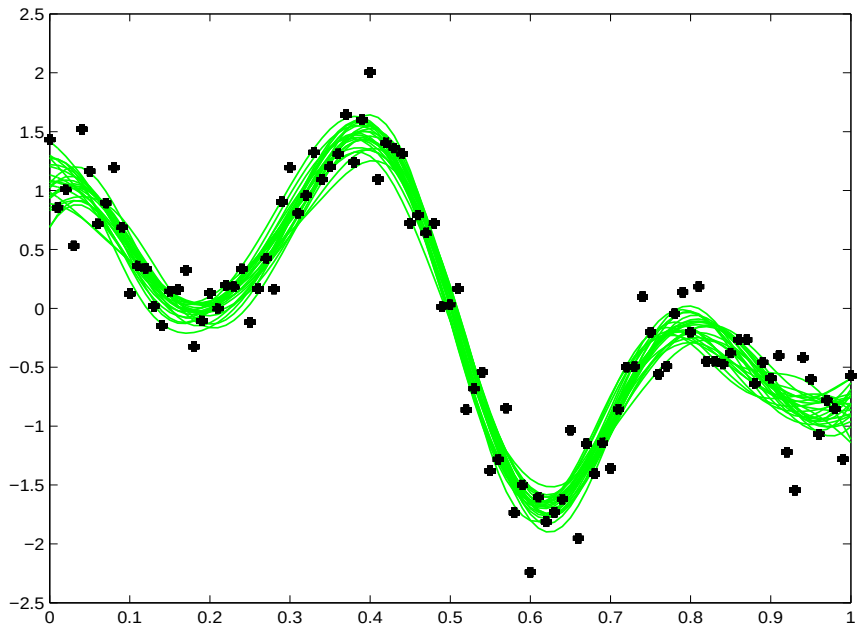
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



Sampling using control points

Few samples drawn during MCMC



Sampling using control points: Adaption of the proposal

Issues that need to be resolved during the burn in MCMC phase

- Number of control points
- Which points should be used as control points
- Improve the acceptance rate by
  - Adapting the variance of $q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$ during the burn in period
  - Sampling the control points in a block-wise manner (keep some of them fixed when you sample others)

For the transcription factor modelling application there are natural choices for all the above issues. In the data we have considered so far we only need to adapt the variances of $q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$

Transcriptional regulation

- Data: Gene expression levels $\mathbf{y} = (y_{jt})$ of N genes at T times
- Goal: We suspect/know that a certain protein regulates (i.e. is a transcription factor (TF)) these genes and we wish to model this relationship
- Model: Use a differential equation (Barenco et al. [2006]; Rogers et. al. [2007])

$$\frac{dy_j(t)}{dt} = B_j + S_j g(f(t)) - D_j y_j(t)$$

- where

  t - time

  $y_j(t)$ - expression of the jth gene

  $f(t)$ - concentration of the transcription factor protein

  D_j - decay rate

  B_j - basal rate

  S_j - Sensitivity

Transcriptional regulation using Gaussian processes

- Solve the equation

$$y_j(t) = \frac{B_j}{D_j} + A_j \exp(-D_j t) + S_j \exp(-D_j t) \int_0^t g(f(u)) \exp(D_j u) du$$

- Apply numerical integration using a very dense grid $(u_i)_{i=1}^P$ and $\mathbf{f} = (f_i(u_i))_{i=1}^P$

$$y_j(t) \simeq \frac{B_j}{D_j} + A_j \exp(-D_j t) + S_j \exp(-D_j t) \sum_{p=1}^{P_t} w_p g(f_p) \exp(D_j u_p)$$

  Assuming Gaussian noise for the observed gene expressions $\{y_{jt}\}$, the ODE defines the likelihood $p(\mathbf{y}|\mathbf{f})$

- Bayesian inference: Assume a GP prior for the transcription factor $\mathbf{f}$ and apply MCMC to infer $(\mathbf{f}, \{A_j, B_j, D_j, S_j\}_{j=1}^N)$
  - $\mathbf{f}$ is inferred in a continuous manner ($P \gg T$)

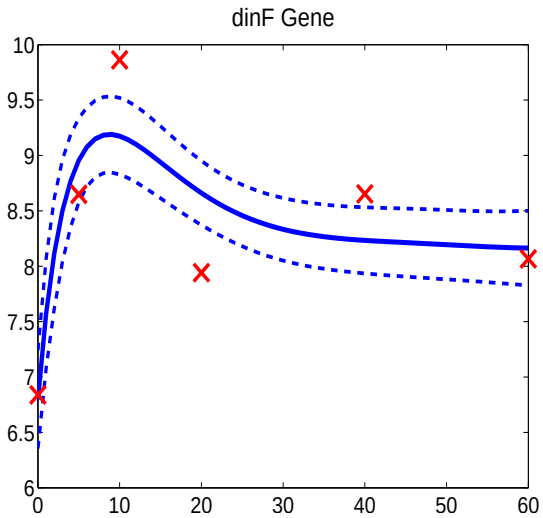
Results in E.coli data: Rogers, Khanin and Girolami (2007)

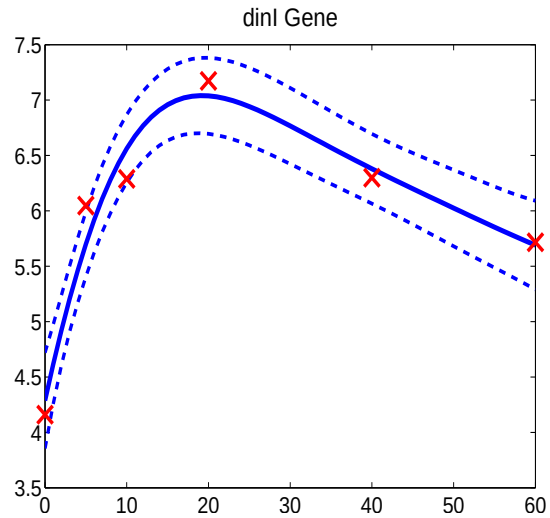
- One transcription factor (lexA) that acts as a repressor. We consider the Michaelis-Menten kinetic equation

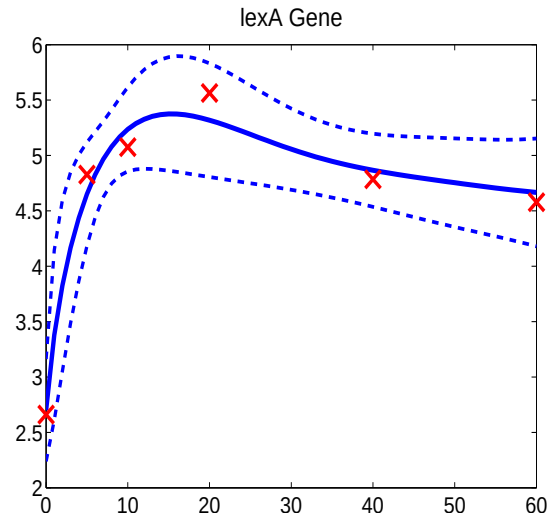
$$\frac{dy_j(t)}{dt} = B_j + S_j \frac{1}{\exp(f(t)) + \gamma_j} - D_j y_j(t)$$

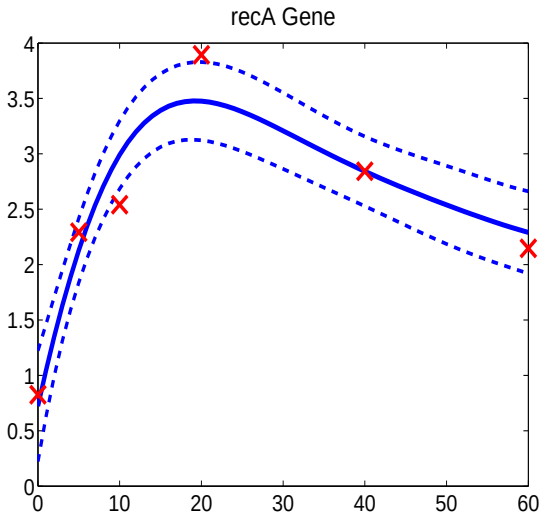
- We have 14 genes (5 kinetic parameters each)
- Gene expressions are available for $T = 6$ time slots
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 6 control points are used (placed in a equally spaced grid)
  - Running time was 5 hours for 2 million sampling iterations plus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

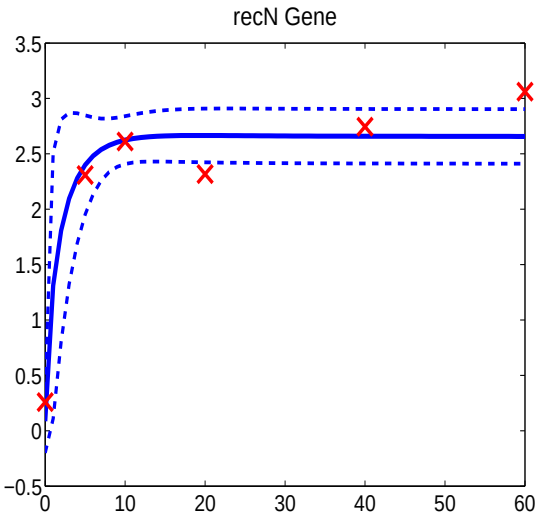
Results in E.coli data: Predicted gene expressions

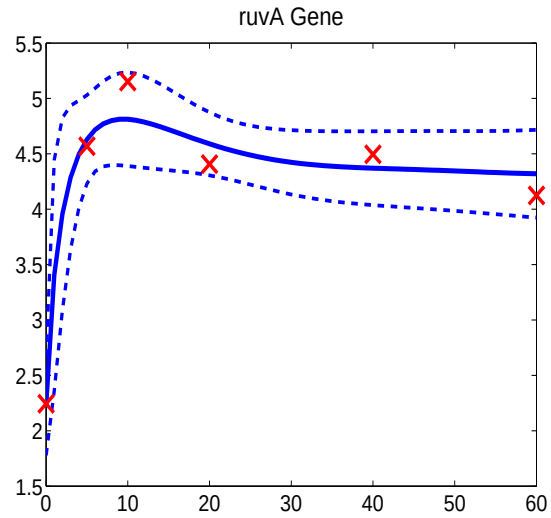




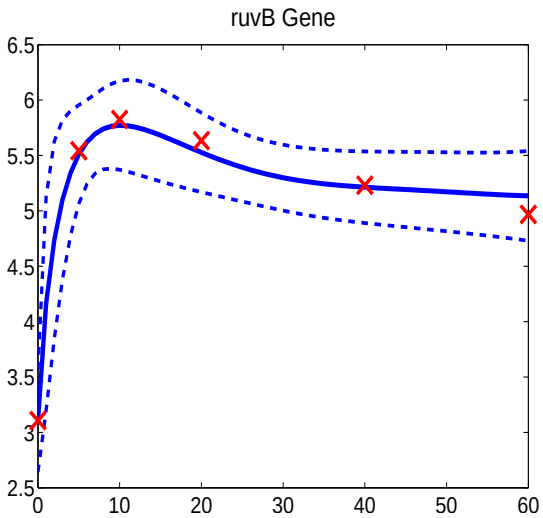


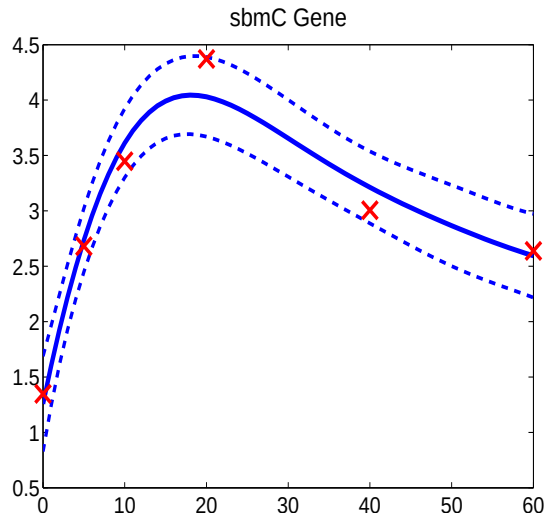


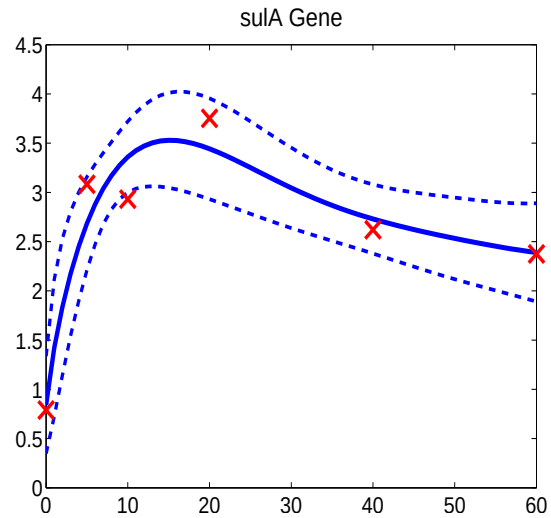


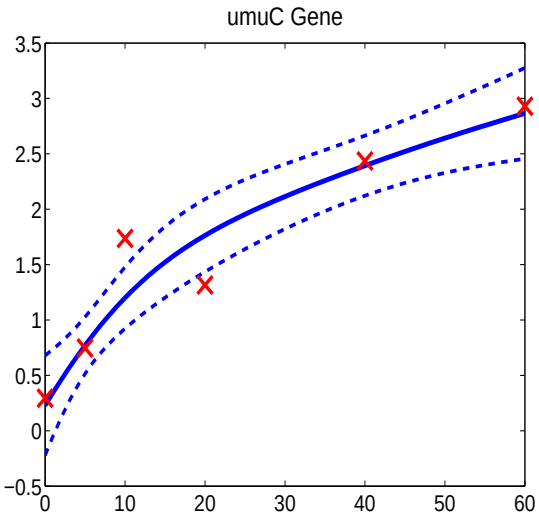


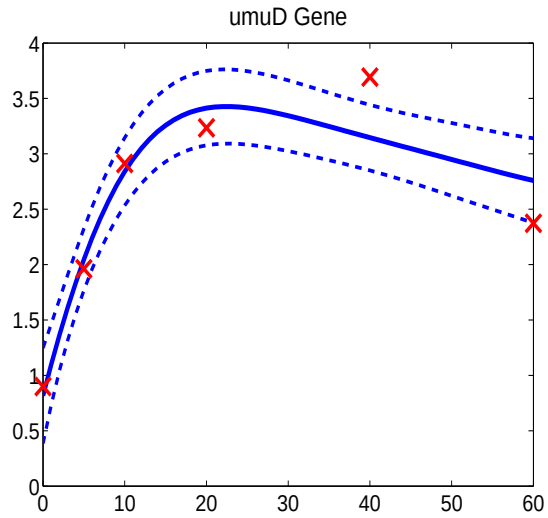
Results in E.coli data: Predicted gene expressions

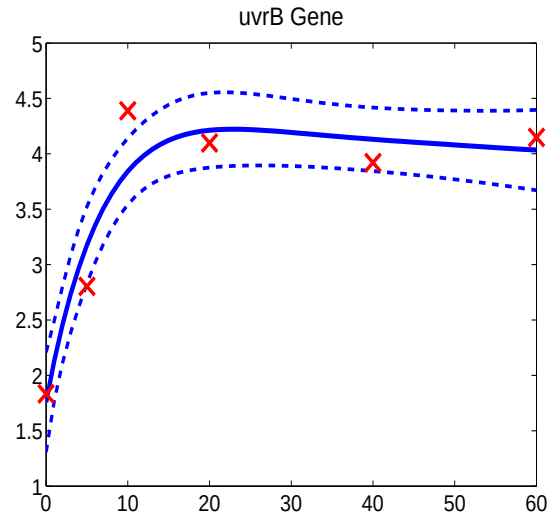




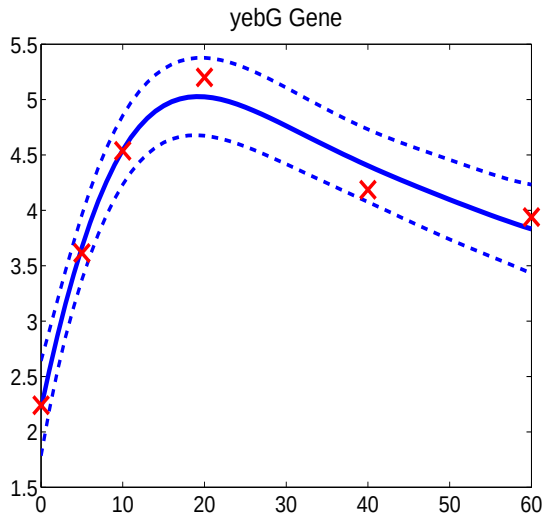


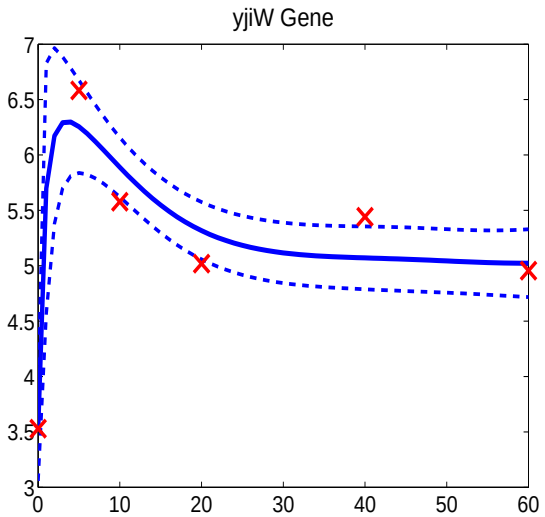




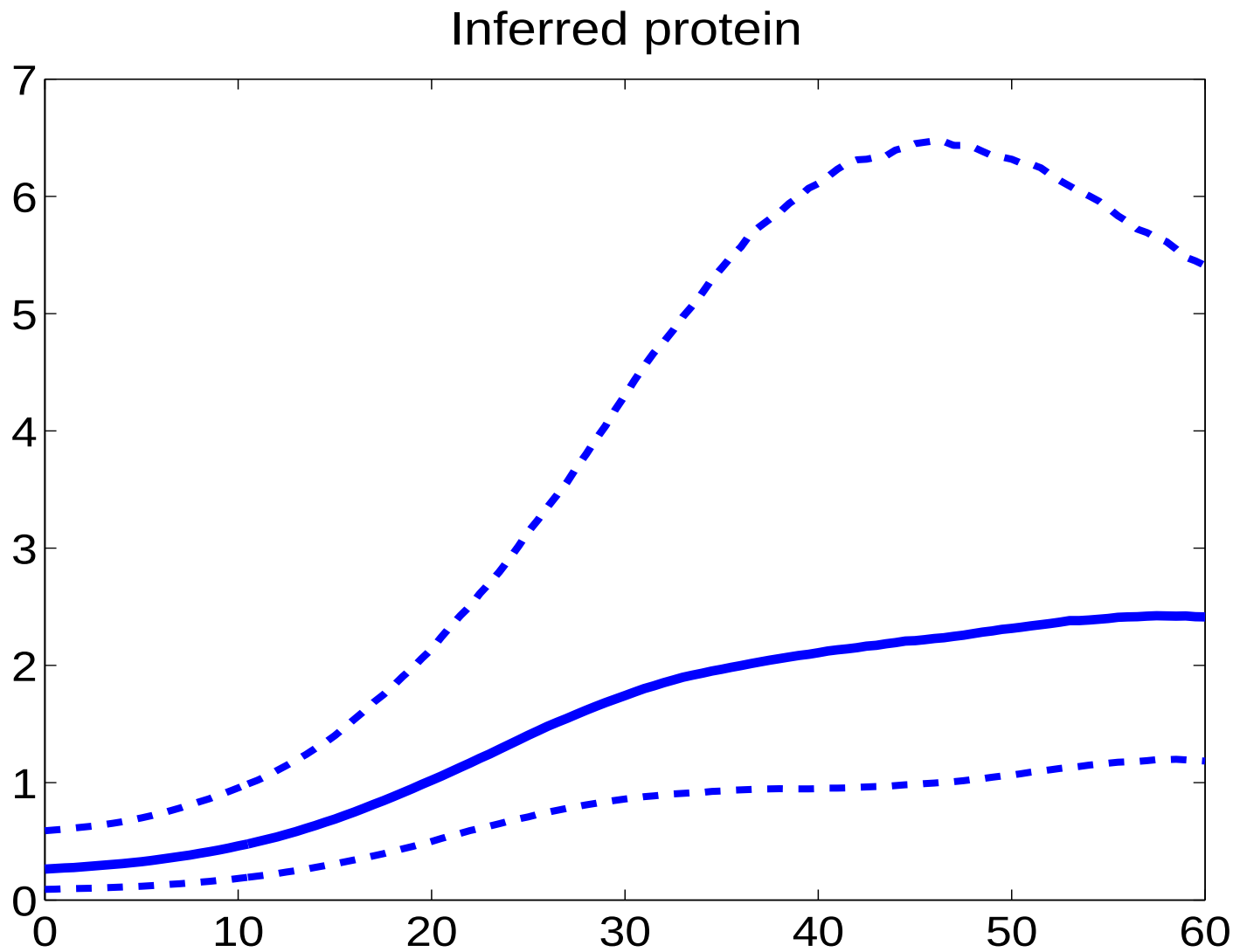


Results in E.coli data: Predicted gene expressions

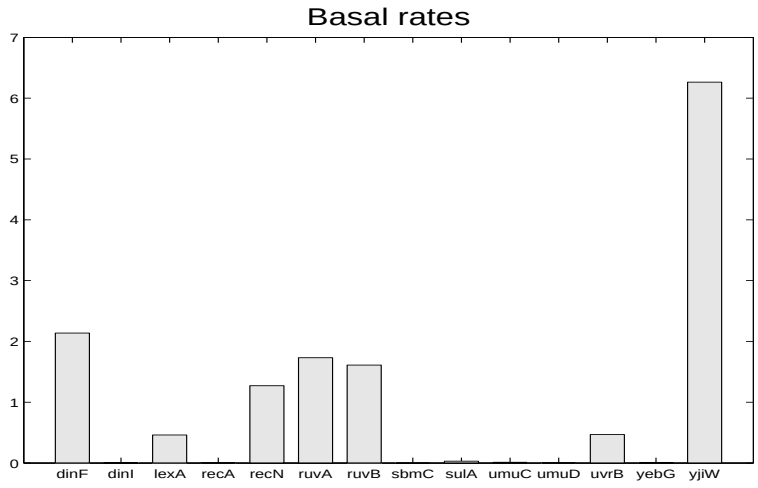


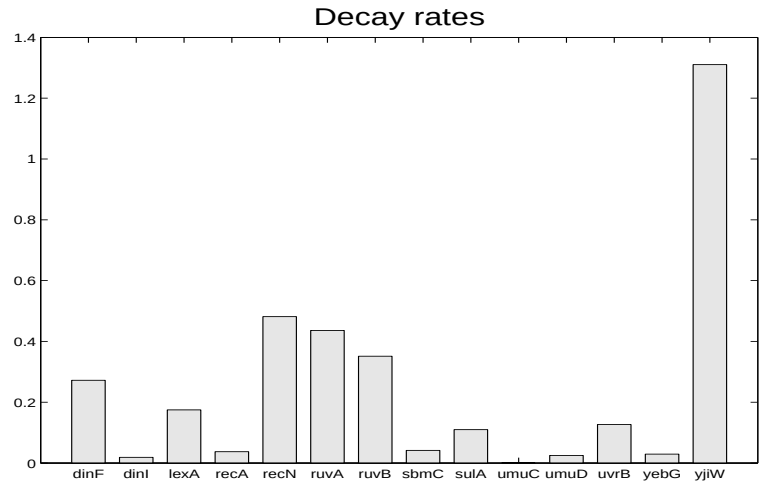


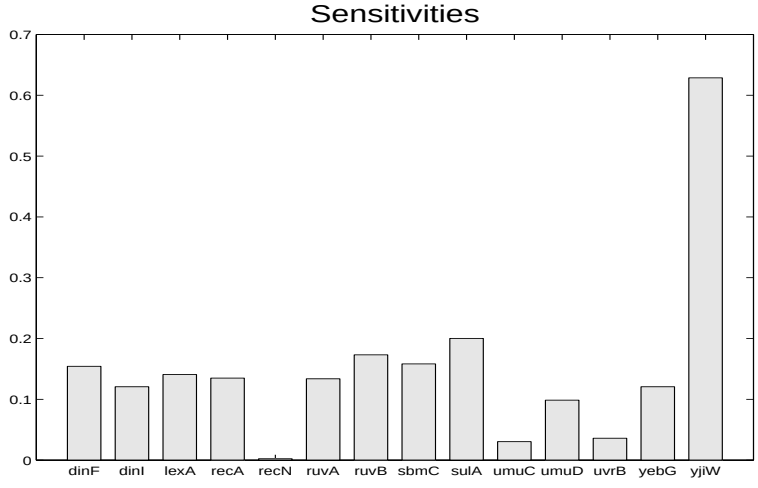
Results in E.coli data: Protein concentration

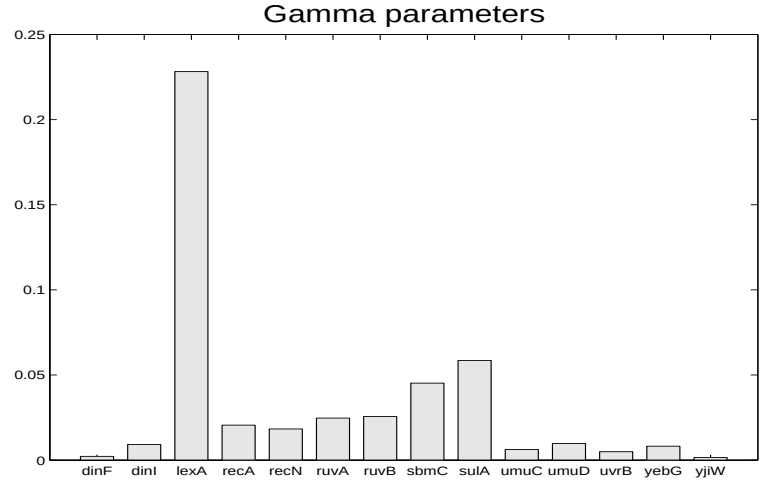


Results in E.coli data: Kinetic parameters

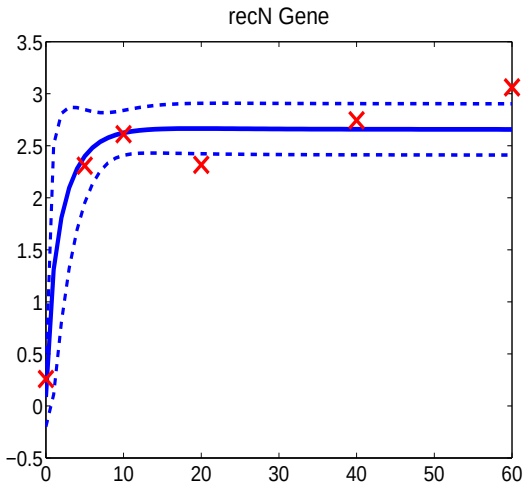


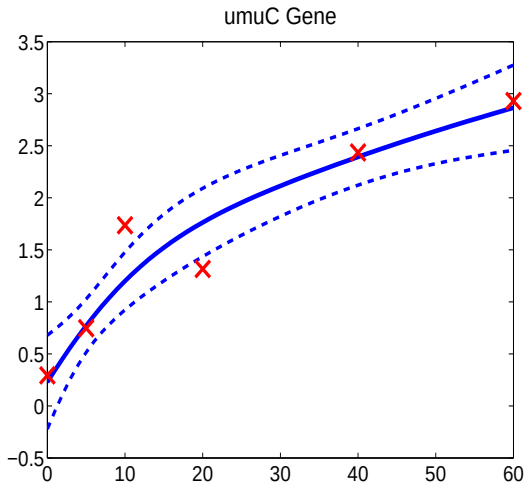


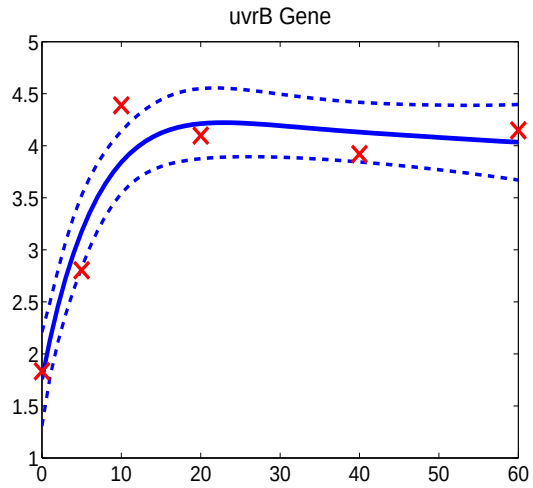


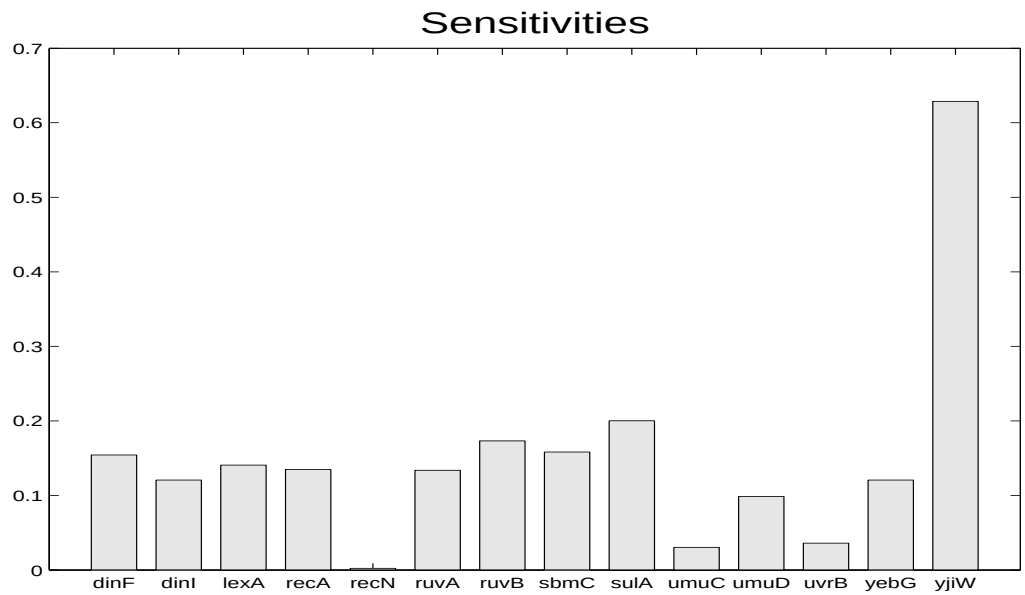


Results in E.coli data: Genes with low sensitivity value

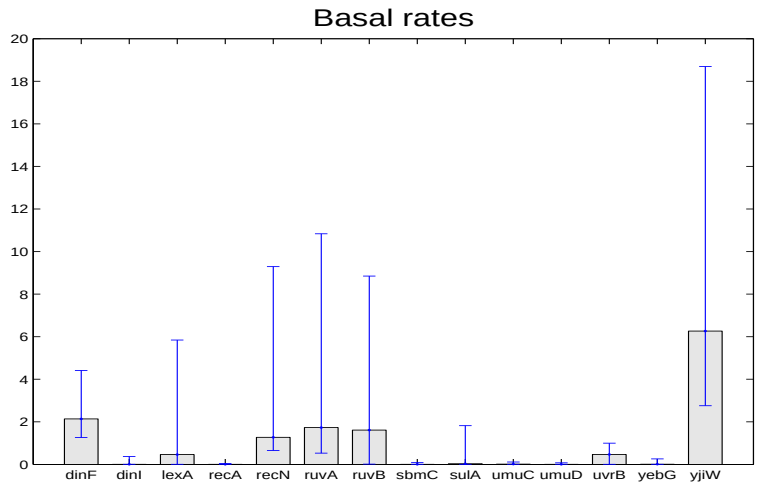


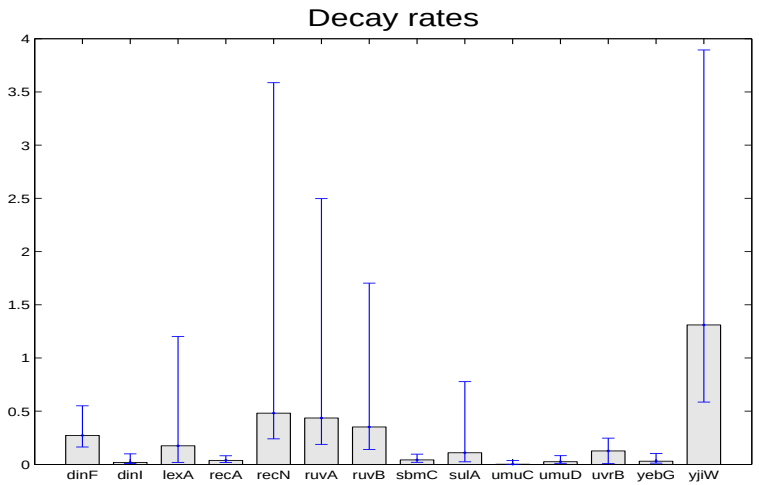


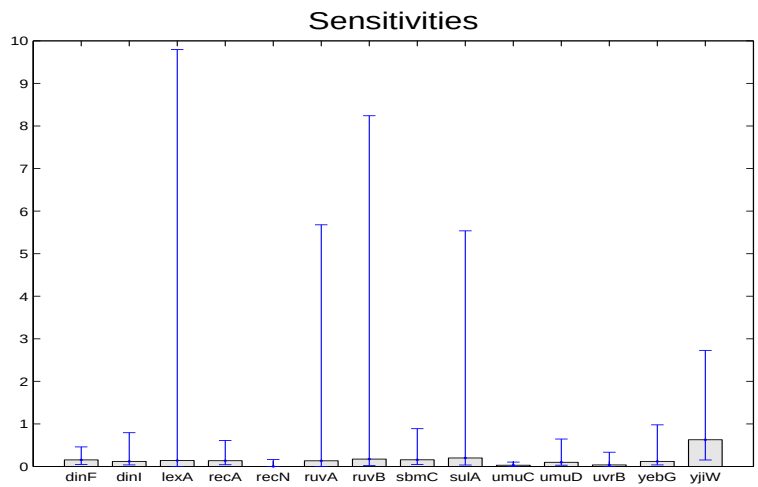


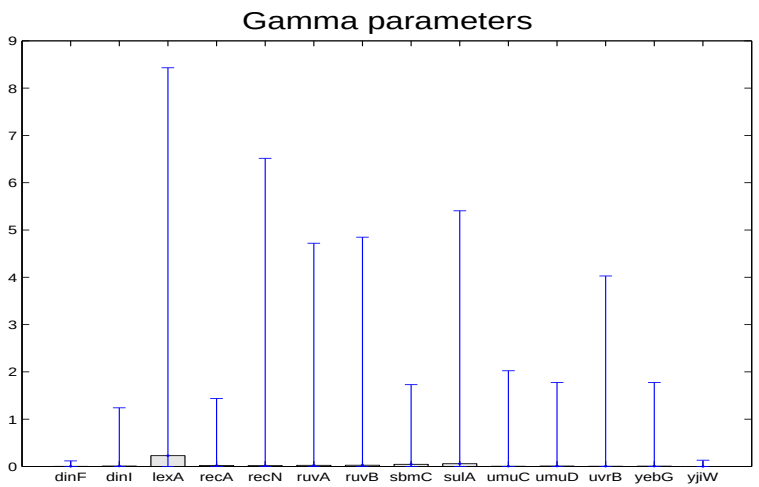


Results in E.coli data: Confidence intervals for the kinetic parameters









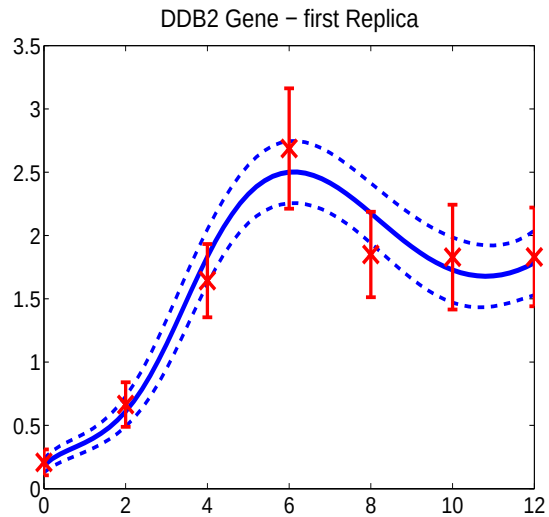
Data used by Barenco et al. [2006]

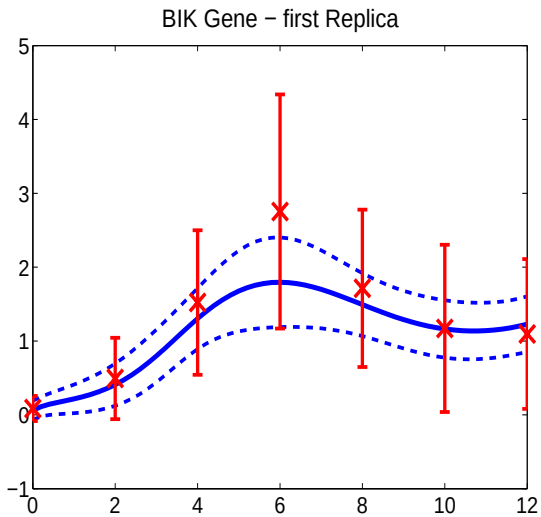
- One transcription factor (p53) that acts as an activator. We consider the Michaelis-Menten kinetic equation

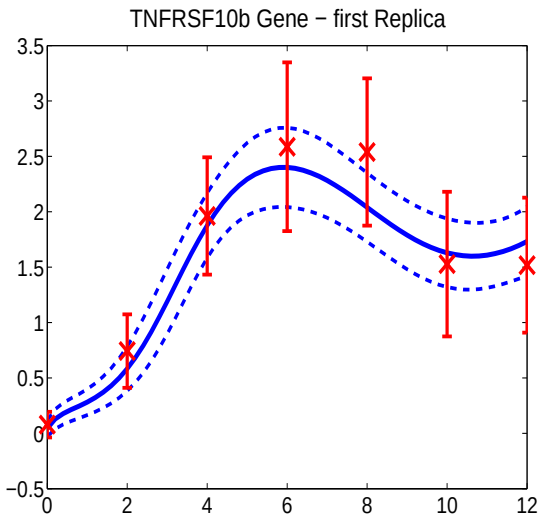
$$\frac{dy_j(t)}{dt} = B_j + S_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - D_j y_j(t)$$

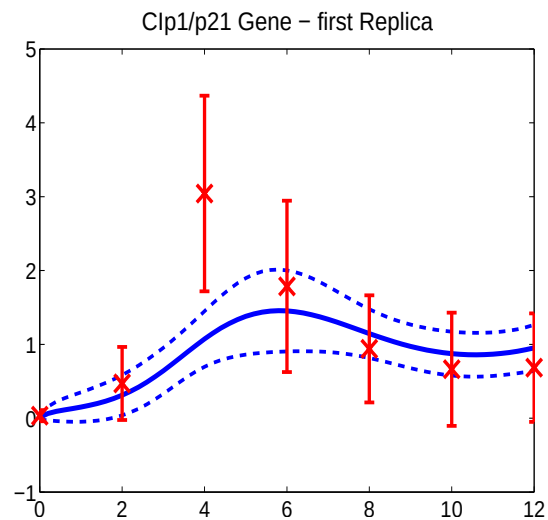
- We have 5 genes
- Gene expressions are available for $T = 7$ times and there are 3 replicas of the time series data
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 7 control points are used (placed in a equally spaced grid)
  - Running time 4/5 hours for 2 million sampling iterations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

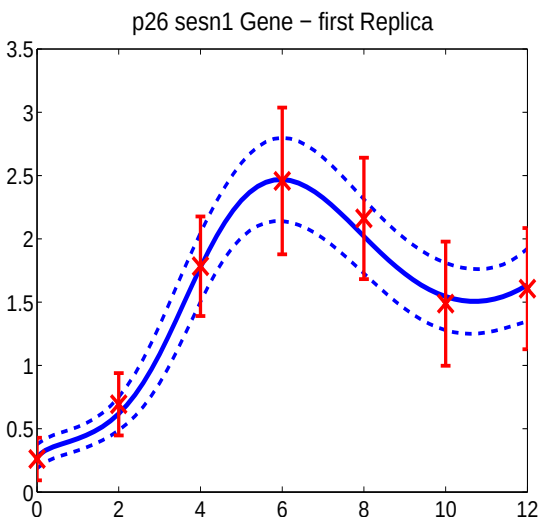
Data used by Barenco et al. [2006]: Predicted gene expressions for the 1st replica



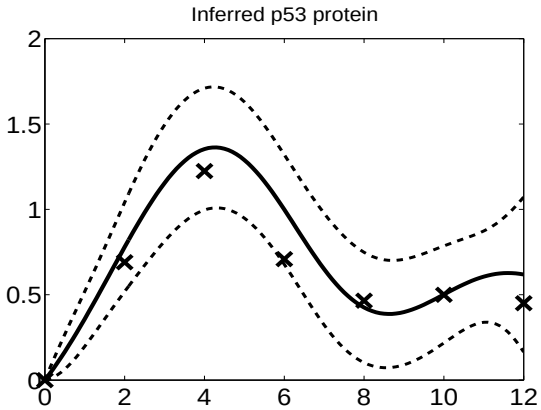


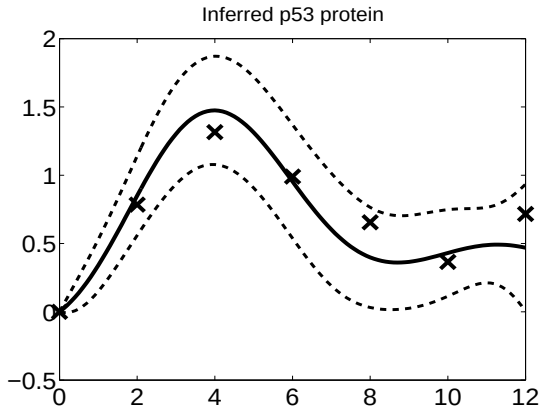


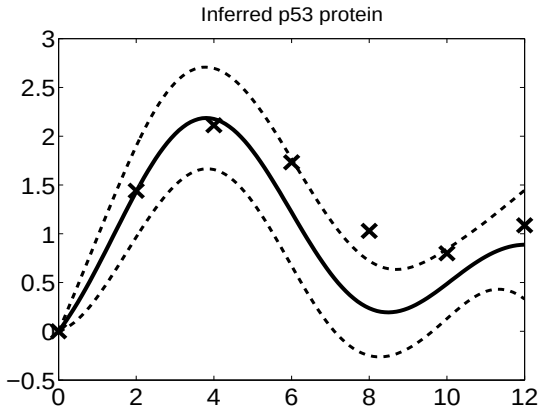




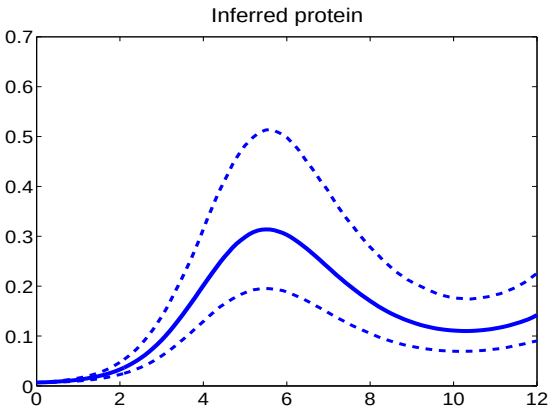
Data used by Barenco et al. [2006]: Protein concentrations

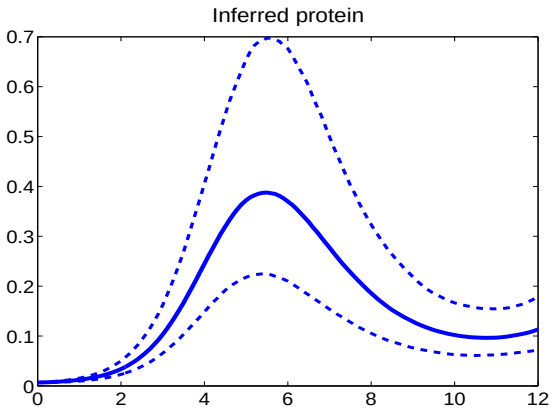


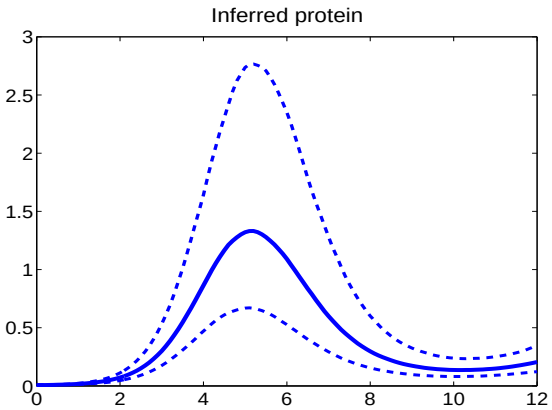




Linear model (Barenco et al. predictions are shown as crosses)

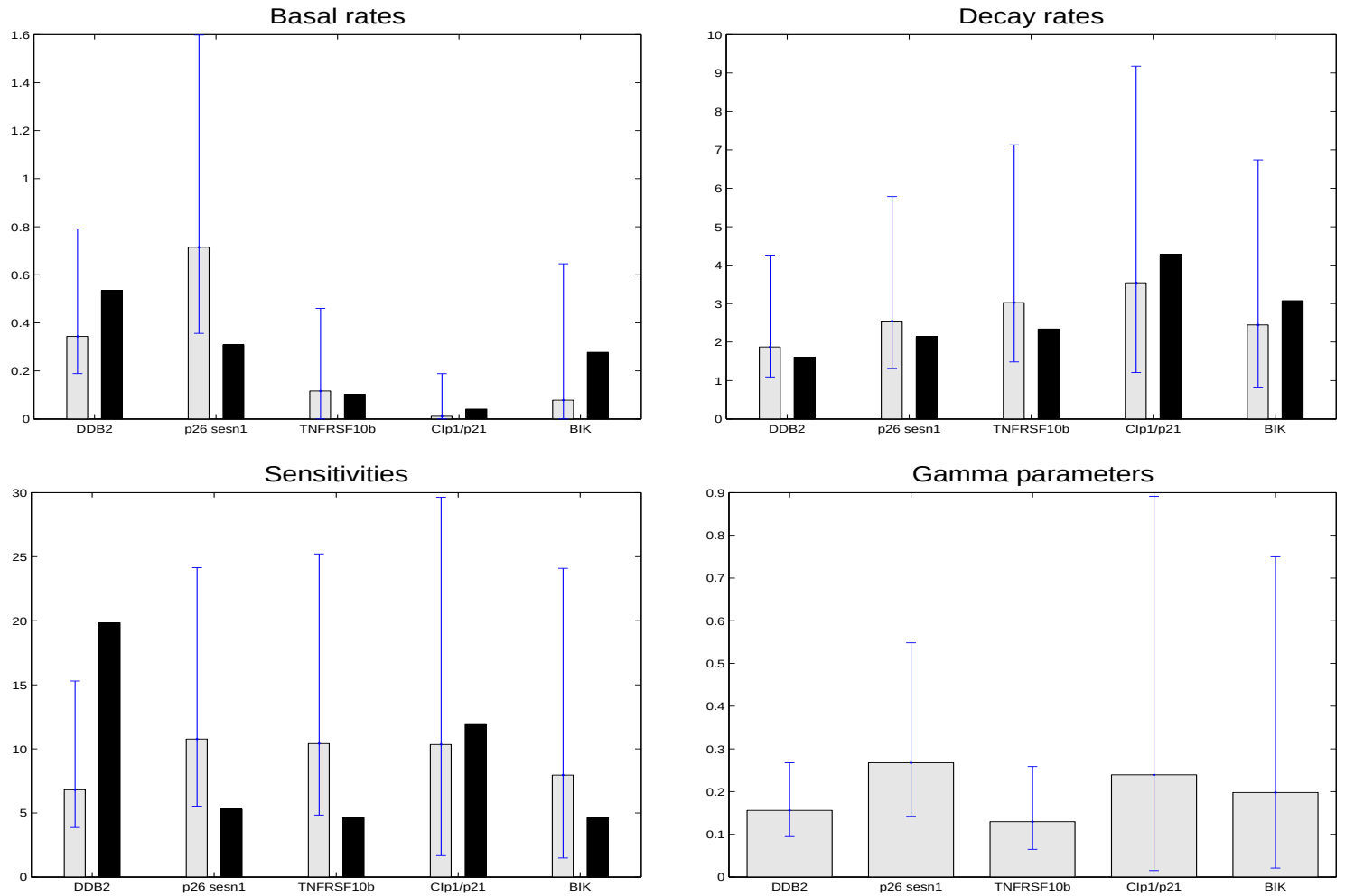






Nonlinear (Michaelis-Menten kinetic equation)

Data used by Barenco et al. [2006]: Kinetic parameters



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

Summary/Future work

Summary:

- A new MCMC algorithm for Gaussian processes using control points
- Continuous full Bayesian inference in transcription factor networks

Future issues:

- Deal with larger systems of ODEs
- Incorporate domain knowledge when you define priors over parameters such as the kinetic parameters