# Estimation of Multiple Transcription Factors using ODEs and Gaussian Processes

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

1 April 2008

## Outline

- Learning a single TF with Gaussian processes
- Multiple TFs
- Experiments and conclusions

# Transcriptional regulation

- Data: Gene expression levels $\mathbf{y} = (y_{jt})$ of $N$ genes at $T$ times
- Goal: Model the dynamics of the expression of a set of genes and infer the single transcription factor (TF) that regulates them
- Model: Use a differential equation (Barenco et al. [2006]; Rogers et. al. [2007]; Lawrence 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 $j$th 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

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

we place a GP prior on the TF concentration function $f(t)$.

- If $g$ is a linear function, then the estimation problem becomes similar to standard GP repression; see Lawrence et. al. [2007].
- However the linear model is less biological plausible. We need to consider positivity constraints and saturation.
- Michaelis-Menten kinetic equation can model this ( the GP prior is placed on the log of the TF):
    - Activation $\frac{dy_j(t)}{dt} = B_j + S_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - D_j y_j(t)$
    - Repression $\frac{dy_j(t)}{dt} = B_j + S_j \frac{1}{\exp(f(t)) + \gamma_j} - D_j y_j(t)$

# Multiple TFs

- We wish to generalize the Gaussian process framework in order to estimate from gene expression data multiple and possibly interacting TFs.
- We are interested in the non-linear response case

# Learning multiple TFs

- General form of the multiple TF model

$$\frac{dy_j(t)}{dt} = B_j + S_j g(f_1(t), \ldots, f_I(t); \mathbf{w}_j) - D_j y_j(t), \qquad (1)$$

where the $I$-dimensional vector $\mathbf{w}_j$ stores the interaction weights between the $j$th gene and the $I$ TFs. There may be also some bias weight $w_{0j}$ for each gene.

# Sigmoid model

- Choose the joint activation function $g(u)$ to be the sigmoid

$$h_j = \sum_{i=1}^{I} w_{ji} f_i(t) + w_{j0},$$

$$g(h_j) = \frac{1}{1 + \exp(-h_j)}.$$

- For single TF the above activation function gives rise to Michaelis-Menten when we fix $w_j = 1$.
- For the repressor case we set $w_j = -1$, which however doesn't give rise to the exact Michaelis-Menten repressor equation

# Bayesian model

- Likelihood:

$$\prod_{j=1}^{N}\prod_{t=1}^{T} p(y_{jt}|\{\mathbf{f}_i(1 \le p \le P_t)\}_{i=1}^{I}, \{A_j, B_j, D_j, S_j\}, \mathbf{w}_j, \sigma_j^2), \tag{2}$$

  where these terms are Gaussians and $\sigma_j^2$ is gene-specific variance
- Prior
  - Kinetics $\{A_j, B_j, D_j, S_j\}$ are positive and are represented in the log space: Gaussian priors are used
  - $\{\mathbf{f}\}_{i=1}^{I}$ are the log of the TFs: GP rbf priors with separate lengthscales
  - $\{\mathbf{w}_j\}$ take real values: Gaussian priors are used
  - Noise variances and GP lengthscales $\{\sigma_j^2, \ell_j^2\}$: Gamma priors

# MCMC

Component-wise M-H algorithm. Iteratively sample from conditional posteriors:

1. For $i = 1, \ldots, I$ sample $\mathbf{f}_i$ from the conditional posterior based on the approach of Titsias et. al [2009]
2. For $j = 1, \ldots, N$ sample the kinetic parameters $\{A_j, B_j, D_j, S_j\}$
3. For $j = 1, \ldots, N$ sample the interaction weights $\mathbf{w}_j$
4. For $j = 1, \ldots, N$ sample the gene-specific noise variance $\sigma_j^2$.
5. For $i = 1, \ldots, I$ sample the lengthscale $\ell_i^2$ of the rbf kernel function.

# Side Information

Learning the real TFs that produced the gene expression is not easy because of identifiability problems in parameter space and limited amount of data. Side information obtained from ChiP data can be useful.

- Side information involves the weights $W$ that represent the interactions between genes and TFs. $W$ is $N \times I$ matrix where $N$ the number of genes and $I$ the number of TFs.
- Side information can be expressed as a binary $N \times I$ matrix $X$. When $x_{ji} = 0$, there is no interaction between the $j$ gene and the $i$ TF, thus $w_{ji} = 0$. When $x_{ji} = 1$, the value $w_{ji}$ can take a positive or negative value which must be inferred by MCMC.
- This scheme can be generalized to probabilistically expressed side information where each $x_{ij}$ is drawn from some probability $\pi_{ji}$ that expresses our prior belief that the $j$ gene has been regulated by the $i$ TF.

# Artificial data

- We consider a toy example with two TFs, that can regulate 20 genes.
- We assume that we have deterministic side information for 8 out of 20 genes. i.e. we know which weights $w_{j1}$ and $w_{j2}$ are zero for these 8 genes, say $j = 1, \ldots, 8$.
- We also assume that the initial conditions in the differential equations are all zero and also that we know the initial (at $t = 0$) activation of the TFs. The number of non-zero elements in the $20 \times 2$ matrix $W$ is 25.

# Artificial data

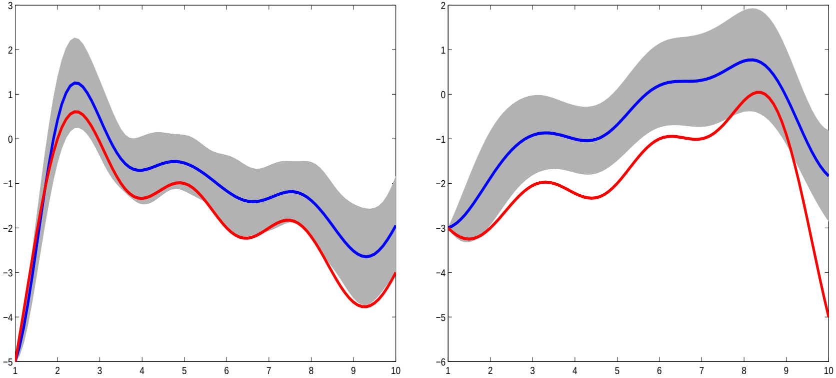

Figure: The inferred profiles of the two TFs (in the log space). With red solid lines are the ground-truth TFs used to generate the toy data. With blue lines shaded error bars are the inferred TF profiles.

# Artificial data

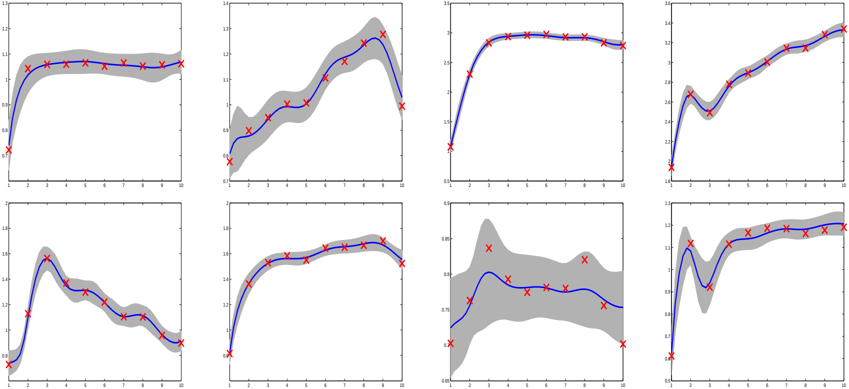

Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

# Artificial data

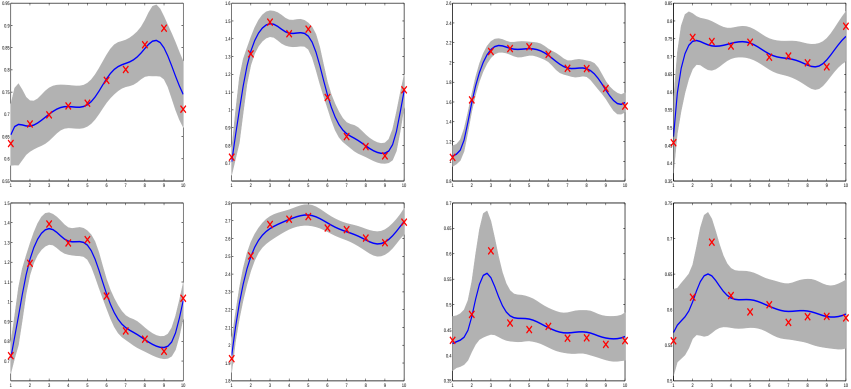

Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

# Artificial data

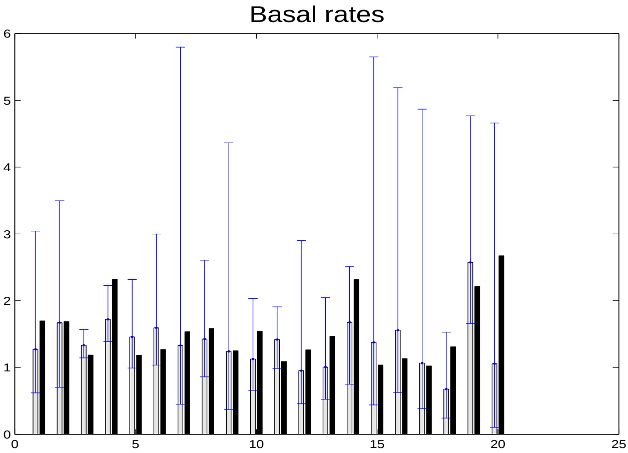

Figure: The inferred basal rates for the 20 genes.

# Artificial data

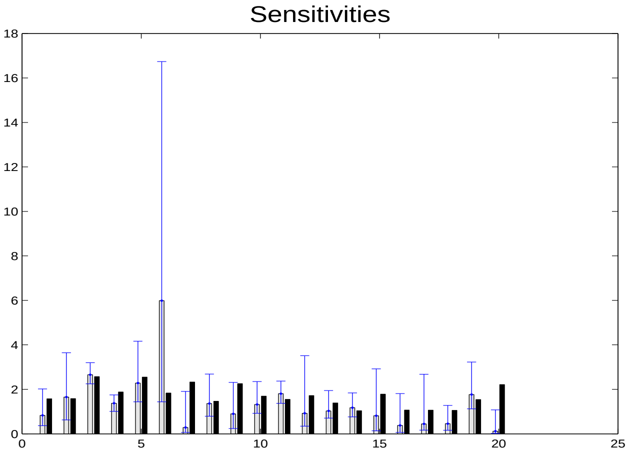

Figure: The inferred sensitivities for the 20 genes.

# Artificial data

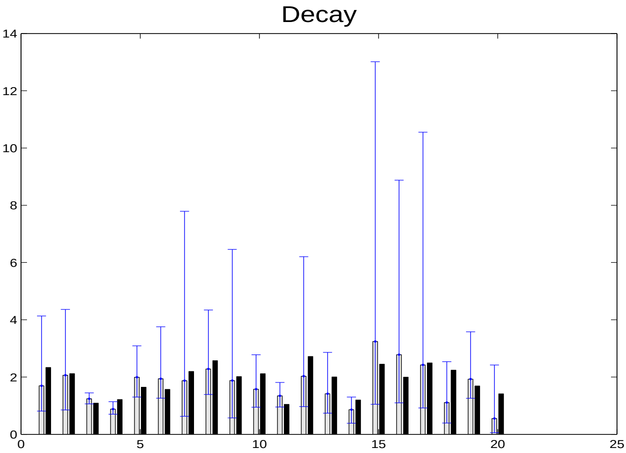

Figure: The inferred decays for the 20 genes.

# Artificial data

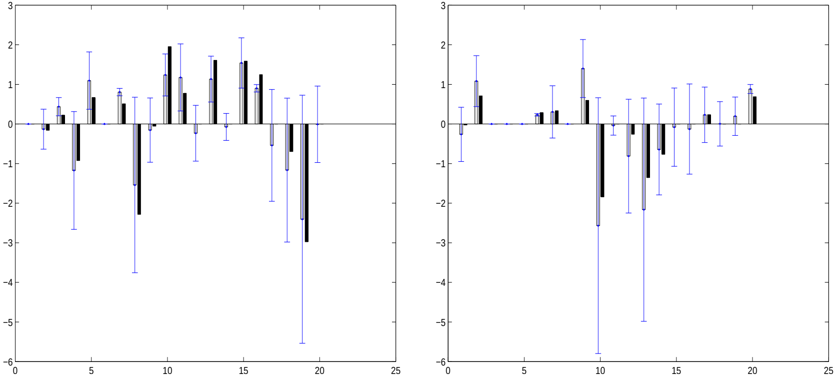

Figure: The inferred interaction weights W. (left) show the interaction weights between the first TF and the 20 genes. (right) show the corresponding weights for the second TF.

## Spelman et. al. yeast data

We selected 30 genes regulated by 3 TFs. The 3 TFs are MBP1, FKH2 and STE12. The selection was done based on the ChiP data avalaible so that only the genes that are regulated exclusively by at least one of these 3 TFs were selected.

# Yeast data

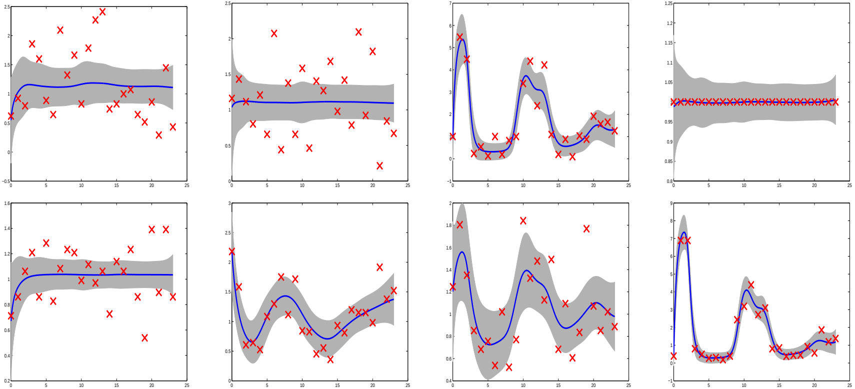

Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

# Yeast data

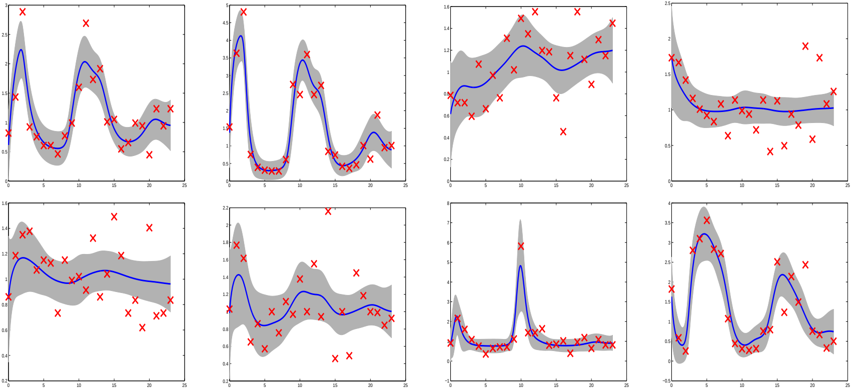

Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

# Yeast data

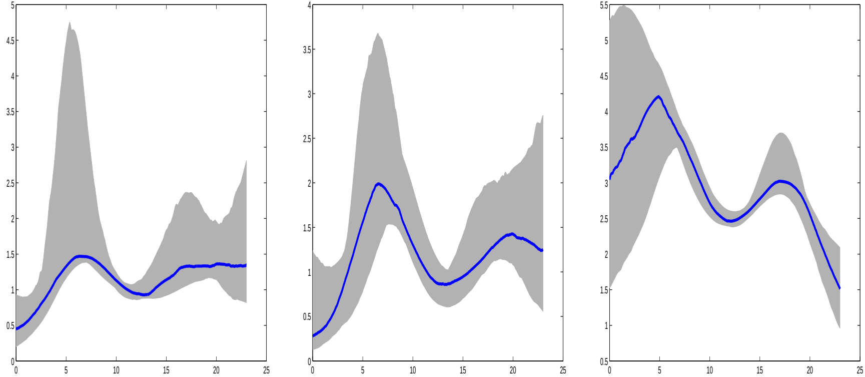

Figure: TF profiles

## Sigmoid model

- The sigmoid model is perhaps less biologically plausible. Particularly it assumes that all TFs (activators and repressors) are combined by multiplication

$$sigmoid = \frac{1}{1 + \prod_{p=1} [\exp(f_p(t))]^{-w_{jp}} \exp(-w_{j0})}$$

recall that $\exp(f_p(t))$ is the TF.

- This does not look so intuitive.
- Can we define activation functions where the combination is done by addition?
- Saturation and the ability of repressors to turn off the gene expression must be incorporated.
- Next we discuss such a model which can be viewed as a generalization of the Michaelis-Menten model for the single TF case.

# Michaelis-Menten multiple TF model

$$\frac{dy_j(t)}{dt} = B_j + S_j g(f_1(t), \ldots, f_I(t); \mathbf{w}_j) - D_j y_j(t), \qquad (3)$$

- Let $\mathcal{P} = \{1, \ldots, P\}$ be the set of all TFs
- $A_j$ be the set of TFs that are activators for $j$th gene and $R_j$ the set of repressors.
- $A_j \cup R_j \subseteq \mathcal{P}$. That is some of the TFs may not regulate the $j$th gene
- The activation function takes the form

$$g = \frac{\sum_{i \in R_j} w_{ji} + \sum_{i \in A_j} w_{jp} \exp(f_i(t))}{1 + \sum_{i \in R_j} w_{ji} \exp(f_i(t)) + \sum_{i \in A_j} w_{ji} \exp(f_i(t))}$$

where $w_{ji}$ are now non-negative and can be thought as relative sensitivities

# Michaelis-Menten multiple TF model

$$g(f_1(t),\ldots,f_I(t);\mathbf{w}_j)=\frac{\sum_{i\in R_j} w_{ji}+\sum_{i\in A_j} w_{ji}\exp(f_i(t))}{1+\sum_{i\in R_j} w_{ji}\exp(f_i(t))+\sum_{i\in A_j} w_{ji}\exp(f_i(t))}$$

- Michaelis-Menten equation for a single TF can be obtained as a special case
  - Activation: $A_j=\{1\}$, $R_j=\emptyset$,

$$g(f_1(t);\mathbf{w}_j)=\frac{w_{j1}f_1(t)}{1+w_{j1}f_1(t)}=\frac{f_1(t)}{\gamma_j+f_1(t)}$$

  - Repression: $A_j=\emptyset$, $R_j=\{1\}$

$$g(f_1(t);\mathbf{w}_j)=\frac{w_{j1}}{1+w_{j1}f_1(t)}=\frac{1}{\gamma_j+f_1(t)}$$

where $\gamma_j=\frac{1}{w_{j1}}$

# MCMC

- Similarly to the sigmoid model. But the set of the activators $A_j$ and the set of repressors $R_j$ are sampled based on Gibbs sampling by taking all possible combinations.

# Artificial data

- We consider a set of 30 genes regulated by 3 TFs.
- Side information: We assume we know which TFs regulate each gene, but we do not know whether a TF activates or represses a certain gene
- We wish to estimate the TF profiles, kinetic parameters, etc
    - and to predict which TFs are activators and which are repressors for each gene

# Artificial data

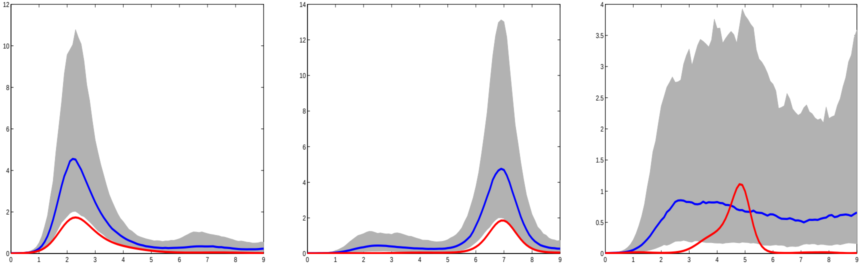

Figure: The inferred profiles of the three TFs. With red solid lines are the ground-truth TFs used to generate the toy data. With blue lines shaded error bars are the inferred TF profiles.

# Artificial data

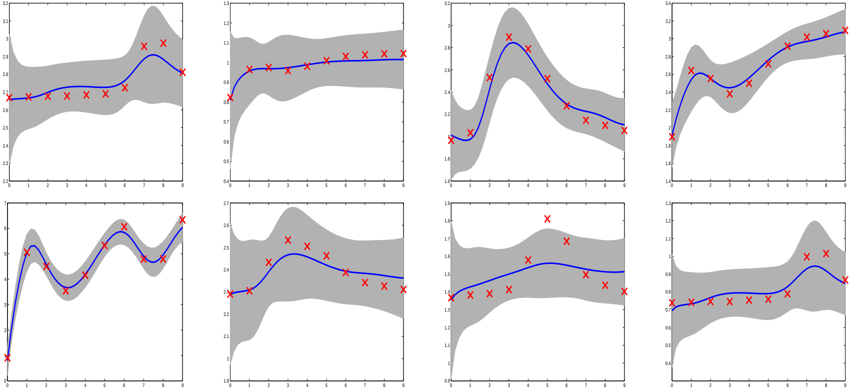

Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

# Artificial data

Total classfication error regarding which TFs are activators and which are repressors for each gene

$$0.2447 \pm 0.0617$$

# Summary/Conclusions

- We presented a framework for learning multiple TFs using ODEs and Gaussian Processes
- The sigmoid and a Michaelis-Menten type of function for combining the TFs was used
- Identifiability issues between TFs profiles, interaction weigths, kinetic parameters can cause problems. Side information can be important.
- Learning multiple TFs is a much more difficult task than estimating a single TF. Our resutls are encouraging, but further research is needed to improve our current methodology