BAYESIAN INFERENCE FOR MODELS OF TRANSCRIPTIONAL REGULATION USING MARKOV CHAIN MONTE CARLO SAMPLING

Florian Stimberg, Andreas Ruttor and Manfred Opper

Department of Computer Science, Technische Universität Berlin
FR 5-8, Franklinstraße 28/29, D-10587 Berlin, Germany
flostim@cs.tu-berlin.de, ruttor@cs.tu-berlin.de, opperm@cs.tu-berlin.de

ABSTRACT

The activity of transcription factors is often difficult to measure directly in micro-array experiments. An alternative approach is to infer these quantities from noisy expression data of the target genes. In this paper we present a Markov chain Monte Carlo sampler suitable for this task. The algorithm uses the full time series of available observations, so that the dynamics of the system state as well as transcriptional parameters can be estimated. Samples are generated either directly from conditional probability distributions or by means of Metropolis-Hastings steps. We test our method on toy data sets of different sizes and compare the result for a real data set of the yeast metabolic cycle with an existing approximation.

1. INTRODUCTION

Understanding the interactions in gene networks is a major challenge in biology. The advances in technology have made available a large amount of data about transcriptional regulation. Although large-scale analysis of these data sets have been conducted for some years (see e.g. [1]), they are usually restricted to structural inference and estimating connection strengths due to limited computational resources. For smaller networks of transcriptional regulation inferring the dynamics appears feasible and is an ongoing topic in systems biology [2, 3].

Transcription of genes is controlled by proteins which can bind to particular base-sequences of DNA. These are called transcription factors (TFs), as they either suppress or amplify the transcription of DNA to mRNA, which then influences the production rate of the corresponding protein. While micro-array data for the expression levels of TF target genes is available, measuring the activity of the TF itself is complicated [1].

In this contribution we present a Markov chain Monte Carlo (MCMC) sampler which infers the TF activity based on a model of differential equations from noisy expression data of the target genes. Additionally, we compute posterior distributions over several parameters of the transcriptional regulation process. Previously, the model used here has been treated by methods of approximate inference [4]. To our knowledge, exact inference has only been done for minimal networks with one TF regulating one gene [3]. In contrast, our approach asymptotically converges to the exact results and can handle multiple TFs regulating multiple genes with combinatorial influences. Additionally, our method can be applied to models with or without system noise. Many approaches often assume deterministic dynamics, even though it is known that genetic regulatory networks are intrinsically noisy [5].

2. MODEL

Expression levels, as given by the concentration of the corresponding mRNA, change smoothly and rather slowly. In contrast, the activity of a TF varies on a much shorter time scale [6]. Therefore we follow [3] and model the dynamics of each TF as a continuous time Markov jump process with two states, active ($\mu_k = 1$) and inactive ($\mu_k = 0$), which is also called *telegraph process*. The slow dynamics of the mRNA concentrations x_i is modelled as a diffusion process with a drift depending on the TF activity.

In the following we consider a system with n TFs and m target genes. Then the dynamics of the mRNA concentration corresponding to the i-th target gene is given by the stochastic differential equation

$$dx_i = (\mu(t)^\top A_i \mu(t) + b_i - \lambda_i x_i)dt + \sigma_i dW, \qquad (1)$$

of the Ornstein-Uhlenbeck type, where $\mu(t) \in \{0,1\}^n$ is the TFs activity, b_i is the base production rate, λ_i is the degradation rate, and σ_i is the strength of the intrinsic noise of the production process, which is modeled by a Wiener process dW. A_i is a n-by-n matrix with $(A_i)_{k,l}$ being the change in production rate of protein i if TF k and l are active. For $l = k$ this is the effect of a single TF on the rate and for $l > k$ the elements are fixed to zero to avoid ambiguity.

The telegraph process $\mu(t)$ has one parameter, its transition rate f. The time Δt between successive switches of a single TF is then exponentially distributed according to $p(\Delta t) = f \exp(-f\Delta t)$, where f denotes the expected number of changes between the two states in one time unit. From this it follows that the total number of switches of a single TF in the time interval $t \in [0; T]$ obeys a Poisson distribution with mean $f \cdot T$.

While the activity of the TFs are completely hidden, we have observations d_{t_j} of the mRNA concentrations at discrete points in time. These measurements are corrupted by Gaussian noise with standard deviation s_i.

3. MCMC SAMPLER

Given a set of noisy observations $D = \{d_{t_0}, \ldots, d_T\}$ of the mRNA concentrations we want to sample from the joint distribution of μ, x, and a subset $\Theta = \{A, b\}$ of the parameters. Here and in the following μ denotes the full path of the TF activity over time, which can be represented by the set of jump times for each TF. Similarly, x stands for the mRNA concentrations at all jump times and observations, which is sufficient to describe this process for sampling purposes. We use a Metropolis-within-Gibbs sampler, which alternates between sampling from $P(x|\mu, \Theta, D)$, $P(\mu|x, \Theta, D)$, and $P(\Theta|x, \mu, D)$. Note, that our algorithm is not based on discretizing the time and therefore does not introduce discretization errors. For models without system noise ($\sigma_i = 0$) sampling alternates between $P(\mu|\Theta, D)$ and $P(\Theta|\mu, D)$ instead, because x is a deterministic function of μ and Θ in this case.

3.1. Sampling x

To sample x, given μ, Θ and the observations a forward-backward algorithm as in [7] is employed to compute the time-dependent drift $\tau(x, t)$ of the posterior process. We can iteratively sample $x_{t_{i+1}}$ given an initial condition from

$$P_{post}(x_{t_{i+1}}|x_{t_i}, \mu, \Theta, D) = \mathcal{N}(x_{t_{i+1}}; m_f, v_f), \quad (2)$$

where m_f and v_f are solutions to

$$\frac{dm_f}{dt} = \tau(m_f(t), t), \qquad m_f(t_i) = x_{t_i} \quad (3)$$

$$\frac{dv_f}{dt} = 2v_f(t)(-\lambda - \frac{\sigma^2}{v(t)}) + \sigma^2, \quad v_f(t_i) = 0. \quad (4)$$

3.2. Inferring the TF activity

Calculating the prior probability $P(\mu|\Theta)$ for a given path of μ is easy (for further information see [8, p. 221]). Since the x-process is Markovian for fixed jump times, its likelihood $P(x|\mu, \Theta)$ can be computed iteratively:

$$P(x|\mu, \Theta) = \prod_{t_i} P(x_{t_{i+1}}|x_{t_i}, \mu, \Theta), \quad (5)$$

$$P(x_{t_{i+1}}|x_{t_i}, \mu, \Theta) = \mathcal{N}(x_{t_{i+1}}; m, v)$$

$$m = x_{t_i} e^{-\lambda \Delta t} + \frac{A\mu + b}{\lambda}(1 - e^{-\lambda \Delta t})$$

$$v = \frac{\sigma^2}{2\lambda}(1 - e^{-2\lambda \Delta t}), \quad (6)$$

where $\Delta t = t_{i+1} - t_i$. As before x consists of sample points at the positions t_i of observations and jumps of the μ-process. Now our goal is to sample from the posterior

$$P(\mu|x, \Theta) \propto P(\mu|\Theta)P(x|\mu, \Theta), \quad (7)$$

which is not directly feasible. Therefore we employ a Metropolis-Hastings step, which accepts a proposal μ^* for the TF activity with probability

$$A = \min\left(1, \frac{P(\mu^*|\Theta)P(x|\mu^*, \Theta)}{P(\mu|\Theta)P(x|\mu, \Theta)} \frac{Q(\mu|\mu^*)}{Q(\mu^*|\mu)}\right), \quad (8)$$

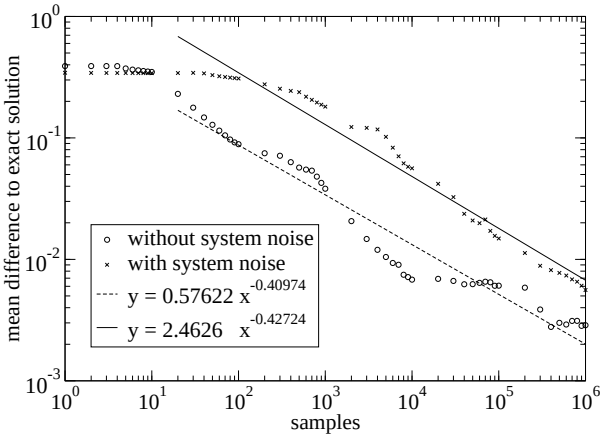

Figure 1. Mean absolute difference between the exact posterior expectation value of μ and the result of the MCMC sampler as a function of the number of iterations. The straight lines are a least squares fit of $y = ax^b$ to the data.

where $Q(\mu^*|\mu)$ is the probability to generate μ^* starting from μ. Otherwise the old sample is used again.

As proposal for a new μ-path we choose one of five possible actions, which modify the current sample:

- Shift jump: A new jump time is drawn from a Gaussian centered around the old jump time. The Gaussian is truncated at the neighboring jump times.
- Add one jump: The time of the new jump is drawn uniformly from the simulation time frame.
- Remove one jump: One of the jumps is drawn from a uniform distribution over all jumps and removed.
- Add two jumps: The time of the first jump is drawn as for adding one jump. The time of the second jump is drawn from a uniform distribution starting at the first jump time and ending at the next jump time.
- Remove two jumps: Two neighboring jumps are selected by drawing from a uniform distribution over all neighboring jump pairs and deleted.

After adding or removing one jump it is chosen with equal probability to invert the state of the μ-process before or after the jump time. While this proposal does not use any information from the data, it is very fast to compute and quickly converges to reasonable states. The option to add or remove two jumps is necessary, because adding or removing only one jump at a time will result in poor acceptance rates, as the whole process after or before the jump is inverted. Our method of adding or removing two jumps on the other hand only inverts the TF activity between these two jumps.

3.3. Sampling the parameters

The most interesting parameters to sample from are probably A and b, which represent the effect of the TF activity

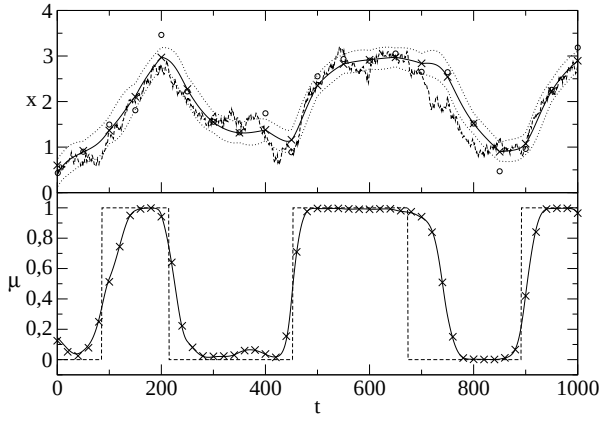

Figure 2. Comparison of the results for the small toy model with system noise. The exact solution (solid line) and the MCMC results(crosses) are shown for the posterior of the mRNA (top) and the TF (bottom). The true values are represented by dashed lines and the noisy observations by circles.

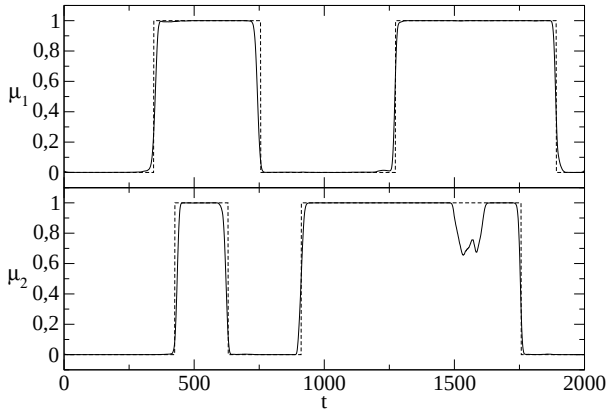

Figure 3. Inference results for the posterior profile of μ for the toy model with two TFs. The dashed lines represent the true path, while the results of the MCMC sampler, which are the probabilities for the active state $\mu(t) = 1$, are plotted as solid lines. To obtain 250,000 samples the sampler took 130 minutes.

on the mRNA production rates and the base production rates, respectively. It is easy to see that A and b are Gaussian in (6), which means that $P(A|x, \mu)$ is also Gaussian, if we chose a Gaussian prior over A, and the same holds for b. This enables us to get direct samples from A and b by iteratively computing the mean and variance generated from (6). Sampling λ, σ, s or the jump rates is not that easy. Direct sampling is not possible, but since the exact likelihood can be evaluated, a Metropolis-Hastings step can be employed, given a suitable proposal distribution.

3.4. Sampling without system noise

Without system noise sampling μ given x does not work, because μ would need to fit *exactly* to the current x-path. Instead, we integrate the x-process out and sample from $P(\mu|D, \Theta)$. Here the likelihood $P(D|\mu, \Theta)$ can be calculated analytically. As before both A and b are linear in the likelihood and thus can be sampled directly.

4. RESULTS

4.1. Synthetic data

For a simple model with only one TF and one target gene it is feasible to compute the exact solution for $x(t)$ and $\mu(t)$ numerically. This calculation is based on solving the backward and forward Chapman-Kolmogorov equations [9], which are partial differential equations describing the time evolution of marginal probability distributions. Details of that algorithm will be provided in a further publication. Then the expectation values obtained from the exact posterior distribution can be compared with the MCMC results. We do this for a toy data set to verify that our sampler converges to the right solution.

Figure 1 shows that this is indeed the case. The deviation of the MCMC result roughly decreases proportional to the square root of the number of samples, which is quite typical for quantities calculated by averaging over samples. The x and μ posterior of the data set with system noise can be seen in figure 2. The results of the MCMC sampler were drawn as crosses at intervals because the differences would otherwise be hard to see. For both simulations 1 million samples were generated from which the first 1000 were dropped as burn-in. The sampler implemented in Matlab ran for 38 (no system noise) and 244 (with system noise) minutes, respectively.

To test our algorithm further another synthetic dataset was generated for a model with 4 target genes and 2 TFs. The first two genes are only positively regulated by the first and the second TF, respectively. This fact was known to the algorithm to allow identifiability of the model. Exact inference for a model of this dimension is not feasible, but both the profile of the TFs and the inferred A parameters fit the true values well, as seen in the figures 3 and 4. The prior distribution over A was very broad with a standard deviation of 0.5 and zero mean. The observation noise had a standard deviation of 0.5 with the steady states lying roughly between 10 and 20.

4.2. Yeast metabolism

For a real example we used a subset of the data from [10], which measured the gene expression of yeast cells during three metabolic cycles. The cycles were triggered by alternating between forcing starvation and providing glucose. As in [4] we took 10 genes and two TFs (FHL1 and RAP1). To guarantee identifiability three of the genes were known to be only regulated by FHL1 and two were only regulated by RAP1. The parameters were not sampled in this case, but fixed to the maximum-likelihood results of the approximation given in [4]. Thus it is possible to compare the state inference of both algorithms without influences caused by differing parameter estimates. Figure 5 shows the inferred TF activity compared to the ap-

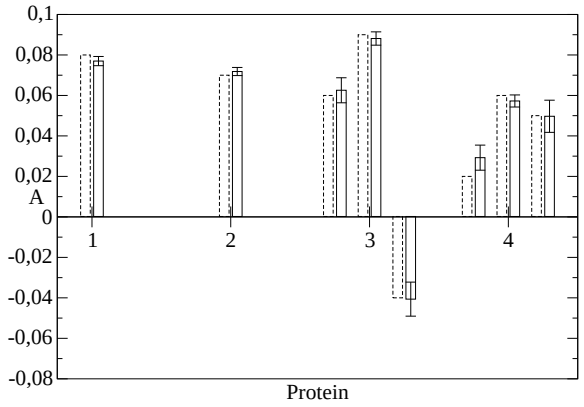

Figure 4. The solid bars show the posterior mean value for the A parameter. For protein 3 and 4, which are influenced by both TFs, the first and second value is the influence of the first and second TF, respectively. The third value is the additional influence when both TFs are active. The dashed bars denote the true values.

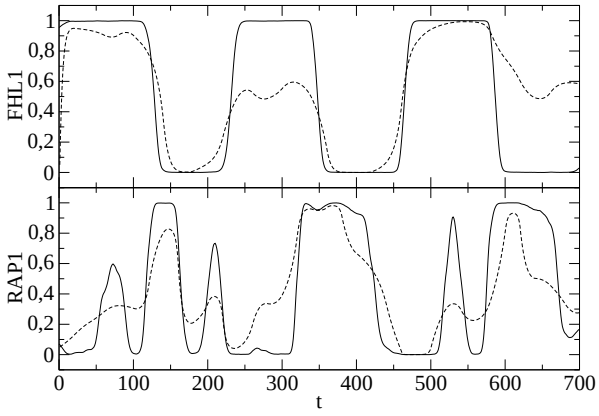

Figure 5. Posterior profile of TF activity from 1 millions runs of the MCMC sampler (solid line) and the approximation of [4] (dashed line). The MCMC sampler used the maximum likelihood parameters of the approximation to allow comparison between the results. The sampler ran for roughly 260 minutes to give 1 million samples.

proximation of [4]. It can be seen that the sampler's results share the general structure of the TF activity with the approximation, but is more confident in most regions.

5. CONCLUSION

Our results prove that MCMC samplers can be used to identify TF activity from noisy data. Compared to existing methods we get samples from the posterior without discretization error or approximative assumptions for non-trivial networks and thus better results than approximations and a broader applicability than exact inference. Parameter inference is an important part of systems biology and we showed that our approach can infer parameters of transcriptional regulation by applying it to toy data sets of multiple TFs and proteins. Besides optimizing the computational costs future extensions might involve models where TFs not only influence the production rate of target genes, but other factors. Additionally, we want the sampler to handle more complicated structures like feed-forward networks.

6. REFERENCES

[1] J. C. Liao, R. Boscolo, Y.-L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, "Network component analysis: Reconstruction of regulatory signals in biological systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 26, pp. 15522–15527, Dec. 2003.

[2] 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*, vol. 7, no. 3, 2006.

[3] G. Sanguinetti, A. Ruttor, M. Opper, and C. Archambeau, "Switching regulatory models of cellular stress response," *Bioinformatics*, vol. 25, pp. 1280–1286, 2009.

[4] M. Opper and G. Sanguinetti, "Learning combinatorial transcriptional dynamics from gene expression data," *Bioinformatics*, vol. 26, no. 13, pp. 1623–1629, July 2010.

[5] M. Thattai and A. van Oudenaarden, "Intrinsic noise in gene regulatory networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 15, pp. 8614–8619, July 2001.

[6] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits (Chapman & Hall/CRC Mathematical & Computational Biology)*, Chapman and Hall/CRC, 1 edition, July 2006.

[7] A. Ruttor and M. Opper, "Efficient statistical inference for stochastic reaction processes," *Phys. Rev. Lett.*, vol. 103, no. 23, pp. 230601, 2009.

[8] D. J. Wilkinson, *Stochastic Modelling for Systems Biology*, Chapman & Hall / CRC, London, 2006.

[9] C. W. Gardiner, *Handbook of Stochastic Methods*, Springer, Berlin, second edition, 1996.

[10] B. P. Tu, A. Kudlicki, M. Rowicka, and S. L. McKnight, "Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes.," *Science (New York, N.Y.)*, vol. 310, no. 5751, pp. 1152–8, 2005.