BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–3

A Bayesian Regression Approach to the Inference of Regulatory Networks from Gene Expression Data - Supplementary Information

Simon Rogers and Mark Girolami

Bioinformatics Research Centre, Department of Computing Science, University of Glasgow, Glasgow UK

ABSTRACT

Motivation: There is currently much interest in reverse-engineering regulatory relationships between genes from microarray expression data. We propose a new algorithmic method for inferring such interactions between genes using data from gene knockout experiments. The algorithm we use is the Sparse Bayesian regression algorithm of Tipping and Faul. This method is highly suited to this problem as it does not require the data to be discretised, overcomes the need for an explicit topology search and, most importantly, requires no heuristic thresholding of the discovered connections.

Results: Using simulated expression data, we are able to show that this algorithm outperforms a recently published correlation based approach. Crucially, it does this without the need to set any threshold on possible connections.

Availability: Matlab code which allows all experimental results to be reproduced is available from `http://www.dcs.gla.ac.uk/~srogers/reg_nets.html`

Contact: srogers@dcs.gla.ac.uk

1 ADDITIONAL FIGURES

1 DATA GENERATION

We have adopted the following methodology for our experiments, as used in Rice *et al*., 2004. Firstly, we generate a synthetic network using an approximate power-law degree distribution. For each gene, we sample the output degree (O_i) - the number of genes regulated by this gene - from the following distribution

$$P(O_i) = \begin{cases} K^{-1}O_i^{-\eta} & O_{min} \leq O_i \leq O_{max} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where the normalisation constant is given by

$$K = \sum_{i=O_{min}}^{O_{max}} O_i^{-\eta}. \quad (2)$$

Following Rice *et al*., 2004 and originally Wagner, 2002 the constant η is set to 2.5. Having sampled O_i, we then randomly sample O_i output connections for each gene, avoiding self-connections. For each connection we randomly sample whether this connection is to be excitatory or inhibitory. In all experiments, the probability of a particular connection being excitatory was set to 0.5. Having created a synthetic network, we then move onto the generation of data from it.

Following Rice *et al*., 2004 and Yeung *er al*., 2002, we simulate knock-out experiments by numerically integrating a system of coupled stochastic differential equations. Denoting by $\dot{\mathbf{e}}_t$ the vector of time derivatives of the expression of the M genes at time t,

$$\dot{\mathbf{e}}_t = -\lambda \mathbf{e}_t + \frac{\kappa + \mathbf{A}_+ \mathbf{e}_t^{\gamma}}{1 + \mathbf{A}_+ \mathbf{e}_t^{\gamma} + \mathbf{A}_- \mathbf{e}_t^{\beta}} + \mathbf{n} \quad (3)$$

where λ is set to unity, the standard rate of degradation κ is set to 0.5. $\mathbf{A}_+$ and $\mathbf{A}_-$ correspond to the positive (excitatory) and negative (inhibitory) components of the connection matrix $\mathbf{A}$. The co-efficients γ and β control the order of the dependency between the children and the parents and $\mathbf{n}$ is an isotropic Gaussian noise term which is independent over time and has standard deviation ϵ.

Although analytically intractable, we can easily track this system to a steady state by using the Euler - Maruyama method (for example, see Higham, 2000). Essentially, given a set of initial conditions (in our experiments, these were always zero) and a step size (δt), we use the following for each gene i

$$\mathbf{e}_{t+1} = \mathbf{e}_t + \delta t \dot{\mathbf{e}}_t + \mathbf{n} \quad (4)$$

where $\mathbf{n}$ is a vector of samples from an isotropic Gaussian with standard deviation ϵ, normalised with respect to δt to ensure that it is invariant under different time steps. Whilst in some cases the choice of step size is crucial to the stability of the Euler - Maruyama method, we found that even with large step sizes, the system converged rapidly to it's steady state.

Gene knock-outs are simulated by forcing the expression of a particular gene to 0 throughout the course of the Euler - Maruyama algorithm. This enables us to take R replicates of any given gene knockout for both wild-type and mutants. Due to the stochastic nature of the system, we are able to simulate these repetitions by simply sampling repeatedly from the

system once it has reached steady state. Some details regarding the implementation of the EM method can be found in Appendix 2.

2 NUMERICAL INTEGRATION OF THE DIFFERENTIAL EQUATIONS

The creation of the expression data required the numerical integration of the system of differential equations defined in Section 2. We have used the Euler-Maruyama method to do this, which is essentially the forward Euler method, with Gaussian noise added at each time step.

The general schema is as follows. Starting with an initial condition (we took this to be zero expression for each gene), we compute the derivative of the expression for each gene and from this create the next expression value as the current value plus the product of the derivative and the time step being used. At each stage we also add Gaussian noise with standard deviation ϵ which is also suitably scaled by the time step (to do this we multiply the noise by the square root of the time step) to insure invariability of noise levels across different size time steps.

The step size used in all experiments was $\delta t = 0.01$, and the system was run from $t = 0$ to $t = 30$. This system would normally have converged by $t = 5$. Data was then sampled at every $100\delta t$ samples from the end.

3 ALGORITHM DETAILS

Suppose we are trying to infer the connections into gene i. We assume that this gene can never be chosen from the M total genes - i.e. we are really choosing from $M - 1$. Defining the following quantities

$$s_j = \mathbf{e}_j^T \mathbf{C}_{-i}^{-1} \mathbf{e}_j \quad (5)$$

$$q_j = \mathbf{e}_j^T \mathbf{C}_{-i}^{-1} \mathbf{e}_i \quad (6)$$

where $\mathbf{C}_{-j}$ is $\mathbf{C}$ with the contribution of possible parent j removed, we can state the algorithm as a slightly modified version of the algorithm given in Tipping and Faul, 2003. All other quantities are defined in section 3.

1. Initialise $\sigma^2 = \text{var}(\mathbf{e}_i)$.
2. Start with a single candidate parent j, and set α_j as follows
$$\alpha_j = \frac{||\mathbf{e}_j^2||}{||\mathbf{e}_j^T \mathbf{e}_i||^2 / ||\mathbf{e}_j||^2 - \sigma^2} \quad (7)$$
3. Compute s_j and q_j for *all* M possible parents.
4. Select a new candidate parent j from all M possible parents.
5. Calculate $\theta_j = q_j^2 - s_j$.
6. If $\theta_j > 0$ and $\alpha_j < \infty$ (j is already a parent), re-estimate α_j.
7. If $\theta_j > 0$ and $\alpha_j = \infty$ (j is not already a parent), add $\mathbf{e}_j$ to the set of parents and calculate α_j.
8. If $\theta_j \leq 0$ and $\alpha_j < \infty$, then j is currently a parent but should be removed, so remove $\mathbf{e}_j$ from the parent set and set $\alpha_j = \infty$.
9. Update s_j and q_j.
10. If convergence has been reached, continue to step 11, otherwise, return to step 4.
11. Calculate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using the final parent set.

REFERENCES

Bhattacharjee, A., Richards, W., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E., Lander, E., Wong, W., Johnson, B., Golub, T., Sugarbaker, D., Meyerson, M., (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma sub-classes, *Proc. Natl. Acad. Sci.*, **98**, 13790–13795.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B. (1998) Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization, *Molecular Biology of the Cell*, **9**, 3273–3297.

Friedman, N., Linial, M., Nachman, I., Pe'er, D. (2000) Using Bayesian Networks to Analyze Expression Data, *Journal of Computational Biology*, **7**, 601–620.

Voit, E. (2000) Computational Analysis of Biochemical Systems, Cambridge University Press.

Husmeier, D. (2003) Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, *Bioinformatics*, **19**, 2271–2282.

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, E., Wild, D., Falciani, F. (2004) Modeling T-cell activation using gene expression profiling and state-space models, *Bioinformatics*, **20**, 1361–1372.

Rice, J., Tu, Y., Stolovitzky, G. (2004) Reconstructing biological networks using conditional correlation analysis, *Bioinformatics Advanced Access, 14/10/2004*

Tipping, M. and Faul, A. (2003) Fast Marginal Likelihood Maximisation for Sparse Bayesian Models, *Proceedings of Artifical Intelligence and Statistics*.

Yeung, M., Tegner, J., Collins, J. (2002) Reverse engineering gene networks using singular value decomposition and robust regression, *Proc. Natl. Acad. Sci.*, **99**, 6163–6168.

Wagner, A. (2002) Estimating coarse gene network structure from large-scale gene perturbation data, *Genome Research*, **12**, 309–315.

Higham, D. (2000) An algorithmic introduction to numerical simulation of stochastic differential equations, *Department of Mathematics, University of Strathclyde*, 26(2000).

Zak, D., Doyle, F., Gonye, G., Schwaber, J. (2001) Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data, *Proceedings of the second international conference on systems biology*, 231–238.

Zak, D., Doyle, F., Schwaber, J. (2002) Local identifiability: when can genetic networks be inferred from microarray data?, *Proceedings of the third international conference on systems biology*, 236–237.

Weaver, D., Workman, C., Sotrmo, G. (1999) Modeling regulatory networks with weight matrices, *Proceesings of the Pacific Symposium on Biocomputing*, **4**, 112–123.

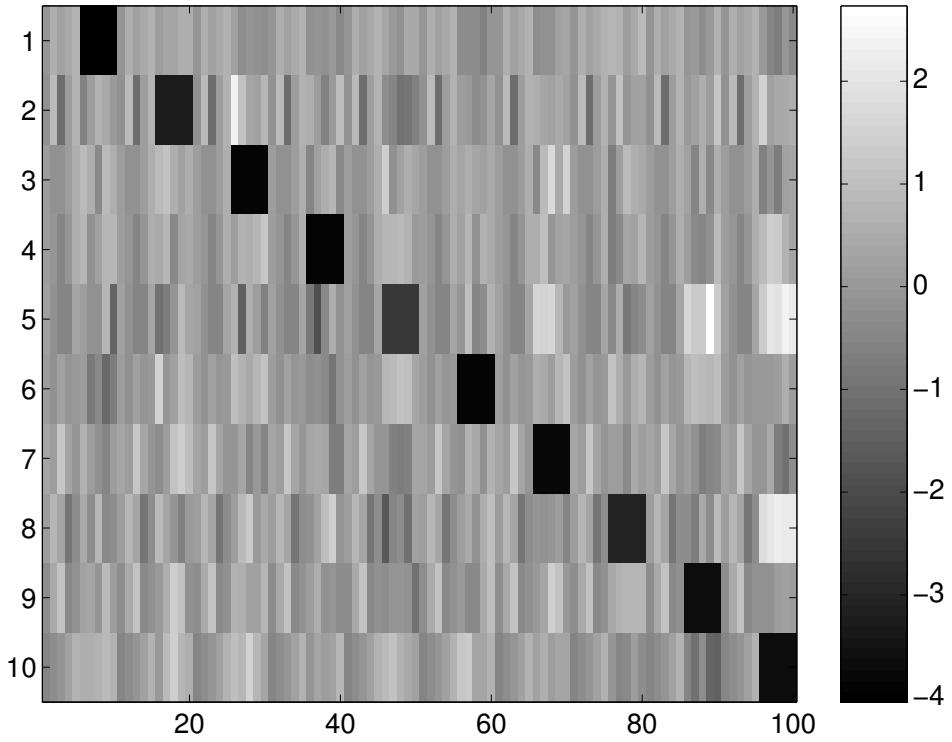

Fig. 1. Example data from a 10 gene network, with $\epsilon = 0.1$. The dark blue patches on the diagonal correspond to the samples where each particular gene has been knocked-out

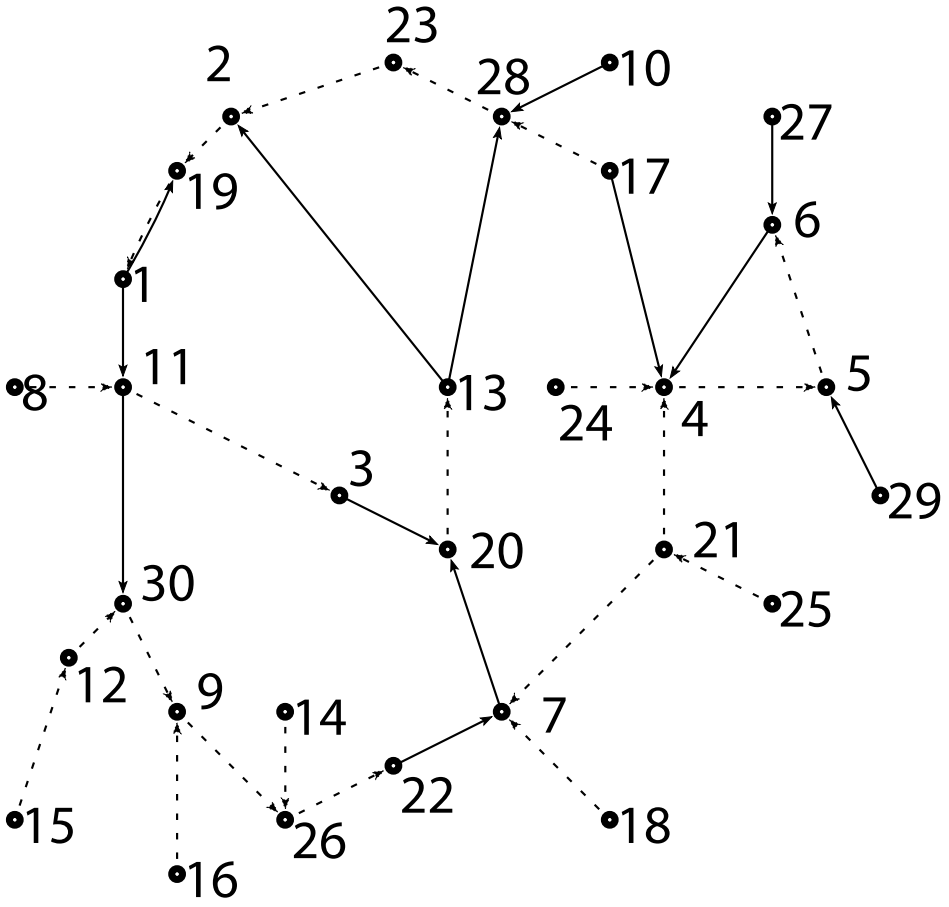

Fig. 2. Example network with $M = 30$ genes