Incorporating heuristics in a swarm intelligence framework for inferring gene regulatory networks from gene expression time series

Kyriakos Kentzoglanakis, Matthew Poole, and Carl Adams

School of Computing
University of Portsmouth
Portsmouth, UK
kyriakos.kentzoglanakis@port.ac.uk
matthew.poole@port.ac.uk
carl.adams@port.ac.uk

Abstract. In this paper, we address the problem of reverse-engineering a gene regulatory network from gene expression time series. We approach the problem by implementing an ant system to generate candidate network structures. The quality of a candidate structure is evaluated using a particle swarm optimization algorithm that tunes the parameters of the corresponding model, by minimizing the error between the actual time series and the trained model's output. We extend this approach by incorporating domain-specific heuristics to the ant system, as a mechanism that has the potential to bias the pheromone amplification effect towards biologically plausible relationships. We apply the method to a subset of genes from a real world data set and report on the results.

1 Introduction

Gene expression is the process by which a gene's DNA sequence is converted through a series of steps into a functional product: the protein. This cellular process constitutes the central dogma of molecular biology, i.e. that genes code for proteins. During this process, DNA is first transcribed (copied) to an intermediate macromolecular form, the mRNA (messenger RNA), which is then translated to protein. Proteins are involved in essential functions of a living organism, including transcription, the catalysis of chemical reactions, cell signalling etc.

Certain genes code for special proteins called transcription factors, which are responsible for regulating the expression of other genes (targets). Transcription factors bind a cis-regulatory site in the promoter region of the target gene, thus inducing a change in the target's rate of transcription. The nature of change specifies this effect as either activatory, in case of an increase in the target's rate of transcription, or repressive (inhibitory) in case of a decrease [1].

A gene regulatory network (GRN) is a complex network of causal relationships between genes, where connections represent regulatory interactions between activators or repressors and targets.

With the advent of DNA microarray technology that measures the mRNA levels of thousands of targets, it has become possible to observe such complex biological processes by taking snapshots of the cellular state and capturing the expression profiles of thousands of genes simultaneously. Gene expression data can either be static, with gene profiles from different organisms, each typically characterized by a class value, or dynamic in the form of gene expression time series from the same organism.

The problem of reverse-engineering GRNs from gene expression data is a major issue in systems biology [2]. A principal obstacle is the relative insufficiency of observations (typically tens or a few hundreds) compared to the number of genes measured (in the order of thousands or a few tens of thousands), the so-called curse of dimensionality.

Additionally, the common practice of validating the biological plausibility of inferred causal relationships by consulting the relevant literature, albeit unavoidable, is controversial because, in the absence of such experimental evidence for a putative connection, there is no apparent method of classifying it either as a previously unknown interaction or as just a spurious edge [3].

In this paper, we describe a swarm intelligence approach to the problem of reverse-engineering GRNs from gene expression time series. We model a GRN as a graph, upon which the ant colony optimization (ACO) meta-heuristic is implemented for the selection of putative GRN architectures. The selected structure is then modelled as a recurrent neural network (RNN), whose parameters (weights and bias terms) are optimized using particle swarm optimization (PSO), so as to minimize the error between the model's output and the actual time series.

Our approach extends the work by Ressom et al. [4], first by changing the way candidate architectures are constructed by individual artificial ants and, second, by introducing a heuristic metric with the intention to bias the probabilistic edge selection process towards biologically plausible relationships.

In the next section, we present an overview of existing approaches to the problem of GRN inference from time course gene expression data. In section 3, the proposed framework is outlined by describing its components and their interrelationships. In section 4, we report on the results of applying the method to a subset of known genes from the yeast gene expression data set and we discuss some of the issues that emerged, before the paper's conclusion in section 5.

2 Existing Approaches

The earliest approaches to the problem of inferring gene relationships from time course gene expression data, were cluster analysis methods, mostly based on global correlation metrics, such as Pearson correlation coefficient, mutual information etc., that extracted co-regulation information out of co-expressed gene clusters [5][6]. These pioneering, model-free methods essentially group genes according to their expression levels, providing an insight into the functionality of unknown genes based on the cluster in which they belong. However, they do not

take the temporal nature of data into consideration and do not assign regulatory roles to genes, since, given two genes that are co-expressed (have similar expression), it is not clear which regulates the other. Nevertheless, cluster analysis is still useful, primarily as a technique to reduce the search space and improve the performance of algorithms.

Model-based methods, on the other hand, operate by assuming the existence of a model that represents the gene regulatory network and attempt to train this model based on the available artificial or experimental data. In essence, they attempt to reconstruct the architecture by reproducing the system dynamics. Such models include Boolean networks, Bayesian networks, linear additive models, systems of differential equations, power law systems etc. [7]

In Boolean networks, the state of a node at one time point is a boolean function of the states of K other nodes at the previous time point. As such, they constitute binary idealizations of genetic network architectures that, while succeeding in the simulation and analysis of global dynamics [8], seem to suffer from the problem of information loss during data binarization.

Dynamic Bayesian networks are models of joint, multivariate probability distributions that attempt to represent conditional independence relationships between variables. Their strength in representing noisy, stochastic processes due to their probabilistic nature, makes them good candidates for addressing the problem of inferring gene regulatory networks [9].

In linear additive (neural) models [10][11], the output of each node is a combination of inputs from all other nodes, a function of the weighted sum of their expression levels. Zero weights indicate no regulation, positive weights signify activation, while negative weights signify repression. The assumption of linearity is not a severe one [12], especially if one considers the statistical treatment of microarray data and the increased levels of noise.

Ressom et al. [4] implement a swarm intelligence framework where an ant system, driven only by pheromone amplification, is used for the selection of putative network structures. For each gene (regulator), each artificial ant considers all 2^n regulator-target combinations, where n is the number of genes, for the construction of a candidate architecture. After a structure has been formed, the corresponding model (RNN) is optimized using PSO, in order to evaluate the quality of the selected structure.

Xu et al. [13] deploy a discrete version of PSO for structure selection and a continuous version for model training. They also discuss the relative difficulty of reconstructing the correct regulatory network structure over reproducing the correct dynamics, explaining that there is no unique network to satisfy the data upon which inference is based. Reconstructing the structure depends upon reproducing the system dynamics and, therefore, is a problem of higher order.

3 Methods

Our approach uses an ACO implementation, on a graph with nodes representing genes and directed edges representing regulatory (causal) relationships, to select

putative network architectures, driven by pheromone amplification and heuristic information, where:

- pheromone trails are updated according to the ability of the model (RNN) that represents the selected structure to reproduce the time series, after having been trained using a PSO algorithm.
- the desirability value for a particular edge is calculated by a suitably defined heuristic function.

A candidate gene network structure is represented by a recurrent neural network model, whose update equation is given by:

$$x_i(t) = f(\sum_{j=1}^{N} w_{ij} x_j(t-1) + b_i) \tag{1}$$

where $x_i(t)$ is the value (expression level) of node i at time t, b_i a bias term and weights w_{ij} express the influence of node j to node i, ranging from -1 (gene j represses gene i) to 1 (gene j activates gene i). A value of 0 signifies no regulation. f is a nonlinear transfer function, either the logistic or the hyperbolic tangent.

Network architectures are constructed using the ACO meta-heuristic [14], whereby artificial ants navigate a graph of N nodes, where N is the number of genes in the time series. Each artificial ant probabilistically selects K regulator nodes for each target node in the graph, resulting in a candidate network structure $S = \{e_{ji}\}$ of NK connections. The parameter K reflects the fact that gene networks are sparse and that a gene is regulated by only a handful of other genes. An edge e_{ji} represents a regulatory relationship from node j to node i. The probability of selection of node j as a potential regulator of node i is given by:

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{j=1}^{N} \tau_{ij}^{\alpha} \eta_{ij}^{\beta}} \tag{2}$$

where τ_{ij} is the pheromone value of edge e_{ji}, η_{ij} is the selection desirability of edge e_{ji} based on a suitably defined heuristic function and α, β are their respective relative influences.

After a candidate structure S has been constructed, its quality is assessed by tuning the corresponding model's parameters in order to compare its predicted output with the actual time series. The synaptic weights of the edges that are not part of the selected structure are locked to 0.

Optimization of the model's parameters is performed using a PSO algorithm [15], where each particle's position is encoded as a vector $\mathbf{w}_S$ of size $N(K+1)$ that contains the weights of the selected edges, as well as the bias terms. The quality of a particle's position is determined by calculating the MSE between the predicted model output and the actual time series:

$$\epsilon(\mathbf{w}_S) = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} [x_i(t) - x_i^{\mathbf{w}_S}(t)]^2 \tag{3}$$

where T is the number of available time points, N is the number of genes, $x_i(t)$ is the actual expression level of the i^{th} gene at time t and $x_i^{\mathbf{w}_S}(t)$ is the predicted expression level of the i^{th} gene at time t. The predicted time series are calculated by setting up the model using $\mathbf{w}_S$ and running it using each state of the actual time series, in order to obtain the next state of the predicted time series.

After the threshold of maximum allowed PSO iterations has been reached, the minimum achieved error $\epsilon(\mathbf{w}_S)$ is returned to the ACO algorithm as the quality of the selected structure S. The pheromone matrix is then updated according to:

$$\tau_{ij} = \frac{1}{\epsilon(\mathbf{w}_S)} \qquad \forall e_{ji} \in S \tag{4}$$

The incorporation of heuristics to probabilistic structure selection offers a way of enriching a domain-agnostic procedure with problem-specific insights. The heuristic factor η_{ij} from equation (2) can be defined as a function $\eta : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{R}$ that maps a pair (i, j) to a score that reflects the strength and nature of gene's j influence on gene i. In this context, strength means the likelihood of regulation and nature means the type of regulation (activation or repression).

Table 1. Scoring matrix for event matching. The score of a pair of symbols is a function of the time lag dt between two events. $S(dt)$ is a linearly decreasing function with $0 < S(dt) < 1$, so that the bigger the time lag, the less likely a causal effect is to be assumed. In case of a negative dt, the match is assigned a maximum penalty. Parameters a and b range from 0 to 1 and their role is to emphasize particular matching forms, based on biological arguments [16].

	R	C	F
R	$S(dt)$	0	$-bS(dt)$
C	0	0	0
F	$-bS(dt)$	0	$aS(dt)$

For the purpose of demonstrating our approach, we are using a heuristic proposed by Kwon et al. [16]. They hypothesize that if a rise in the expression of gene A is followed by a rise in the expression of gene B, then this indicates that gene A potentially activates gene B. Conversely, if a rise in the expression of gene A is followed by a fall in the expression of gene B, then gene A is a potential repressor for gene B.

These expression changes in a gene's temporal profile are encoded as 'events', by calculating the slope of the expression profile at every time interval and classifying it as either 'R' (rising), 'F' (falling) or 'C' (constant). A variation of the Needleman-Wunsch algorithm for sequence alignment [17] is then used to determine the best possible alignment for a pair of event strings, by using the event scoring matrix shown in Table 1.

Given the expression levels of two genes, one of which is assumed to be the regulator and the other the target, the algorithm first calculates the score for the presumed activatory relationship and then for the inhibitory relationship, by complementing the event string of the target. This is done by swapping 'R's with 'F's, while 'C's remain intact. The maximum score of the two is returned

to ACO as the overall score of the particular relationship and is cached to avoid recalculation.

4 Results

We selected 5 cyclin genes that are known to be involved in cell cycle regulation, from the *S. cerevisiae* (yeast) data set published in Spellman et al. [18], for the purpose of comparing our results to those of Ressom et al. [4]. The yeast data set contains multiple time series from the yeast cell cycle; we chose the *cdc15* time series, consisting of 24 time points (more than the others). Gene expression levels were first smoothed, by using a sliding window method (convolution of a scaled Hann window with the expression profile), and consequently normalized between 0 and 1.

Table 2. The known relations for the collection of selected genes come from PathwayStudio software, as reported in Ressom et al. [4]. The last column summarizes how our algorithm compares with their predictions.

Relation Type	Known Relation	Predicted by [4]	Our Prediction
Expression	CLB1 $\leftarrow$ CLB6	yes (reversed)	yes
Expression	CLB1 $\leftarrow+$ CLB2	yes	yes
Regulation	CLB6 $\rightarrow$ CLB5	yes (reversed)	yes
Regulation	CLB6 $+\rightarrow$ CLB2	no	yes (opposite sign)
Regulation	CLB1 $\leftarrow+$ CLB5	yes (reversed)	no
MolSynthesis	CLB1 $+\rightarrow$ CLB2	yes	yes
Direct Regulation	CLB6 $+\rightarrow$ cdc28	yes (reversed)	yes (reversed)
Direct Regulation	CLB5 $+\rightarrow$ cdc28	yes	yes
Direct Regulation	CLB2 $+\rightarrow$ cdc28	yes	yes
Direct Regulation	CLB2 $\leftarrow+$ CLB5	no	yes (opposite sign)
Direct Regulation	CLB1 $+\rightarrow$ cdc28	yes	no

For the PSO implementation we used a swarm with the global best topology, a population size of 15 particles, a maximum number of 2000 iterations, $\phi_1 = \phi_2 = 2$ and a random inertia weight ω drawn from a uniform distribution, ranging from $\omega_{min} = 0.3$ to $\omega_{max} = 0.8$.

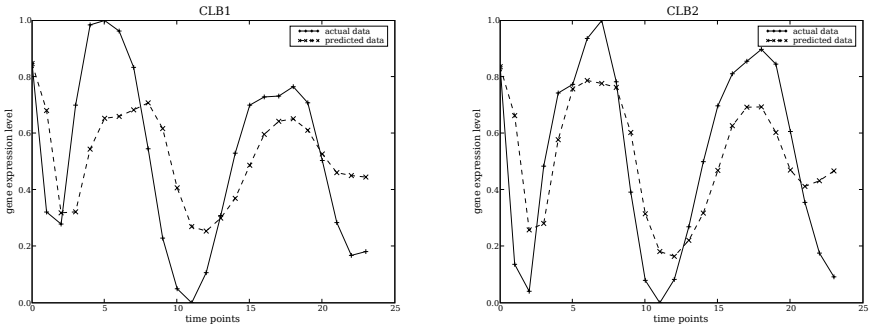
The settings for ACO were set as follows: the relative influences of pheromone and heuristic value $\alpha = 1$ and $\beta = 1$ respectively, the pheromone evaporation rate $\rho = 0.1$ and the number of regulators for a given target gene $K = 2$. The colony size was set to 5 and it was allowed to run for 50 steps.

We performed 10 such experiments and recorded the number of times each edge was selected. We considered a particular relationship to be inferred if the corresponding graph edge was selected at least half of the times, during all experiments. The average MSE of RNN training was 0.058 with a standard deviation of 0.0026.

The results, as shown in Table 2, do not indicate a notable (if any) improvement over the predictions in [4]. The incorporation of the selected heuristic metric

does not seem to influence structure selection in a decisive manner. Perhaps, this is due to the relative influences of pheromone value and heuristic desirability, α and β, being equally weighted.

Table 3. Two examples of actual gene expression levels from the original time series and predicted levels from the optimal RNN that resulted from the experiments.



Two of our predicted, putative connections, namely CLB2 → CLB6 and CLB5 → CLB6, are not reported as known relationships by [4] and their biological plausibility can only be verified experimentally.

5 Further Work

The reported early results that have been presented in this paper, form part of an ongoing study into a swarm intelligence perspective to the problem of reverse-engineering gene regulatory networks. The proposed framework allows for the incorporation of an arbitrary number of problem-specific heuristics, perhaps with an appropriately defined weighting scheme, to a model-based optimization approach.

The behaviour of the ant system needs to be studied in relation to the values of its parameters and the aggregation of heuristics. The suitability of different models, representing selected structures, is also a path to be explored.

Furthermore, we note that our experiments have used a hand-picked subset of temporal gene expression profiles. An investigation of the algorithm's scalability is necessary, particularly when considering the full set of genes, whose expression levels are captured in a real world data set.

References

1. Alon, U.: An introduction to systems biology: design principles of biological circuits. Chapman & Hall/CRC (2007)

2. Kitano, H.: Computational systems biology. Nature **420** (2002) 206–210
3. Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics **19**(17) (2003) 2271–2282
4. Ressom, H., Zhang, Y., Xuan, J., Wang, Y., R. Clarke: Inference of gene regulatory networks from time course gene expression data using neural networks and swarm intelligence. In: IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology. (2006) 1–8
5. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. PNAS **95**(25) (1998) 14863–14868
6. D'Haeseleer, P., Wen, X., Fuhrman, S.: Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. In: Second International Workshop on Information Processing in Cell and Tissues. (1998) 203–212
7. de Jong, H.: Modeling and simulation of genetic regulatory systems : a literature review. Journal of Computational Biology **9**(1) (2002) 69–105
8. Somogyi, R., Fuhrman, S., Askenazi, M.: The gene expression matrix: towards the extraction of genetic network architectures. Nonlinear Analysis, Theory, Methods & Applications **30**(3) (1997) 1815–1824
9. Perrin, B., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., d'Alche Buc, F.: Gene networks inference using dynamic Bayesian networks. Bioinformatics **19**(suppl. 2) (2003) ii 138–ii 148
10. Vohradsky, J.: Neural model of the genetic network. Journal of Biological Chemistry **276**(39) (2001) 36168–36173
11. Wahde, M., Hertz, J.: Modeling Genetic Regulatory Dynamics in Neural Development. Journal of Computational Biology **8**(4) (2001) 429–442
12. Pournara, I., Wernisch, L.: Factor analysis for gene regulatory networks and transcription factor activity profiles. BMC Bioinformatics **8**(61) (2007)
13. Xu, R., Wunsch, D.C., I., Frank, R.: Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ACM Transactions on Computational Biology and Bioinformatics **4**(4) (2007) 681–692
14. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from natural to artificial systems. Oxford University Press (1999)
15. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks. Volume 4. (1995) 1942–1948
16. Kwon, A., Hoos, H., Ng, R.: Inference of transcriptional regulation relationships from gene expression data. Bioinformatics **19**(8) (2003) 905–912
17. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48** (1970) 443–453
18. Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., P. O. Brown, D. Botstein, B. Futcher: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297