Gene Network Inference using a Swarm Intelligence Framework

Kyriakos Kentzoglanakis
School of Computing
University of Portsmouth
UK
kyriakos.kentzoglanakis@port.ac.uk

Matthew Poole
School of Computing
University of Portsmouth
UK
matthew.poole@port.ac.uk

ABSTRACT

In this paper, we present a framework for inferring gene regulatory networks from gene expression time series. A model-based approach is adopted, according to which the quality of a candidate architecture is evaluated by assessing the ability of the corresponding trained model to reproduce the available dynamics. Candidate architectures are generated in the context of the ant colony optimization (ACO) meta-heuristic and model training is performed using particle swarm optimization (PSO). We propose a novel solution construction heuristic for artificial ants, based on growth and preferential attachment, in order to generate candidate structures that adhere to well-known gene network properties. Preliminary results using an artificial network demonstrate the potential of the framework to infer the underlying network architecture to a promising degree of success.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization; J.3 [**Life and Medical Sciences**]: Biology and Genetics—*Systems Biology*

General Terms

Algorithms, Design, Experimentation

Keywords

gene regulatory networks, inference, scale-free, ant colony optimization, particle swarm optimization, swarm intelligence

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. 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.

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.

In this paper, we extend an integrated framework for the reconstruction of gene networks from gene expression time series [7], by proposing a novel approach to restricting the structure search space. This approach exploits certain well-known gene network characteristics and incorporates them to the candidate structure generation procedure.

The rest of the paper is organized as follows. In section 2, we provide a brief overview of relevant methods and challenges that have to be addressed. The components of the proposed framework are outlined and discussed in section 3. In section 4, we provide some preliminary results in order to showcase the validity of our approach. Directions for further work are discussed in section 5, which also concludes the paper.

2. BACKGROUND

The problem of reverse-engineering GRNs from gene expression data is a major issue in systems biology [8]. One of the main challenges 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 [4].

In this context, the need for artificial data that have been generated by synthetic gene networks, whose structure is known beforehand, is imperative in order to objectively assess and analyze the degree of success of a reverse-engineering algorithm. This rigorous benchmarking approach is supported by the development of artificial network and data

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...$5.00.

generators, such as AGN [12], SynTReN [15], Gene Net Weaver [11] and others.

A variety of mathematical formalisms have also been proposed for modeling causal relationships between genes and the system's dynamical behaviour, including ordinary and partial differential equations, Boolean networks, Bayesian networks and S-systems among others [3].

An often used representation of causal relationships between genes is a weight matrix W, where the value of each entry w_{ij} captures the strength and nature of the relationship between gene i (regulator) and gene j (target) [18]. A positive value indicates an activatory effect, a negative value indicates a repressive effect, whereas a value of zero indicates the absence of a relationship.

This way, the influence of one or more regulators on a target gene can be modelled as a sigmoid function of the weighted sum of inputs and, in this case, model training consists of optimizing the model's weight matrix and any additional parameters associated with the chosen model, so as to minimize the error between actual and predicted data [17, 6, 19, 10].

The observed sparseness of gene networks [16] implies that most of the weight matrix entries will be zero. Additionally, studies on the structural properties of gene networks have revealed organizational features that are common to other complex networks as well [1], such as power law or exponential degree distributions [5, 9].

3. FRAMEWORK

The proposed framework for network inference adopts an approach that separates between structure selection and model training. The stochastic search process in the structure space yields candidate architectures, whose quality is assessed by the success of the corresponding trained models in reproducing the available dynamics.

Moreover, the vast structure search space can indeed be significantly restricted by exploiting knowledge regarding aforementioned gene network properties that have been reported in the literature.

In section 3.1, we present the model used to express a gene network's dynamical behaviour. Section 3.2 discusses the suggested solution construction heuristic, while section 3.3 outlines the model training process. Section 3.4 presents the ACO inference algorithm by joining the presented framework components together.

3.1 Network Modelling

In general, the structure of a gene network can be represented as a directed graph $G = (V, E)$, where a vertex v_i represents gene i and an edge e_{ij} represents the influence of gene i to gene j.

The dynamics of such a representation can be formalized using a recurrent neural network (RNN) model, where the output x_i of each node i, at time point $t + \Delta t$ is calculated by:

$$x_i(t+\Delta t) = \frac{\Delta t}{T_i} f\Big(\sum_j w_{ji} x_j(t) + b_i\Big) + \Big(1 - \frac{\Delta t}{T_i}\Big) x_i(t) \quad (1)$$

where each synaptic weight w_{ji} expresses the influence of node j to node i, b_i and T_i are the bias term and time constant for node i respectively, and f is a nonlinear transfer function, in this case the logistic function $f = \frac{1}{1+e^{-cx}}$, where c=1.

3.2 Generating Candidate Structures

A model to generate directed graphs, based on the principles of growth and preferential attachment has been proposed in [2] and is capable of producing directed graphs whose in- and out- degree distributions are either exponential or power laws, depending on the model's parameter values. This model specifies a stochastic process according to which a graph grows by adding a single, directed edge at each discrete time step. At each such step, a vertex may also be added to the graph.

We extend this model by considering preferential attachment based not only on node degrees but on edge fitness as well, exploiting the concept that certain edges are more likely or "fit" to be included in the graph than others.

In particular, let $d_{in}(v)$ and $d_{out}(v)$ be the in-degree and out-degree of node v respectively, δ_{in} and δ_{out} non-negative real numbers and τ_{ij} the fitness value of edge e_{ij}. At each step of the generative process, either one of three rules is applied:

A/ with probability α, a link is established from a new node v_i selected according to $\tau_{ij}, \forall j$ to an existing node v_j selected according to $d_{in}(v_j) + \delta_{in}$ and τ_{ij}

B/ with probability β, a link is established from an existing node v_i selected according to $d_{out}(v_i) + \delta_{out}$ and $\tau_{ij}, \forall j$ to an existing node v_j selected according to $d_{in}(v_j) + \delta_{in}$ and τ_{ij}

C/ with probability γ, a link is established from an existing node v_i selected according to $d_{out}(v_i) + \delta_{out}$ and $\tau_{ij}, \forall j$ to a new node v_j selected according to τ_{ij}

with $\alpha + \beta + \gamma = 1$.

The described generative process essentially constitutes the solution construction heuristic that will be used in the context of ACO for navigating the structure search space.

3.3 Model Training

Having obtained a candidate structure, the next step is to assess its quality by considering the corresponding RNN model and optimizing its parameters to fit the model to the available data.

Model parameters are trained using PSO on a per-node basis. In particular, for each node v_i (considered as the target), the parameters that need to be optimized are the bias term b_i, the time constant T_i, as well as the weight values w_{ji} that correspond to its incoming links. Weights corresponding to edges that are not part of the candidate structure are not optimized; their values are set to 0.

The training objective is to minimize the mean squared error of the actual target output profile and the predicted target output profile, as in:

$$\epsilon_i = \frac{1}{T} \sum_{t=1}^{T} (x_i^t - \hat{x}_i^t)^2 \quad (2)$$

where T is the number of time points in the time series, x_i^t is the actual value of node i, at time point t and $\hat{x}_i^t$ its predicted value. One-step-ahead prediction is performed, according to which the actual system state $\hat{X}(t)$ is used to calculate the predicted output value $\hat{x}_i(t+1)$ for each of the network nodes.

3.4 Gene Network Inference

The ACO meta-heuristic serves as the mechanism to connect the structure generation heuristic with model training, in a way so as to guide the search in structure space towards solutions that succeed in reproducing actual gene profile dynamics.

At each ACO iteration, each artificial ant generates a candidate network architecture by supplying the colony's pheromone matrix to the generative process that was described in section 3.2, as the edge fitness matrix. The model that corresponds to the generated structure is trained using PSO as detailed in section 3.3. The result of model training is a vector of mean squared errors for each of the network nodes (targets).

At the end of each ACO iteration, all candidate solutions are structurally decomposed to determine which combination of regulators achieved the lowest prediction error for each target gene. Decomposition results in a local best solution (set of edges) L, for the current iteration. Following a typical pheromone evaporation procedure, the pheromone matrix entries that correspond to the best performing combination of regulators for each target are updated according to:

$$\tau_{ij} \leftarrow \tau_{ij} + \frac{1}{1+\epsilon_j}, \quad \forall (i,j) \in L \tag{3}$$

Local search is also applied to the local best solution, by pruning the edges $e_{i,j}$, for which the corresponding trained RNN weights $|w_{i,j}| < \theta w_{max}$, where $\theta \in [0,1]$ and w_{max} is the maximum allowed weight value.

After the specified number of ACO iterations has been completed, the best solution is a network structure consisting of those combinations of regulators that achieved the minimum prediction error for each node (target) in the network.

4. PRELIMINARY RESULTS

An empirical estimation of the framework's inferential power was attempted, by applying it to a 10-node network (shown in figure 1), that was generated using the process outlined in section 3.2. Three time series with 10 time points each were generated using the corresponding RNN instances with randomly initialized parameters.

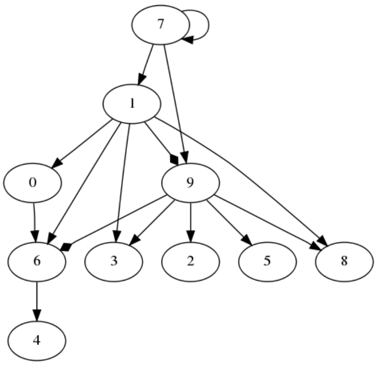

Figure 1: The network that was used for testing the proposed framework. Normal arrow heads denote activation and diamond-shaped arrow heads denote repression.

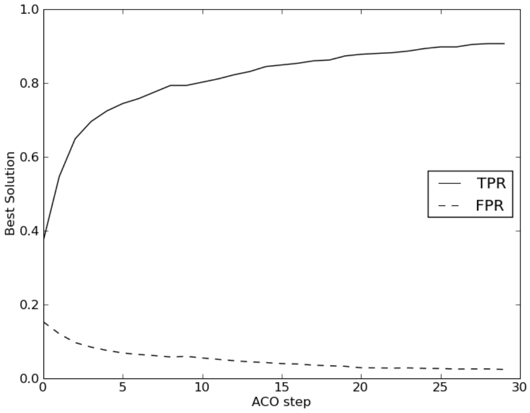

Figure 2: True positive and false positive rates of the best solution during 30 ACO iterations, averaged over 30 independent trials.

A family of N inferred structures was assembled by running $N = 30$ independent ACO trials and recording the best solution achieved at the end of each trial. A simple voting scheme was applied, according to which the score for each edge is calculated as

$$v_{ij} = \frac{f_{ij}}{N} \tag{4}$$

where f_{ij} is the frequency of appearance of edge e_{ij} in the family of inferred structures. Figure 3 presents the scores v_{ij} of all regulatory relationships for the tested network. The average true positive rate of the best solutions across all trials was $TPR = 0.91$, while the corresponding false positive rate was $FPR = 0.02$. The progression of average TPRs and FPRs over 30 ACO iterations is shown in figure 2.

For each of 30 independent ACO trials, a population of 5 artificial ants was allowed to build solutions for 30 ACO iterations, with a pheromone evaporation parameter $\rho = 0.1$ and a local search edge pruning parameter $\theta = 0.05$. For each candidate structure, PSO was allowed to run for 300 steps for the optimization of each target's parameters.

The difficulty in predicting the activatory synergistic influence of nodes 0 and 1 to node 6 is perhaps worth pointing out. As is evident from figure 3, edge $e_{0,6}$ is always inferred whereas edge $e_{1,6}$ is not, in which case the predicted RNN weight $w_{0,6}$ is approximately twice its actual value, in order to account for the missing edge $e_{1,6}$.

5. FURTHER WORK

The proposed network reconstruction framework, incorporating a novel solution construction heuristic, produced promising initial results and demonstrated a potential for further improvement of its inferential power.

More specifically, future work includes experimentation with different GRN models such as S-systems, and publicly available artificial data sets for which the underlying network is known [14], so that comparisons can be drawn with other reverse-engineering approaches.

The proposed solution construction heuristic can be further improved by incorporating additional aspects of struc-

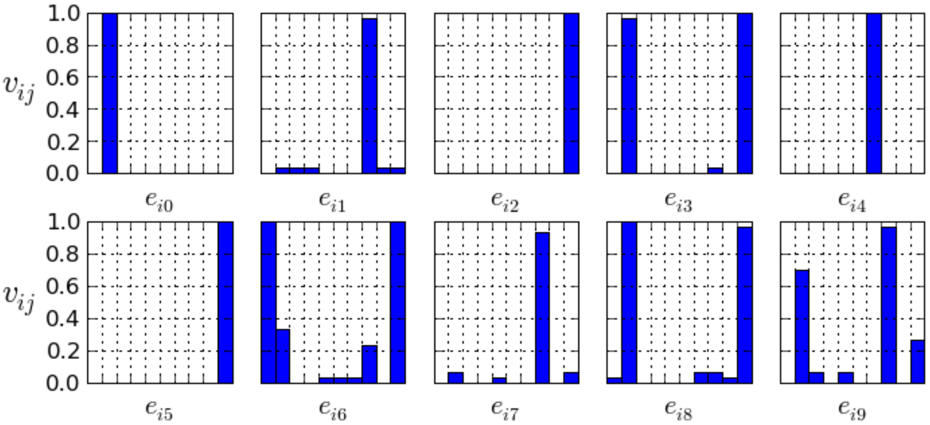

Figure 3: Voting scores v_{ij} for every putative regulatory relationship e_{ij}, where i denotes the regulator node and j the target.

tural network properties besides degree distributions, for example the presence of motifs [13]. Ongoing work also includes an investigation of ACO performance and the impact of measures such as pheromone matrix (re-)initialization to the search process.

6. REFERENCES

[1] A. L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews. Genetics*, 5(2):101–113, 2004.

[2] B. Bollobás, C. Borgs, J. Chayes, and O. Riordan. Directed scale-free graphs. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 132–139, 2003.

[3] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1):69–105, 2002.

[4] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[5] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[6] E. Keedwell and A. Narayanan. Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):231–242, 2005.

[7] K. Kentzoglanakis, M. Poole, and C. Adams. Incorporating heuristics in a swarm intelligence framework for inferring gene regulatory networks from gene expression time series. In *6th International Workshop on Ant Colony Optimization and Swarm Intelligence*, volume 5217 of *Lecture Notes in Computer Science*, pages 323–330. Springer, 2008.

[8] H. Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

[9] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann, and M. Gerstein. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, 431:308–312, 2004.

[10] D. Marbach, C. Mattiussi, and D. Floreano. Replaying the evolutionary tape: Biomimetic reverse engineering of gene networks. *Ann N Y Acad Sci*, 1158:234–245, 2009.

[11] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

[12] P. Mendes, W. Sha, and Y. Keying. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(Suppl. 2), 2003.

[13] R. Milo, Shen S. Orr, S. Itzkovitz, N. Kashtan, D. Chklovski, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[14] G. Stolovitzky, D. Monroe, and A. Califano. Dialogue on reverse-engineering assessment and methods : The dream of high-throughput pathway inference. *Ann N Y Acad Sci*, 1115:1–22, 2007.

[15] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(43), 2006.

[16] A. Wagner. Estimating coarse gene network structure from large-scale gene perturbation data. *Genome Res*, 12:309–315, 2002.

[17] M. Wahde and J. Hertz. Modeling genetic regulatory dynamics in neural development. *Journal of Computational Biology*, 8(4):429–442, 2001.

[18] D. C. Weaver, C. T. Workman, and G. D. Stormo. Modeling regulatory networks with weight matrices. In *Pacific Symposium on Biocomputing*, volume 4, pages 112–123, 1999.

[19] R. Xu, D. C. Wunsch II, and R. L. Frank. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(4):681–692, 2007.