

A Swarm Intelligence Framework for Reconstructing Gene Networks: Searching for Biologically Plausible Architectures

Kyriakos Kentzoglanakis, Matthew Poole

Abstract—In this paper, we investigate the problem of reverse-engineering the topology of gene regulatory networks from temporal gene expression data. We adopt a computational intelligence approach comprising swarm intelligence techniques, namely particle swarm optimization (PSO) and ant colony optimization (ACO). In addition, the recurrent neural network (RNN) formalism is employed for modelling the dynamical behaviour of gene regulatory systems. More specifically, ACO is used for searching the discrete space of network architectures and PSO for searching the corresponding continuous space of RNN model parameters. We propose a novel solution construction process in the context of ACO for generating biologically plausible candidate architectures. The objective is to concentrate the search effort into areas of the structure space that contain architectures which are feasible in terms of their topological resemblance to real-world networks. The proposed framework is initially applied to the reconstruction of a small artificial network that has previously been studied in the context of gene network reverse-engineering. Subsequently, we consider an artificial data set with added noise for reconstructing a subnetwork of the genetic interaction network of *S. cerevisiae* (yeast). Finally, the framework is applied to a real-world data set for reverse-engineering the SOS response system of the bacterium *Escherichia coli*. Results demonstrate the relative advantage of utilizing problem-specific knowledge regarding biologically plausible structural properties of gene networks over conducting a problem-agnostic search in the vast space of network architectures.

Index Terms—Gene regulatory networks, network inference, recurrent neural networks, swarm intelligence, particle swarm optimization, ant colony optimization, degree distribution.



1 INTRODUCTION

WITH the advent of modern experimental technologies, the accumulation of an unprecedented amount of biological data has created the pressing need for the development of computational methods to analyze and interpret such data. This paper is concerned with the problem of reverse-engineering the topology of gene regulatory networks from temporal gene expression data that capture the network's dynamical behaviour. The elucidation of the logic of transcriptional regulation, represented by a gene regulatory network, constitutes a major challenge in the wider context of understanding global cellular behaviour [1].

The topology of a gene regulatory network (GRN) captures a complex web of causal relationships, where connections represent regulatory interactions between genes. In this context, causality refers to the regulation of the gene expression process, i.e. the biochemical chain of events according to which particular genetic code segments (genes) are transformed to executable compounds (proteins). Gene regulatory interactions are indirect in the sense that causal influences are exerted

not by the genes themselves, but rather by specialized protein products (transcription factors). Transcription factors bind to specific regions in the sequence of target genes in order to modify their expression by inducing changes in the rate of production of the corresponding proteins. Depending on the nature of these changes, the regulatory influence of transcription factors (regulators) can be classified either as activation when the rate of protein synthesis increases, or as repression when the rate of protein synthesis decreases.

The global cellular patterns of gene expression can be monitored using microarrays, which can simultaneously quantify the relative abundance of thousands of mRNA transcript species, in the form of gene expression profiles. However, the analysis of gene expression data for the identification of the underlying relationships does not come without important difficulties [2]. First of all, the information contained in a gene expression data set is polluted by considerable amounts of biological and experimental noise. Secondly, the number of genes whose expression levels are measured in the data set is, typically, two to three orders of magnitude greater than the number of observations or time points. The relative insufficiency of observations compared to the number of measured genes constitutes what is often referred to as the “curse of dimensionality” [3].

In this paper, we use a hybrid ACO/PSO system in order to reverse-engineer the topology of a gene regulatory network from temporal data that capture the

- Kyriakos Kentzoglanakis and Matthew Poole are with the School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK
E-mails: kkentzo@nimr.mrc.ac.uk, matthew.poole@port.ac.uk
- Kyriakos Kentzoglanakis is also with the National Institute of Medical Research, Division of Mathematical Biology, London NW7 1AA, UK.

network's dynamical behaviour. The RNN formalism is employed for modelling the network's dynamics. ACO is used for searching the discrete space of network architectures. We formulate a novel solution construction process for artificial ants which extends a stochastic graph generation model proposed by Bollobás et al. [4]. This process yields candidate structures that adhere to a specific topological property, which characterizes the degree distributions that have been observed in numerous empirical studies on the topology of biological networks. This way, the vast structure space can be significantly restricted by considering only feasible architectures with respect to the aforementioned property. The quality of each candidate architecture is evaluated by training the parameters of the corresponding RNN model using PSO, so as to minimize the model's prediction error.

The rest of the paper is organized as follows. Section 2 presents the relevant background and an overview of existing approaches to the gene network inference problem. Section 3 presents the proposed ACO/PSO framework along with a detailed description of the novel method of generating candidate network architectures in the context of ACO. Finally, Section 4 discusses the experimental application of the framework to artificial and real-world time-course data sets, before a conclusion is reached in Section 5.

2 BACKGROUND

A large group of approaches to gene expression data analysis can be characterized by the use of various pair-wise similarity metrics between individual patterns of gene expression. The earliest such attempts consisted of clustering temporal gene expression patterns according to pair-wise correlation coefficients [5] and euclidean distances [6]. Information-theoretic methods have also been proposed, where mutual information is used to define the similarity between pairs of gene expression patterns [7], [8], [9]. In addition, the use of Bayesian networks also constitutes a popular approach as a means of modelling the statistical dependencies between gene expression patterns [10], [11], [12], [13].

A variety of mathematical formalisms (dynamical models) that encapsulate both the system's structure and dynamics have also been used in order to model, simulate and predict the dynamical behaviour of gene regulatory networks [14]. Such formalisms range from boolean networks, where gene expression levels are modelled as binary variables [15], to more detailed biochemical models such as the power-law formalism (S-system) [16].

The dynamical behaviour of gene networks can also be represented by additive regulation models [17], where the combined causal influence of a set of regulators to a target is expressed as the weighted sum of their expression levels. In this case, the strength and nature of the causal interaction between gene i (target) and gene j (regulator) is expressed as a real number w_{ij} . A positive value of w_{ij} denotes an activatory relationship (gene j

activates gene i), a negative value denotes a repressive relationship (gene j represses gene i) and a zero value signifies the absence of a causal relationship between the two genes. This way, the structure of a gene network can be described by a *weight matrix* $W = [w_{ij}]_{N \times N}$, where N is the number of genes in the network [18], [19].

A frequently used model for describing the dynamics of such a system is the recurrent neural network (RNN) [20], [21], [22], [23], [24], [25], [26], whereby the expression level x_i of the i^{th} gene varies temporally according to:

$$x_i(t + \Delta t) = \frac{\Delta t}{c_i} f\left(\sum_{j=1}^N w_{ij} x_j(t) + b_i\right) + \left(1 - \frac{\Delta t}{c_i}\right) x_i(t) \quad (1)$$

where b_i is a bias term that can be interpreted as the basal expression level of the i^{th} gene and c_i is a time constant that acts as a scaling factor. Function f is a sigmoidal function such as the logistic function $f(x) = \frac{1}{1 + e^{-x}}$.

The parametric modelling of the dynamical behaviour of gene networks using appropriate mathematical formalisms (dynamical models) such as boolean networks, S-systems or RNNs suggests the potential to uncover the underlying causal relationships as captured in the dynamical behaviour of the regulatory system. In this context, the objective of a model-driven reverse-engineering strategy is the estimation of the model's parameter values so as to reproduce the available time series. In essence, this objective constitutes an optimization problem in which the parameters of the model are estimated (trained) so as to minimize the error between the actual and simulated time series. Formally, the objective is to infer a model instance $p_c \in P$, comprising a set of parameter values, such that:

$$f(p_c) = \min\{f(p)\}_{p \in P} \quad (2)$$

where P is the set of all model instances. Function f calculates a measure of the deviation between the actual and simulated time series, such as the mean squared error (MSE):

$$f(p) = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T (x_i(t) - \hat{x}_i(t))^2 \quad (3)$$

where N is the number of genes measured, T is the number of time points in the time series and $x_i(t)$ and $\hat{x}_i(t)$ are the actual (observed) and simulated (predicted) expression levels of the i^{th} gene at time point t respectively.

However, as a consequence of the curse of dimensionality, the system is severely underdetermined and there exists no unique solution p_c to fit the available data. One approach to addressing the dimensionality problem is the application of clustering, whereby genes are substituted by gene clusters and individual temporal expression patterns by average cluster temporal patterns [19], [21], [23], [24], [27], [28]. Despite the advantage

of the significant reduction in the number of model parameters that have to be trained, the underlying transcriptional regulation network is replaced by a network of gene clusters and subsequent inferential efforts can only identify causal relationships between these clusters rather than individual genes.

A variety of stochastic search methods, such as simulated annealing [21], [29], genetic algorithms [23], [24], [28], [30], [31] and particle swarm optimization [25], [32], have been applied to the parameter estimation problem. However, the quadratic scaling of the number of model parameters with respect to the number of genes measured in the time series constitutes a serious impediment to keeping the problem dimensionality within feasible bounds. As such, global parameter estimation is impossible for realistic values of N , as in the order of hundreds or thousands of genes. For this reason, problem decomposition strategies have been proposed as a means to splitting the global problem of estimating the full set of model parameters to several local sub-problems, each corresponding to estimating the parameters associated with a single gene [28], [31], [33], [34].

In addition to the problem decomposition strategy, there exists a safe assumption regarding the topology of gene networks that can be made in order to further reduce the number of model parameters to be trained: gene networks are sparse (loosely interconnected) since most genes are regulated by only a handful of other genes [2]. This implies that most of the model's parameters that represent pair-wise regulatory relationships, such as weight matrix entries in RNNs or kinetic order exponents in S-systems, can be set to zero and ignored during model training. One approach to incorporating the sparseness property to model training is the gradual optimization strategy, according to which parameters with small estimated values are set to zero and the model is optimized again with reduced dimensionality [18], [23], [24], [29], [30], [35]. Alternatively, a penalty term can be added to the fitness function represented by Equation 3 with the objective of eliminating "futile" parameters [21], [28], [30], [34].

However, the necessity for the development of an appropriate environment within which topological restrictions of any kind can be enforced suggests the possibility of distinguishing between the structural aspect of the problem and the dynamical aspect of the problem. This possibility can be realized by implementing a decoupling strategy whereby the discrete space of network architectures is separated from the continuous space of model parameters [25], [26], [32], [36], [37]. The search in the discrete network structure space is guided by the search in the corresponding continuous space of model parameters, as the quality of each candidate architecture is assessed by the degree of success of the corresponding trained dynamical model in reproducing the available time series. The degree of success is expressed by the minimum achieved prediction error of the trained dynamical model. The introduction of a

combinatorial aspect to the problem, reflected on the need to generate candidate architectures for searching the discrete space of network structures, is counterbalanced by the reduction in the dimensionality of the corresponding parameter estimation problem.

One of the earliest approaches to imposing structural restrictions on candidate network architectures was the introduction of a parameter k_0 that represents the allowed number of regulatory inputs per gene [38], [39]. The combinatorial task of selecting $k \leq k_0$ inputs for each gene in the network can be performed either by exhaustive enumeration [40], [41], [42] or by sampling the discrete space of k -combinations of inputs using stochastic search methods [36], [37], [43].

However, there exists evidence of observed properties concerning the global topology of biological networks which can inspire the formulation of additional topological restrictions to candidate network architectures. In particular, empirical studies on the statistical characterization of the structural properties of biological networks have demonstrated that such networks have degree distributions which appear to be best approximated by power laws of the form $P(k) \sim k^{-\gamma}$ [44], [45], [46], [47]. The exponent γ specifies the slope of the linear relationship between the degree k and $P(k)$ on a logarithmic scale. Other empirical studies that have considered directed graph representations of the transcriptional network of *S. cerevisiae* (yeast) reported that the distribution of in-degrees can be best approximated using exponential laws of the form $P(k) \sim e^{-\beta k}$, while the distribution of out-degrees is best approximated by power-laws [48], [49].

In general, a power law distribution of node degrees suggests the heterogeneous nature of the graph's connectivity. This is in contrast to bell-shaped (e.g. Poisson) distributions where the connectivity is homogeneous and the concept of a typical node can be defined with respect to the graph's average degree $\langle k \rangle$. The heterogeneity of the graph's connectivity implies the existence of a large number of nodes that have few links and a small number of highly-connected nodes, commonly termed hubs. Graphs with power-law degree distributions are often referred to as *scale-free* graphs [50] to denote the scale invariance of the degree distribution.

Empirical studies on the topology of biological networks, set in the wider context of complex networks, have indeed produced a body of evidence in support of the ubiquity of the scale-free property. However, the observed scale-free nature of biological networks may be due to introduced biases in hand-crafted pathway maps [51]. Such biases concern functionally important genes which happen to have been popular targets for research. The empirical and heuristic nature of scale-free studies has also prompted criticism by graph theorists regarding the lack of "rigorous mathematical work" [52]. Further criticism refers to methodological decisions with respect to the derivation of the degree distribution from topological data. For example, Tanaka et al. [53] under-

lined the ambiguous nature of frequency-degree plots and showed that the use of rank-degree plots results in distributions that can often be best approximated by exponential rather than power laws. Nevertheless, the empirical observations of the power-law degree distribution in biological networks are valuable since they can be exploited for the specification of biologically plausible topological restrictions to be enforced on candidate network architectures.

3 METHODS

The proposed framework for the reconstruction of gene networks from temporal gene expression data comprises the following components:

- the RNN formalism for modelling the dynamical behaviour of the regulatory system;
- ACO for generating biologically plausible candidate architectures; and
- PSO for training the corresponding RNN models.

The rest of this section describes the implementation of the framework's components. Section 3.1 is concerned with the representation of the structure and dynamics of gene networks. Section 3.2 discusses the PSO implementation used for the purpose of model training. Finally, Section 3.3 presents the ACO algorithm used for searching the discrete space of network structures. A novel method for the stochastic generation of biologically plausible candidate architectures in the context of ACO is also presented in detail.

3.1 Gene Network Representation

The structure of a gene network can be represented as a directed graph $G = (V, E)$, where each vertex $v_i \in V$ represents a gene and each edge $e_{ij} \in E$ corresponds to the regulatory influence of gene v_j (regulator) to gene v_i (target). Equivalently, the structure of the network can be represented as an adjacency matrix $M = [m_{ij}]_{N \times N}$, where N is the fixed number of nodes in the network. The binary value of each matrix entry m_{ij} determines whether a directed edge exists from node v_j to node v_i .

The dynamics of the system are expressed using the RNN formalism described by Equation 1 (in Section 2). The complete specification of an RNN instance consists of the weight matrix $W = [w_{ij}]_{N \times N}$, the bias vector $B = [b_i]_{1 \times N}$ and the time constants vector $C = [c_i]_{1 \times N}$. The specified RNN instance is used in order to reproduce the available time series by calculating each predicted (simulated) system state $\hat{x}(t + \Delta t)$ from the previous actual (observed) system state $x(t)$ (one-step ahead prediction).

3.2 Model Training using PSO

The quality of a candidate network architecture, in the form of an adjacency matrix M , is evaluated by estimating the parameters of the corresponding RNN model so as to minimize the error between the actual and

simulated time series. We apply a problem decomposition strategy [33] according to which the global problem of estimating the full set of $N(N + 2)$ RNN parameters, where N is the number of genes in the network, is split to N independent sub-problems, each associated with estimating the parameters of an individual target gene. Furthermore, the weights that correspond to non-existent edges in the network structure under consideration are locked to zero and excluded from the optimization process. More specifically, for the i^{th} sub-problem, the parameters under training include the weights $W_i = \{w_{ij} \mid m_{ij} = 1\}$ that correspond to the incoming connections of gene i , the bias term b_i and the time constant c_i . The objective of parameter estimation for each independent sub-problem i is to minimize the prediction error ϵ_i for the temporal expression pattern of gene i , according to:

$$\epsilon_i = \frac{1}{T} \sum_{t=1}^T (x_i(t) - \hat{x}_i(t))^2 \quad (4)$$

where $x_i(t)$ and $\hat{x}_i(t)$ are the actual and simulated expression levels of gene i at time point t respectively and T is the number of available time points. Eventually, the overall quality of the candidate architecture under consideration is determined by an error vector $\mathcal{E} = [\epsilon_i]_{1 \times N}$, where each component ϵ_i represents the minimum achieved prediction error for the temporal expression pattern of gene i .

Particle swarm optimization [54] is applied separately to each independent sub-problem i for estimating the corresponding model parameters. In general, PSO is a simple, flexible and computationally efficient stochastic optimization technique which has been demonstrated to yield results of the same or better quality compared to genetic algorithms in a variety of problem domains, with the additional advantage of conducting a faster exploration of the search space [55], [56]. In essence, a particle swarm is a collection of candidate solutions (particles) that are represented as points in the search space. Each particle n is characterized by a position vector \vec{x}_n , a velocity vector \vec{v}_n and a vector \vec{p}_n that serves as memory of the best position in terms of fitness that the particle has, thus far, encountered. Particle position vectors encode the RNN parameters associated with the current (i^{th}) sub-problem under consideration, as discussed above. The quality (fitness) of each particle is evaluated using the RNN parameter values encoded in the particle's position vector in order to calculate the prediction error for the temporal expression pattern of gene i , according to Equation 4.

Particles interact by communicating their best position \vec{p}_n to other particles within a neighborhood in order to determine the neighborhood's best position \vec{p}_b . A dynamic neighborhood selection scheme is applied whereby at each PSO step each particle randomly selects $K = 3$ other particles to share its best position [25]. Each particle n moves within the search space by iteratively

updating its velocity and position vectors, attracted by its own best position \vec{p}_n , as well as its neighborhood's best position \vec{p}_b , according to the equations:

$$\begin{aligned} \vec{v}_n(t+1) = & \omega \vec{v}_n(t) + \vec{U}(0, \phi_1) \otimes [\vec{p}_n(t) - \vec{x}_n(t)] \\ & + \vec{U}(0, \phi_2) \otimes [\vec{p}_b(t) - \vec{x}_n(t)] \end{aligned} \quad (5)$$

$$\vec{x}_n(t+1) = \vec{x}_n(t) + \vec{v}_n(t+1) \quad (6)$$

where ω is the inertia weight parameter and ϕ_1 and ϕ_2 are the particle's acceleration coefficients that control the magnitude of stochastic attraction towards \vec{p}_n and \vec{p}_b respectively, with $\phi_1 = \phi_2 = 1.496$ following Clerc and Kennedy's recommendation [57]. Each vector $\vec{U}(0, \phi_i)$ contains random numbers drawn from a uniform distribution in $[0, \phi_i]$. The operator \otimes denotes element-wise multiplication. The inertia weight ω effectively controls the scope of the search, what is frequently referred to as the balance between exploration and exploitation. We apply a linearly decreasing inertia weight adaptation scheme [58], with $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.3$.

3.3 Network Reconstruction using ACO

Inspired by the foraging activities of ants, ant colony optimization [59] is a class of meta-heuristics that provide a generic framework of communication between simple agents (artificial ants), whose task is to construct candidate solutions to the optimization problem under consideration. The activities of artificial ants are coordinated in a decentralized fashion, based on the concept of stigmergy, a form of indirect communication by modifying the environment [60].

The process of constructing candidate solutions combines two sources of information. On one hand, stigmergic information refers to problem-agnostic knowledge regarding the quality (cost) of previously evaluated solutions. Stigmergic information is represented in the form of a pheromone matrix $T = [\tau_{ij}]_{N \times N}$, where N is the number of genes in the network. Each pheromone matrix entry τ_{ij} is associated with a unique solution component, in this case the corresponding directed edge e_{ij} in the network architecture. The pheromone matrix is continuously updated over the course of search in order to reflect accumulated knowledge regarding the quality of generated solutions. On the other hand, heuristic information represents specific knowledge regarding the problem under consideration, which is utilized in the context of generating candidate solutions. Thus, each solution component is associated with a heuristic value η_{ij} which represents the desirability of adding edge e_{ij} to the solution under construction.

The combination of stigmergic and heuristic information for generating candidate solutions is expressed as the probability of adding a solution component to the candidate solution under construction. Section 3.3.1 describes the generative process used by artificial ants to

construct candidate network architectures. Section 3.3.2 presents the precise algorithmic form of the ant system used to infer the underlying network from the available time-course data.

3.3.1 Stochastic Generation of Candidate Solutions

In the direction of meeting the biological plausibility requirement, a particularly appealing prospect is the utilization of existing graph-theoretic approaches to generating topologies that exhibit the scale-free property [61], [62], [63]. In particular, Bollobás et al. [4] proposed a parametric, generative process, the directed scale-free (DSF) model, based on growth and degree-based preferential attachment that yields directed graphs with tunable degree distributions.

We propose an extension to the DSF model that augments the heuristic degree-based preferential attachment principle of the original model, with a stigmergic pheromone-based preferential attachment principle. This way, the ability of a node to attract new links during graph construction depends not only on its degree (degree-based preferential attachment) but also on its fitness defined in terms of the pheromone values of its adjacent edges (fitness-based preferential attachment). This enhancement provides the opportunity to utilize the dynamic stigmergic information which is available in the form of the ant system's pheromone matrix.

More specifically, the proposed extended DSF (eDSF) model describes a stochastic process according to which a graph (network) grows by adding a single directed edge (regulatory relationship) at each discrete time step. At each such step, a new node (gene) may also be connected in the graph. A node u is considered to be new (unconnected) if it has a degree $k(u) = k_{\text{in}}(u) + k_{\text{out}}(u) = 0$. If $k(u) > 0$, node u is considered to be existing (connected). At each step of the stochastic process, *exactly one* of the following three operations is performed:

- A/ a regulatory relationship is established between a new regulator and an existing target
- B/ a regulatory relationship is established between an existing regulator and an existing target
- C/ a regulatory relationship is established between an existing regulator and a new target

Regulatory relationships (edges) are defined in terms of the genes (nodes) they connect. Thus, each of the three model rules comprises two steps: the selection of the regulator gene and the subsequent selection of the target gene on the basis of the chosen regulator. The heuristic, degree-based preferential attachment principle of the DSF model is preserved in the probabilistic selection of a node. A stigmergic, pheromone-based preferential attachment principle is also utilized for node selection, based on the pheromone values of the node's adjacent edges. Specifically, the selection of a regulator gene is performed according to its out-degree (heuristic factor) and the pheromone values of its outgoing edges (stigmergic factor). Subsequently, the target gene is selected according to its in-degree (heuristic factor) and

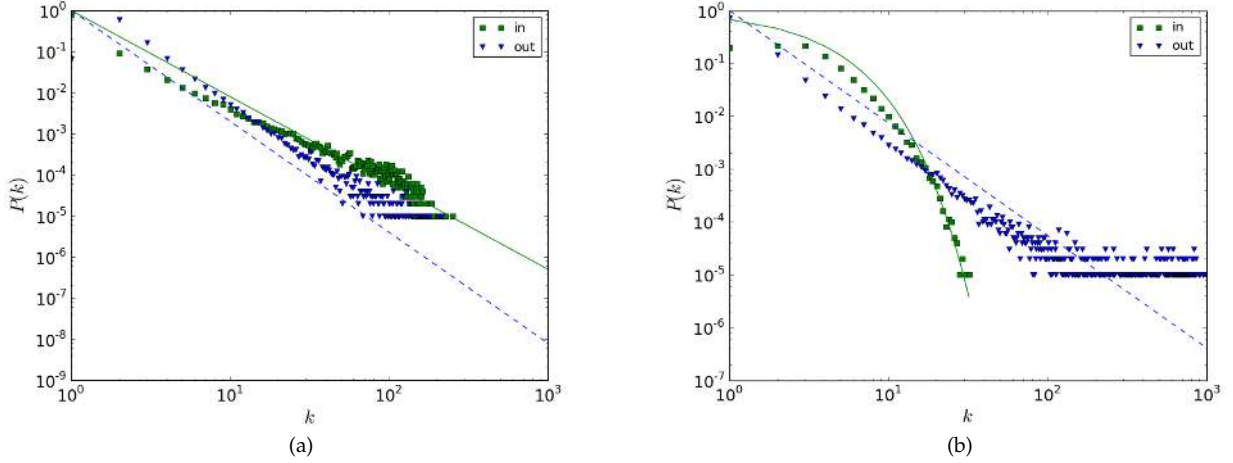


Fig. 1: In- and out-degree distributions averaged across 100 random graphs of 1000 vertices, generated using the DSF model [4]. (1a) Parameter values were set at $\alpha = 0.412$, $\beta = 0.56$, $\gamma = 0.028$, $\delta_{\text{in}} = 0.157$ and $\delta_{\text{out}} = 0$, as in the web graph example from [4]. The degree distributions can be approximated using power laws with exponents $\gamma_{\text{in}} = 2.1$ (solid line) and $\gamma_{\text{out}} = 2.7$ (dashed line). (1b) Parameter values were set at $\alpha = 0.1$, $\beta = 0.6$, $\gamma = 0.3$, $\delta_{\text{in}} = 0.01$ and $\delta_{\text{out}} = 0.05$. The resulting distribution of in-degrees can be approximated by an exponential law (solid line) of the form $P(k) \sim e^{-bk}$, with $b = 0.39$. The distribution of out-degrees follows a power law with exponent $\gamma_{\text{out}} = 2.13$.

the pheromone value of the edge that connects it to the chosen regulator (stigmergic factor).

More formally, let $G(t_0)$ be the initial graph, where t_0 is the number of edges and n_0 the number of nodes, measured in $G(t_0)$. Since a single edge is added at each time step, at time t , the graph $G(t)$ will have exactly t edges, and a random number of nodes $n(t)$.

At each discrete time t , let the indices of, yet, new (unconnected) nodes form the set $\mathcal{N}_{\text{new}}(t)$ and the indices of existing (connected) nodes form the set $\mathcal{N}_{\text{old}}(t)$, with $\mathcal{N}_{\text{new}}(t) \cap \mathcal{N}_{\text{old}}(t) = \emptyset$, $\mathcal{N}_{\text{new}}(t) \cup \mathcal{N}_{\text{old}}(t) = \mathcal{N}(t)$ (the set of all nodes indices) and $|\mathcal{N}_{\text{old}}(t)| = n(t)$.

Furthermore, let $\alpha, \beta, \gamma, \delta_{\text{in}}$ and δ_{out} be non-negative, real numbers, with $\alpha + \beta + \gamma = 1$ and $\alpha + \gamma > 0$. Let also $\tau_{ij} \in \mathbb{R}^+$ be the pheromone value associated with edge e_{ij} . In order to obtain $G(t+1)$ from $G(t)$, exactly one of the following operations is performed:

- A/ with probability α , an edge is added from a new node u to an existing node w . Node u is selected from the pool of unconnected nodes according to the pheromone values corresponding to its outgoing edges, with probability:

$$\Pi(u = u_j) = \frac{\sum_i \tau_{ij}}{\sum_{\kappa} \sum_i \tau_{i\kappa}}$$

where $i \in \mathcal{N}_{\text{old}}(t)$ and $j, \kappa \in \mathcal{N}_{\text{new}}(t)$. Having chosen node $u = u_j$, node w is selected from the pool of connected nodes according to $k_{\text{in}} + \delta_{\text{in}}$ and the pheromone value corresponding to its incoming edge from node u_j , with probability:

$$\Pi(w = w_i | u = u_j) = \frac{[k_{\text{in}}(w_i) + \delta_{\text{in}}][\tau_{ij}]}{\sum_{\kappa} [k_{\text{in}}(w_{\kappa}) + \delta_{\text{in}}][\tau_{i\kappa}]}$$

where $i, \kappa \in \mathcal{N}_{\text{old}}(t)$ and $j \in \mathcal{N}_{\text{new}}(t)$.

- B/ with probability β , an edge is added from an existing node u to an existing node w . Node u is selected from the pool of connected nodes according to $k_{\text{out}} + \delta_{\text{out}}$ and the pheromone values corresponding to its outgoing edges, with probability:

$$\Pi(u = u_j) = \frac{[k_{\text{out}}(u_j) + \delta_{\text{out}}][\sum_i \tau_{ij}]}{\sum_{\kappa} [k_{\text{out}}(u_{\kappa}) + \delta_{\text{out}}][\sum_i \tau_{i\kappa}]}$$

where $i, j, \kappa \in \mathcal{N}_{\text{old}}(t)$. Having chosen node $u = u_j$, node w is selected from the pool of connected nodes according to $k_{\text{in}} + \delta_{\text{in}}$ and the pheromone value corresponding to its incoming edge from node u_j , with probability:

$$\Pi(w = w_i | u = u_j) = \frac{[k_{\text{in}}(w_i) + \delta_{\text{in}}][\tau_{ij}]}{\sum_{\kappa} [k_{\text{in}}(w_{\kappa}) + \delta_{\text{in}}][\tau_{i\kappa}]}$$

where $i, j, \kappa \in \mathcal{N}_{\text{old}}(t)$.

- C/ with probability γ , a new edge is added from an existing node u to a new node w . Node u is selected from the pool of connected nodes according to $k_{\text{out}} + \delta_{\text{out}}$ and the pheromone values corresponding to its outgoing edges, with probability:

$$\Pi(u = u_j) = \frac{[k_{\text{out}}(u_j) + \delta_{\text{out}}][\sum_i \tau_{ij}]}{\sum_{\kappa} [k_{\text{out}}(u_{\kappa}) + \delta_{\text{out}}][\sum_i \tau_{i\kappa}]}$$

where $i \in \mathcal{N}_{\text{new}}(t)$ and $j, \kappa \in \mathcal{N}_{\text{old}}(t)$. Having chosen node $u = u_j$, node w is selected from the pool of unconnected nodes according to the pheromone value corresponding to its incoming edge from node u_j , with probability:

$$\Pi(w = w_i | u = u_j) = \frac{\tau_{ij}}{\sum_{\kappa} \tau_{i\kappa}}$$

where $i, \kappa \in \mathcal{N}_{\text{new}}(t)$ and $j \in \mathcal{N}_{\text{old}}(t)$.

In the above rules, $k_{\text{in}}(v)$ is the in-degree and $k_{\text{out}}(v)$ the out-degree of node v , measured in the graph $G(t)$. Index i always enumerates the target genes, while index j always enumerates the regulator genes. It is worth noting that if for all edges e_{ij} , $\tau_{ij} = c$, with $c \in \mathbb{R}^+$, then the selection of any node is equiprobable with respect to the stigmergic information stored in the pheromone matrix. In this case, the eDSF model is equivalent to the original DSF model.

According to the analysis by Bollobás et al. [4], the original DSF model can generate directed graphs whose degree distributions can be predetermined by setting appropriate values for the parameters $\alpha, \beta, \gamma, \delta_{\text{in}}$ and δ_{out} . An empirical investigation of parameter values showed that the degree distributions of the resulting graphs follow either a power law or an exponential. Figure 1 demonstrates the power law and exponential regimes for in- and out- degree distributions across an ensemble of random graphs generated using the DSF model.

As was discussed in Section 2, the topology of gene networks can be characterized by an exponential behaviour with regard to the distribution of in-degrees and a power-law behaviour with regard to the distribution of out-degrees [48], [49]. In this respect, the eDSF model with the empirical parameter values $\alpha = 0.1$, $\beta = 0.6$, $\gamma = 0.3$, $\delta_{\text{in}} = 0.01$ and $\delta_{\text{out}} = 0.05$ can generate directed graphs, the degree distributions of which can be assumed to be fairly consistent with those of directed graphs corresponding to real gene regulatory networks.

3.3.2 Ant Colony Optimization

The eDSF model for generating candidate network architectures by combining stigmergic and heuristic information constitutes ACO's solution construction process in the context of searching the discrete space of network structure. Once a candidate architecture, in the form of an adjacency matrix M , has been generated by an artificial ant, its quality is evaluated by estimating the parameters of the corresponding RNN model so as to minimize the error between the actual and simulated time series, as discussed in Section 3.2. The outcome of evaluating the quality of a candidate architecture is an error vector $\mathcal{E} = [\epsilon_i]_{1 \times N}$, with each component ϵ_i representing the optimizer's degree of success in reproducing the temporal expression pattern of the i^{th} gene.

Over the course of ACO, the algorithm keeps track of the best topology discovered so far (global best solution), as well as the best topology discovered during the current ACO step (local best solution). The local best solution is assembled by examining the generated solutions at the current ACO step, while the global best solution is updated using the assembled local best solution. In both cases, the best topology is determined on a per-gene basis in terms of the achieved prediction errors ϵ_i . As such, the prediction error ϵ_i of each gene is treated independently and the best solution is assembled so that, for each gene, the combination of regulators

that produced the best prediction error is identified and recorded.

More specifically, at each ACO step, k artificial ants generate k candidate architectures, each represented by an adjacency matrix M_k . The quality of each candidate architecture M_k is determined by the error vector \mathcal{E}_k obtained by estimating the parameters of the RNN instance that corresponds to the topology M_k . At the end of each ACO step, all candidate solutions (M_k, \mathcal{E}_k) are examined in order to discover for each gene i the combination of regulators which achieved the lowest prediction error ϵ_i . These combinations are stored in an adjacency matrix $\widehat{M} = [\widehat{m}_{ij}]_{N \times N}$ which represents the (local) best solution for the current ACO step. The corresponding error vector $\widehat{\mathcal{E}} = [\widehat{\epsilon}_i]_{1 \times N}$ stores the lowest achieved prediction errors $\widehat{\epsilon}_i$ for each target gene i . The update of the *local* adjacency matrix \widehat{M} and the error vector $\widehat{\mathcal{E}}$ from each candidate solution (M_k, \mathcal{E}_k) of the current ACO step is performed according to Algorithm 1.

Algorithm 1 Update an adjacency matrix $M = [m_{ij}]$ and an error vector $\mathcal{E} = [\epsilon_i]$ with an adjacency matrix $M' = [m'_{ij}]$ and an error vector $\mathcal{E}' = [\epsilon'_i]$

```

for each target  $i$  do
  if  $\epsilon'_i < \epsilon_i$  then
    for each regulator  $j$  do
       $m_{ij} \leftarrow m'_{ij}$ 
    end for
  end if
end for

```

Besides storing the best topology in the local context of each ACO step, the algorithm also keeps track of the (global) best topology achieved so far over the course of the search. For this purpose, the global adjacency matrix $\widetilde{M} = [\widetilde{m}_{ij}]_{N \times N}$ along with the associated error vector $\widetilde{\mathcal{E}} = [\widetilde{\epsilon}_i]_{1 \times N}$ are updated after the end of each ACO step using local information from \widehat{M} and $\widehat{\mathcal{E}}$ again according to Algorithm 1.

The topology \widehat{M} and the associated error vector $\widehat{\mathcal{E}}$ reflect the best *local* information that has been gathered at the end of each ACO step. This information is used in order to update the colony's artificial pheromone matrix $T = [\tau_{ij}]$ at the end of each ACO step. Subsequently, the pheromone matrix is also updated using the global best topology \widetilde{M} with the error vector $\widetilde{\mathcal{E}}$ so as to reinforce the impact of *global* information that has been accumulated over all ACO steps. The process of updating the pheromone matrix T using an adjacency matrix M and an error vector \mathcal{E} is outlined in Algorithm 2.

Algorithm 2 Update the pheromone matrix $T = [\tau_{ij}]$ using an adjacency matrix $M = [m_{ij}]$ and an error vector $\mathcal{E} = [\epsilon_i]$.

```

for each target  $i$  do
  for each regulator  $j$  do
    if  $m_{ij} = 1$  then
       $\tau_{ij} \leftarrow \tau_{ij} + \log \epsilon_i / (\log \epsilon_i - 1)$ 
    end if
  end for
end for

```

The last action performed before the initiation of the next ACO step is a simulation of the physical process of pheromone evaporation. The motivation behind the evaporation of the pheromone matrix is to promote the exploration of the search space by avoiding the premature convergence of the search process to a suboptimal solution. For this reason, the value of each pheromone matrix entry is recalculated according to Algorithm 3 using the pheromone evaporation rate $\rho \in [0, 1]$.

Algorithm 3 Perform evaporation of the pheromone matrix $T = [\tau_{ij}]$ using the evaporation rate ρ

```

for each target  $i$  do
  for each regulator  $j$  do
     $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}$ 
  end for
end for

```

At the end of all ACO steps, the best discovered solution is represented by the global adjacency matrix \tilde{M} . This topology is optimal in terms of the minimum achieved prediction errors per gene, stored in the associated error vector $\tilde{\mathcal{E}}$. The precise form of the ACO algorithm is outlined in Algorithm 4.

Algorithm 4 ACO algorithm

```

Initialize pheromone matrix  $T = [\tau_{ij}]_{N \times N}$ 
Initialize global adjacency matrix  $\tilde{M} = [0]_{N \times N}$ 
Initialize global error vector  $\tilde{\mathcal{E}} = [\infty]_{1 \times N}$ 
for each ACO step do
  Initialize local adjacency matrix  $\hat{M} = [0]_{N \times N}$ 
  Initialize local error vector  $\hat{\mathcal{E}} = [\infty]_{1 \times N}$ 
  for each artificial ant  $k$  do
    Generate candidate architecture with adjacency matrix  $M_k$ 
    Obtain error vector  $\mathcal{E}_k$  for  $M_k$  (parameter estimation)
    Update  $\hat{M}, \hat{\mathcal{E}}$  with  $M_k, \mathcal{E}_k$  (see Algorithm 1)
  end for
  Update pheromone matrix  $T$  with  $\hat{M}, \hat{\mathcal{E}}$  (see Algorithm 2)
  Update  $\tilde{M}, \tilde{\mathcal{E}}$  with  $\hat{M}, \hat{\mathcal{E}}$  (see Algorithm 1)
  Update pheromone matrix  $T$  with  $\tilde{M}, \tilde{\mathcal{E}}$  (see Algorithm 2)
  Perform pheromone evaporation (see Algorithm 3)
end for
return solution  $(\tilde{M}, \tilde{\mathcal{E}})$ 

```

4 EXPERIMENTS AND RESULTS

The stochastic nature of the employed optimization techniques implies that, for a given temporal data set, the inferred topologies produced by running a number of independent ACO trials will be expected to demonstrate structural differences. For this purpose, an ensemble learning strategy is implemented according to which the space of inferred networks is sampled so as to assemble a family of L inferred topologies, by running L independent ACO trials and recording the best solution represented by the adjacency matrix \tilde{M}_l at the end of the l^{th} trial. Having assembled L adjacency matrices, a voting scheme is applied with the assignment of a score v_{ij} to each edge e_{ij} according to:

$$v_{ij} = \frac{1}{L} \sum_{l=0}^{L-1} \tilde{m}_{ij}^l \quad (7)$$

with $\tilde{m}_{ij}^l \in \tilde{M}_l$ and $v_{ij} \in [0, 1]$. After calculating the scores for all edges e_{ij} , the final inferred network can be represented in the form of an adjacency matrix $\tilde{M} = [\tilde{m}_{ij}]_{N \times N}$, where the value of each entry $\tilde{m}_{ij} \in \tilde{M}$ is determined according to:

$$\tilde{m}_{ij} = \begin{cases} 1, & \text{if } v_{ij} \geq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

and σ is the threshold for the inclusion of an edge to the final inferred network, with $\sigma \in [0, 1]$.

In order to evaluate the inferential power of the framework, the predicted (inferred) network topology represented by the adjacency matrix \tilde{M} is compared to the actual (true) network topology represented by the adjacency matrix M for the purpose of validating the inferred regulatory interactions. An edge e_{ij} can be classified as either a true positive (TP), a false positive (FP), a true negative (TN) or a false negative (FN), on the basis of comparing the values of $\tilde{m}_{ij} \in \tilde{M}$ and $m_{ij} \in M$.

The *true positive rate* (TPR) of the predicted network, also referred to as *sensitivity* or *recall*, is defined as the fraction of the number of edges that were correctly inferred (TP) over the total number of edges that should have been inferred (TP and FN), according to $\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$. The *false positive rate* (FPR) of the predicted network, also referred to as the *complementary specificity*, is defined as the fraction of the number of edges that were falsely inferred (FP) over the total number of edges that should not be inferred (FP and TN), according to $\text{FPR} = \text{FP}/(\text{FP} + \text{TN})$. Finally, the *positive predictive value* (PPV) of the predicted network, also referred to as *precision*, captures the rate of correctly inferred edges (TP) out of all inferred edges (TP and FP), according to $\text{PPV} = \text{TP}/(\text{TP} + \text{FP})$.

The ACO/PSO framework is first applied to the problem of reconstructing gene networks using artificial data sets with results presented in Section 4.1. Subsequently, the framework is applied to a real-world time-course data set which captures the dynamical behaviour of the SOS response system of *E. coli*, with results presented in Section 4.2.

4.1 Inference using artificial data

In order to assess the relative value of utilizing heuristic information in the context of the proposed eDSF model, an alternative stochastic graph construction model in ACO is also considered. The alternative pheromone-based (PHERO) model ignores heuristic, problem-specific information and operates solely on the basis of stigmergic information as represented by the ant colony's pheromone matrix T . In this respect, it is conceptually analogous to the purely stigmergic ACO implementation of Resson et al. [32]. The PHERO model

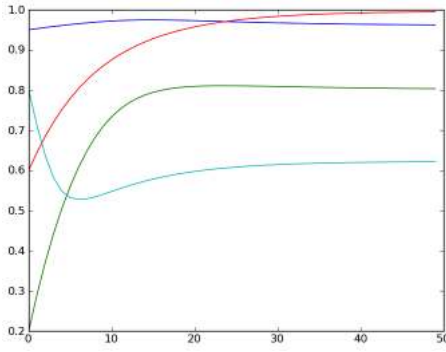


Fig. 2: Training network dynamics for the 4-gene artificial network of Section 4.1.1.

generates sparse candidate solutions by considering the probabilistic addition of each edge e_{ij} to the graph under construction according to the corresponding pheromone trail τ_{ij} . Specifically, for each target gene i , all possible regulators j are considered and the probability of adding each edge e_{ij} to the graph under construction is calculated according to:

$$p_{ij} = \frac{\tau_{ij}}{\sum_j \tau_{ij}} \quad (9)$$

The comparison between the results using the eDSF solution construction model and the results using the PHERO solution construction model will constitute an assessment of the relative advantage of utilizing problem-specific, heuristic information (eDSF model) over performing a problem-agnostic search in the space of network structures (PHERO model).

We begin by considering the reconstruction of a small artificial network in Section 4.1.1, followed by experiments with a subnetwork of the genetic interaction network of *S. cerevisiae* (yeast) using artificial gene expression data with added noise in Section 4.1.2

4.1.1 Reconstructing a small artificial network

Our initial experiment concerns the reconstruction of a small artificial gene network comprising 4 nodes and 8 edges, that has previously been studied in the context of gene network reconstruction [23], [25]. The network dynamics were generated using an RNN model, whose parameter values are displayed in Table 1. The training dynamics (see Figure 2) comprise 50 time points and were assembled by evenly sampling from a time series of 300 time points that was generated by setting $\Delta t = 0.1$ in Equation 1.

The reconstruction experiment consisted of $L = 10$ independent ACO trials of 50 steps each using a population of 5 artificial ants, for both PHERO and eDSF solution construction models. The statistical properties of the resulting network structures at the end of each ACO trial are displayed in Figure 3. These results are consistent with those reported by Xu et al. [25] with regard to the true positive count, when considering single time series. The false positive count is much lower

TABLE 1: RNN model parameters

w_{ij}				b_i	c_i
20	-20	0	0	0	10
15	-10	0	0	-5	5
0	-8	12	0	0	5
0	0	8	-12	0	5

The parameters of the RNN model used to generate the dynamics of the artificial network of Section 4.1.1
Parameter values from [23], [25]

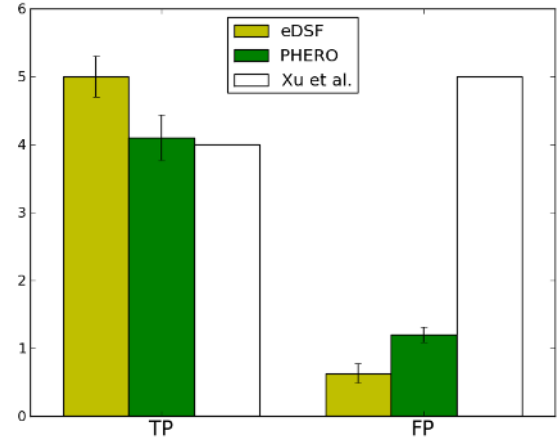


Fig. 3: The statistical properties (true and false positives) of the solutions acquired from $L = 10$ independent ACO trials, for each experimental setting (eDSF and PHERO) of Section 4.1.1. The error bars correspond to the standard error of the sample. The white bars represent the results of Xu et al. [25] on the same network using the same dynamics (Figure 2).

in comparison to Xu et al. in both the eDSF and PHERO experiments, with the eDSF model yielding the best results overall.

With regard to the nature of the inferred relationships, further experiments were conducted with the RNNs that corresponded to the final inferred topologies \tilde{M} , in order to determine the degree to which they succeeded in recovering the nature of the underlying regulatory relationships. In particular, for each experiment (eDSF and PHERO), we trained 100 RNN instances that corresponded to the inferred topology \tilde{M} (with an inclusion threshold $\sigma = 0.5$) using PSO. For each trained RNN instance, we calculated the proportion of true positive edges whose corresponding RNN weight sign (+ or -) matched the nature of the relationship (activatory or repressive) in the actual network (see Table 1). Both experiments succeeded in recovering the correct nature of approximately 65% of the true positive relationships (65% \pm 2% for eDSF and 63% \pm 3% for PHERO; the error terms refer to the sample's standard error).

4.1.2 Reconstructing a real-world network

We proceed by considering the reconstruction of a real-world network using artificial data with added noise in order to compare the performance of the eDSF and PHERO models in the context of ACO/PSO. We used GeneNetWeaver (GNW) [64], an open source software package for the generation of network topologies and

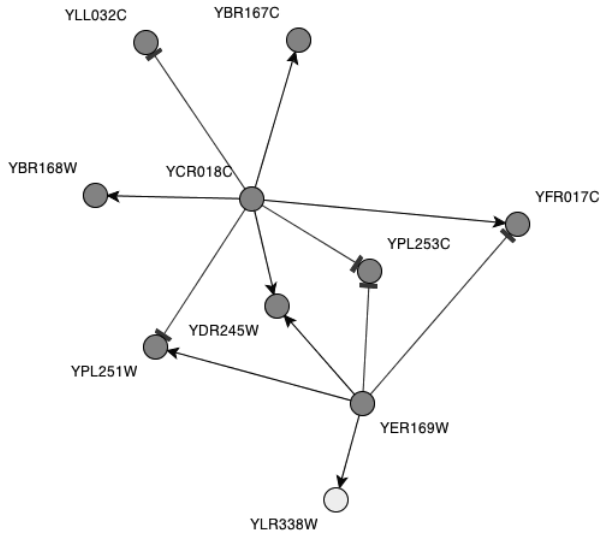


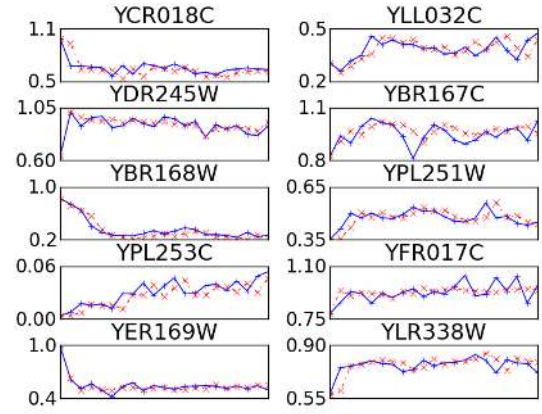
Fig. 4: The topology of the artificial network that was extracted from the yeast genetic interaction network using GNW. Normal arrow heads denote activation, while T-shaped arrow heads denote repression.

in-silico simulation of their dynamical behaviour. GNW generates network topologies by extracting modules from known biological networks and simulates the system dynamics using differential equation models. The software has been used for performing rigorous and systematic assessments of network inference algorithms in the context of the DREAM network inference challenges, a series of competitions for the reverse-engineering of networks from real and artificial gene expression data sets [65].

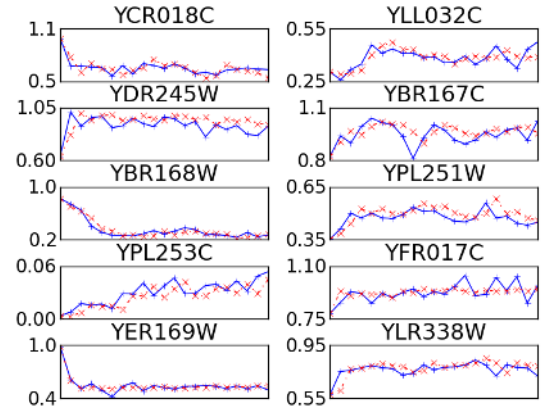
Our target network structure was extracted from the yeast genetic interaction network that is available in GNW. The target network, consisting of 10 nodes and 12 edges, is shown in Figure 4. GNW was also used for generating the dynamics of the target network, which included both additive and synergistic network interactions. We opted for the stochastic differential equation (SDE) model in order to include the effects of noise in the simulated dynamics. We preserved GNW's default noise term coefficient of 0.05, a moderate level that is similar to the simulation settings of the DREAM challenge [66]. The generated dynamics consisted of 21 time points.

For the experiments, the number of independent ACO trials is set to $L = 10$. Each ACO run consists of 100 steps with a population size of 5 artificial ants and a pheromone evaporation rate set to $\rho = 0.1$. The number of iterations for each decomposed sub-problem in PSO is set to 2000, while the swarm population size is calculated by $n_i = \lfloor 10 + 2\sqrt{D_i} \rfloor$. D_i is the dimensionality of the i^{th} sub-problem with $D_i = 2 + \sum_j m_{ij}$, where $m_{ij} \in M$ and M is the adjacency matrix of the network under training.

Table 2 summarizes the characteristics of the final inferred topologies \tilde{M} , assembled using Equation 8 for varying threshold (σ) values. The eDSF model succeeds in discovering a larger proportion of true network edges (TPR), consistently across all threshold values. Figure



(a) eDSF



(b) PHERO

Fig. 5: Actual and predicted time series for the experiments of Section 4.1.2. The predicted dynamics were generated using a trained RNN model that corresponded to the final inferred topology \tilde{M} with a threshold value $\sigma = 0.5$. Actual expression levels have been plotted using solid lines and cross marks, while predicted expression levels using dotted lines and x marks. (a) The eDSF experiment with prediction MSE 2.4×10^{-3} . (b) The PHERO experiment with prediction MSE 2.5×10^{-3} .

5 displays the actual and predicted dynamics for both experiments (eDSF and PHERO), with the respective prediction MSEs being in the same order of magnitude ($\times 10^{-3}$). With regard to the nature (activatory or repressive) of the inferred relationships, we trained 100 RNN instances that corresponded to the inferred topology \tilde{M} (with $\sigma = 0.5$) using PSO, for each experiment (eDSF and PHERO). Both experiments succeeded in recovering the correct nature of approximately 80% of the true positive relationships ($79\% \pm 2\%$ for eDSF and $78\% \pm 2\%$ for PHERO; the error terms refer to the sample's standard error).

The statistical properties of the solution ensembles resulting from L independent ACO trials for each experimental setting are shown in Figure 6. The eDSF model clearly outperforms the PHERO model in terms of the sensitivity (TPR) and the precision (PPV) of the inference

TABLE 2: Characteristics of final inferred networks using ensemble learning

σ	TPR		FPR		PPV		Graph edges	
	eDSF	PHERO	eDSF	PHERO	eDSF	PHERO	eDSF	PHERO
0.4	0.58	0.42	0.24	0.24	0.27	0.21	26	24
0.5	0.50	0.42	0.19	0.22	0.29	0.23	21	22
0.6	0.42	0.25	0.14	0.12	0.31	0.25	16	12
0.7	0.42	0.25	0.14	0.09	0.31	0.30	16	10
0.8	0.42	0.25	0.14	0.09	0.31	0.30	16	10
0.9	0.42	0.25	0.13	0.08	0.33	0.33	15	9
1.0	0.25	0.08	0.08	0.05	0.33	0.20	9	5

Characteristics of the final inferred network topologies \tilde{M} , assembled using Equation 8, for varying levels of the threshold σ . The true (target) network is shown in Figure 4.

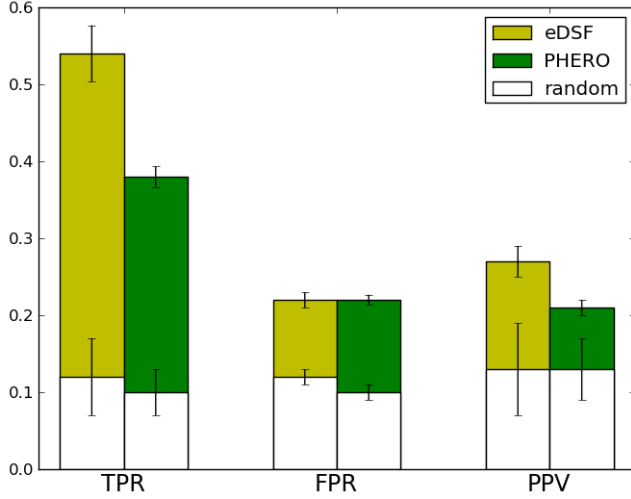


Fig. 6: The statistical properties of the solutions acquired from $L = 10$ independent ACO trials, for each experimental setting (eDSF and PHERO) of Section 4.1.2. The error bars correspond to the standard error of the sample. The *random* values were determined by running the corresponding model (eDSF or PHERO) in order to generate 10,000 random networks of the same size (number of nodes) with the target network, using pheromone matrices with identical entries $\tau_{ij} = 1$.

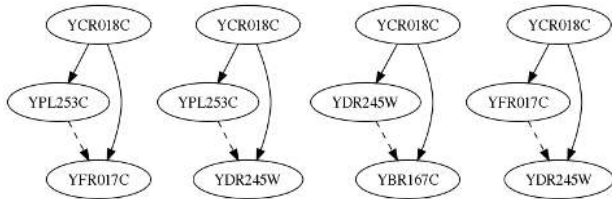


Fig. 7: A commonly recurring structural pattern is demonstrated in four gene triplets extracted from the inferred network topology \tilde{M} , with $\sigma = 0.8$. In the examples shown above, the coregulation of the two target genes by YCR018C has been correctly inferred. However, a spurious edge between the coregulated genes has also been (falsely) inferred. Solid lines correspond to correctly inferred (TP) edges and dashed lines to incorrectly inferred (FP) edges.

results, with the false positive rates being the same across the two models.

A recent study on the comparative assessment of the performance of various network inference methods in the context of the latest DREAM challenge [66] identified certain types of systematic structural prediction errors by performing network motif analysis in the predicted topologies. Our inferred networks also exhibit multiple

instances of one such prediction error type (fan-out error). An investigation of the network topologies \tilde{M} predicted by the ACO/PSO framework reveals that a commonly recurring structural pattern is characterized by a false positive edge connecting a pair of genes that share a common, correctly identified, regulator (gene YCR018C). Figure 7 shows four such examples from the inferred network topology \tilde{M} (with $\sigma = 0.8$) where the correct identification of coregulation in a gene triplet, consisting of one regulator and two target genes, was accompanied by the identification of a spurious edge connecting the two target genes.

4.2 Inference using real-world data

In this section, the ACO/PSO framework, incorporating the eDSF model of generating candidate architectures, is applied to a real-world gene expression data set for the purpose of reconstructing the underlying gene regulatory network. The data set captures the dynamical behaviour of the SOS response system of *E. coli*, a transcriptional network consisting of proteins that are involved in DNA repair activities. SOS response functionality is induced with the detection of sudden increases in DNA damage. There exist about 40 SOS genes [67] involved in DNA repair, the expression of which is regulated by the interplay between two proteins: LexA and RecA. LexA is the master repressor protein of the SOS genes. Under normal conditions, LexA binds to the promoter regions of the SOS genes in order to inhibit their expression. The role of protein RecA is to detect increased levels of DNA damage in the cell. When that happens, RecA switches to an active state and binds to the promoter region of gene *lexA*, thereby mediating the destruction (self-cleavage) of protein LexA. As a result, the SOS genes are de-repressed and the production levels of the corresponding proteins increase in order to address the DNA damage situation. Once DNA damage is repaired or bypassed, the cell returns to its normal state, with the concentration of the activated protein RecA dropping and the transcription factor LexA accumulating in order to re-repress the expression of the SOS genes.

Ronen et al. [68] developed an experimental technique for monitoring the temporal transcriptional activity of SOS response operons using green fluorescent protein (GFP), after initial irradiation of DNA with UV light.

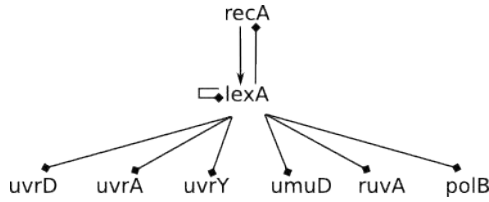


Fig. 8: The structure of the SOS DNA repair transcriptional network of *E. coli* [68]. The displayed relationships express known regulatory interactions between genes. Normal arrow heads denote activation, while diamond-shaped arrow heads denote repression.

TABLE 3: Results from the SOS experiments

Data set	TP	FP	TPR	FPR	PPV
1	3	10	0.38	0.21	0.23
2	8	5	0.89	0.10	0.62
3	4	9	0.50	0.19	0.31
4	0	9	0.00	0.19	0.00

Metric values for the inferred topologies \tilde{M} with a strict threshold value $\sigma = 0.9$ for the four SOS data sets.

Using this method, they obtained time courses capturing the dynamical behaviour of the SOS response system with a temporal resolution in the order of minutes. The transcriptional activity of 8 SOS operons were measured, namely *lexA*, *recA*, *uvrA*, *uvrD*, *uvrY*, *umuD*, *ruvA* and *polB*. Figure 8 displays the known regulatory interactions among the 8 SOS genes measured in the time course experiments. Four experiments were conducted at various UV light intensities, with each experiment consisting of 50 measurements, evenly sampled every 6 minutes¹.

The ACO/PSO framework settings used for inferring the SOS response network from the available gene expression time series were the same as in the artificial data experiments described in Section 4.1.2. The first time point was removed from all four time series, since all expression levels at that point were zero. Subsequently, the temporal gene expression patterns for each gene were normalized in the interval $[0, 1]$. Specifically, the normalized expression level $x'_i(t)$ of gene i at time point t was calculated using:

$$x'_i(t) = \frac{x_i(t) - x_i^{\min}}{x_i^{\max} - x_i^{\min}} \quad (10)$$

where $x_i(t)$ is the actual expression level of gene i at time point t , while x_i^{\min} and x_i^{\max} are the minimum and maximum expression levels of gene i respectively.

The characteristics of the inferred network topologies \tilde{M} for each of the four time series are shown in Table 3. The best prediction was achieved using the second time series, with an inferred topology consisting of 13 edges, 8 of which were true positives and 5 false positives (see Figure 9). The dominance of gene *lexA* was fully recognized with all relationships (with the exception of *recA* \rightarrow *lexA*) being correctly reconstructed. Figure 10 shows

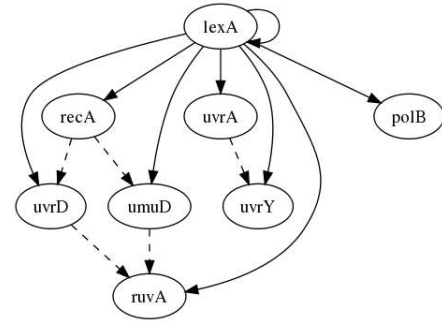


Fig. 9: The predicted topology \tilde{M} of the SOS network, resulting from 10 independent ACO runs, with a strict threshold set at $\sigma = 0.9$. Correctly inferred edges (true positives) have been drawn with solid lines and falsely inferred edges (false positives) with dashed lines.

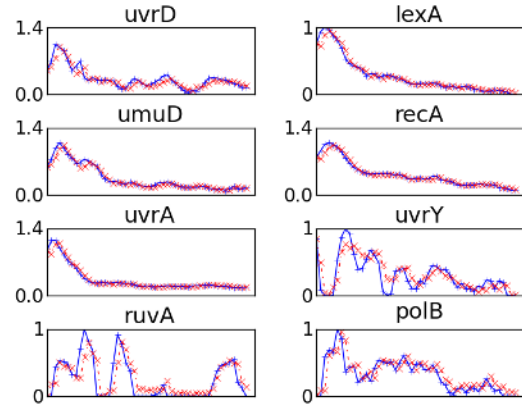


Fig. 10: Actual and predicted dynamics for the SOS experiment, using the second data set. The prediction MSE is 1.2×10^{-2} . The actual dynamics consist of the second time series in the data set of Ronen et al. [68]. The predicted dynamics were generated using a trained RNN model that corresponded to the inferred network topology \tilde{M} , with a strict threshold set at $\sigma = 0.9$. Actual expression levels have been plotted using solid lines and cross marks, while predicted expression levels using dotted lines and x marks.

the actual and predicted dynamics, using a trained RNN model corresponding to the inferred topology \tilde{M} with a threshold value of $\sigma = 0.9$. The prediction MSE is 1.2×10^{-2} .

The differences in the quality of prediction, depending on which data set is used as input to the reverse-engineering framework, may be due to a number of reasons. First, in the second data set, the expression peak of the master repressor *lexA* is smoother and its activity is more sustained than in the other data sets. Second, the expression levels of target genes *uvrY* and *ruvA* appear to be higher in the second data set than in the other data sets. For this reason, a variety of approaches to reconstructing the SOS network using the S-system formalism and the same data sets do not consider the expression profiles of these two genes [69], [70], [71], [72]. However, our method was able to correctly predict the regulation of *uvrY* and *ruvA* by *lexA*, even if their expression was minimally induced by UV light irradiation in the experiments. A detailed comparison of the

1. The four data sets were downloaded from <http://www.weizmann.ac.il/mcb/UriAlon/Papers/SOSData>

TABLE 4: Comparison of predictions for the SOS data set

Known interaction	Predictions by							ACO/PSO
	[12]	[73]	[69] [†]	[25] [‡]	[70] [†]	[71] [†]	[72] [†]	
lexA → lexA	yes	yes	yes	no	yes	yes	yes	yes
lexA → recA	yes	yes	no	yes	yes	yes	yes	yes
recA → lexA	yes	yes	yes	no	yes	yes	yes	no
lexA → uvrA	yes	yes	yes	yes	no	yes	yes	yes
lexA → uvrD	no	no	yes	yes	yes	yes	yes	yes
lexA → uvrY	no	no	–	no	–	–	–	yes
lexA → umuD	no	yes	yes	yes	yes	yes	yes	yes
lexA → ruvA	no	no	–	no	–	–	–	yes
lexA → polB	no	no	yes	yes	yes	yes	yes	yes
Spurious edges (FP)	5	10	6	2	15	16	11	5
Precision (PPV)	0.44	0.33	0.50	0.71	0.29	0.30	0.39	0.62

[†] The profiles of genes uvrY and ruvA were not included in these experiments.

[‡] In addition to this prediction, Xu et al. [25] also report a “less conservative” prediction which includes all nine true relations but more false positives (FP=7), leading to a lower precision value (PPV=0.56).

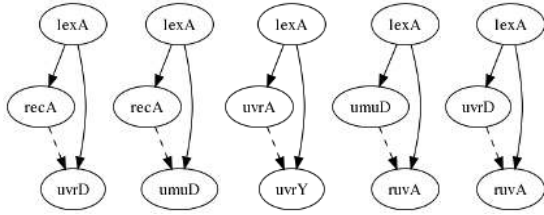


Fig. 11: All five false positive edges in the inferred topology shown in Figure 9 are instances of the fan-out prediction error [66], i.e. the incorrect identification of a link (dashed arrow) between a pair of coregulated genes.

results from different methods (including ours) on the reconstruction of the SOS network using the same data sets is shown in Table 4.

With respect to the inferred topology \tilde{M} , we observed the repeated occurrence of the fan-out prediction error [66], just as in the experiments of Section 4.1.2. In fact, all five incorrectly identified edges of the predicted topology are instances of this type of prediction error as shown in Figure 11. In general, though, the occurrence of spurious edges in the predicted network does not necessarily imply the identification of incorrect relationships. A spurious edge might represent a novel regulatory relationship, whose validity can only be verified experimentally in the laboratory. Alternatively, spurious edges can emerge in the predicted topology as an indirect consequence of the transcriptional activity of external regulators, whose levels of expression have not been measured in the available gene expression data set. For example, there exists evidence for the activation of the uvrY operon by another regulator, gene sdiA, which does not belong to the SOS response network under consideration [74].

5 CONCLUSION AND FURTHER WORK

In this paper, we investigated the problem of reconstructing the structure of gene regulatory networks from time-course data using a decoupled framework based

on swarm intelligence techniques and the RNN formalism. Particular attention was paid to the biological plausibility of candidate solutions in the context of searching the discrete space of network architectures. A novel stochastic process (eDSF model) was formulated for generating candidate architectures that extends a graph generation mechanism proposed by Bollobás et al. [4]. This process yields directed graphs that exhibit topological properties similar to those of real-world biological networks. Experiments demonstrated that the introduction of problem-specific (heuristic) knowledge to the stochastic process of searching the network structure space produced improved results compared to the problem-agnostic approach which operated solely on the basis of ACO’s stigmergic apparatus.

One direction towards strengthening and expanding the conclusions of this paper consists of investigating our framework’s response with respect to increasing network sizes and noise levels. In essence, such an investigation will serve to widen the scope of the artificial data experiments of Section 4.1.2, where we considered only a single network with a noise level which is rather moderate in comparison to real microarray data.

Our implementation of the proposed framework is in a scripted language (Python using the numpy numerical library) and requires approximately 4 hours for a single ACO run of the SOS network on a typical desktop machine running Ubuntu Linux. We are currently finishing an implementation in a compiled language and, at the same time, considering the parallelization capabilities of modern GPUs (using frameworks such as NVidia’s Cuda and OpenCL) in order to be able to handle more realistic network sizes in the future.

In general, the problem of network inference is far from solved. Recent comparative results from the DREAM challenges have shown that a large fraction of methods produce predictions that are, on average, not significantly better than random guessing [66]. Furthermore, the integration of all available sources of data (both steady-state and time series) proved to be a defining characteristic of the best performing methods. The

same results also revealed that the performance of a method that is applied to the reconstruction of a particular network depends strongly on the properties of that network. In other words, caution should be exercised when generalizing with respect to the performance of an inference method using a single network. This serves to underline the need for rigorous testing of reverse-engineering methods on sufficiently large ensembles of diverse target network topologies in order to obtain more accurate statistical estimations of inferential success.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their insightful comments and constructive criticism.

REFERENCES

- [1] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.
- [2] E. P. van Someren, L. F. A. Wessels, E. Backer, and M. J. T. Reinders, "Genetic network modeling," *Pharmacogenomics*, vol. 3, no. 4, pp. 507–525, 2002.
- [3] D. L. Donoho, "High-dimensional data analysis: the curses and blessings of dimensionality," in *American Mathematical Society Conf. Math Challenges of the 21st Century*, 2000.
- [4] 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*, 2003, pp. 132–139.
- [5] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.
- [6] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Barker, and R. Somogyi, "Large-scale temporal gene expression mapping of central nervous system development," *Proceedings of the National Academy of Sciences*, vol. 95, no. 1, pp. 334–339, 1998.
- [7] A. J. Butte and I. S. Kohane, "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements," in *Pacific Symposium on Biocomputing*, 2000, pp. 418–429.
- [8] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, D. R. Favera, and A. Califano, "Reverse-engineering of regulatory networks in human b cells," *Nature Genetics*, vol. 37, no. 4, pp. 382–390, 2005.
- [9] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, D. R. Favera, and A. Califano, "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC Bioinformatics*, vol. 7(Suppl. 1), no. 7, 2006.
- [10] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze gene expression data," *Journal of Computational Biology*, vol. 7, no. 3–4, pp. 601–620, 2000.
- [11] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks," *Bioinformatics*, vol. 19, no. 17, pp. 2271–2282, 2003.
- [12] B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and D. F. Buc, "Gene network inference using dynamic bayesian networks," *Bioinformatics*, vol. 19, no. Suppl. 2, 2003.
- [13] I. Pournara and L. Wernisch, "Reconstruction of gene networks using bayesian learning and manipulation experiments," *Bioinformatics*, vol. 20, no. 17, pp. 2934–2942, 2004.
- [14] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 69–105, 2002.
- [15] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, pp. 437–467, 1969.
- [16] M. A. Savageau, "Power-law formalism: a canonical nonlinear approach to modeling and analysis," in *First World Congress of Nonlinear Analysts '92*, vol. 4, 1995, pp. 3323–3334.
- [17] P. D'Haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.
- [18] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," in *Pacific Symposium on Biocomputing*, vol. 4, 1999, pp. 112–123.
- [19] E. P. van Someren, L. F. A. Wessels, and M. J. T. Reinders, "Linear modeling of genetic networks from experimental data," in *Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 355–366.
- [20] E. Mjolsness, D. H. Sharp, and J. Reinitz, "A connectionist model of development," *Journal of Theoretical Biology*, vol. 152, pp. 429–453, 1991.
- [21] E. Mjolsness, T. Mann, R. Castano, and B. Wold, "From coexpression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale gene expression data," in *Advances in Neural Information Processing Systems*, vol. 12, 2000, pp. 928–934.
- [22] J. Vohradsky, "Neural model of the genetic network," *Journal of Biological Chemistry*, vol. 276, no. 39, pp. 36168–36173, 2001.
- [23] M. Wahde and J. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks," *Biosystems*, vol. 55, no. 1–3, pp. 129–136, 2000.
- [24] —, "Modeling genetic regulatory dynamics in neural development," *Journal of Computational Biology*, vol. 8, no. 4, pp. 429–442, 2001.
- [25] R. Xu, D. C. Wunsch, 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*, vol. 4, no. 4, pp. 681–692, 2007.
- [26] D. Marbach, C. Mattiussi, and D. Floreano, "Replaying the evolutionary tape: Biomimetic reverse engineering of gene networks," *Ann N Y Acad Sci*, vol. 1158, pp. 234–245, 2009.
- [27] Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakola, and I. Simon, "A new approach to analyzing gene expression time series data," in *Sixth Annual International Conference on Computational Biology*, 2002, pp. 39–48.
- [28] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya, "Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm," *Bioinformatics*, vol. 21, no. 7, pp. 1154–1163, 2005.
- [29] O. R. Gonzalez, C. Küper, K. Jung, P. C. Naval, and E. Mendoza, "Parameter estimation using simulated annealing for S-system models of biochemical networks," *Bioinformatics*, vol. 23, no. 4, pp. 480–486, 2007.
- [30] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinformatics*, vol. 19, no. 5, pp. 643–650, 2003.
- [31] S. Y. Ho, C. H. Hsieh, F. C. Yu, and H. L. Huang, "An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 648–660, 2007.
- [32] H. W. Renssom, Y. Zhang, J. Xuan, Y. Wang, and 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, pp. 1–8.
- [33] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, and Y. Eguchi, "Inference of genetic network using the expression profile time course data of mouse P19 cells," *Genome Informatics*, vol. 13, pp. 382–383, 2002.
- [34] N. Noman and H. Iba, "Inferring gene regulatory networks using differential evolution with local search heuristics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 634–647, 2007.
- [35] M. Vilela, I. C. Chou, S. Vinga, T. R. Vasconcelos, V. Voit, and J. S. Almeida, "Parameter optimization in S-system models," *BMC Systems Biology*, vol. 2, no. 35, 2008.
- [36] C. Spieth, F. Streichert, N. Speer, and A. Zell, "Optimizing topology and parameters of gene regulatory network models from time-series experiments," in *Genetic and Evolutionary Computation, GECCO'04*. Springer, 2004, pp. 461–470.

- [37] E. Keedwell and A. Narayanan, "Discovering gene networks with a neural-genetic hybrid," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 3, pp. 231–242, 2005.
- [38] R. Somogyi and C. Sniegowski, "Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation," *Complexity*, vol. 1, pp. 45–63, 1996.
- [39] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," in *Pacific Symposium on Biocomputing*, 1999, pp. 17–28.
- [40] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Pacific Symposium on Biocomputing*, vol. 3, 1998, pp. 18–29.
- [41] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.
- [42] T. S. Gardner, D. Dibernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.
- [43] 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*, ser. Lecture Notes in Computer Science, vol. 5217. Springer, 2008, pp. 323–330.
- [44] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.
- [45] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, pp. 41–42, 2001.
- [46] I. Farkas, H. Jeong, T. Vicsek, A. L. Barabási, and Z. N. Oltvai, "The topology of the transcription regulatory networks in the yeast, *saccharomyces cerevisiae*," *Physica A: Statistical Mechanics and its Applications*, vol. 318, no. 3–4, pp. 601–612, 2003.
- [47] A. H. Tong, et al., "Global mapping of the yeast genetic interaction network," *Science*, vol. 303, no. 5659, pp. 808–813, 2004.
- [48] N. Guelzim, S. Bottani, P. Bourguin, and F. Kepes, "Topological and causal structure of the yeast transcriptional regulatory network," *Nature Genetics*, vol. 31, no. 1, pp. 60–63, 2002.
- [49] 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*, vol. 431, pp. 308–312, 2004.
- [50] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [51] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.
- [52] B. Bollobás and O. Riordan, "Robustness and vulnerability of scale-free random graphs," *Internet Mathematics*, vol. 1, no. 1, pp. 1–35, 2003.
- [53] R. Tanaka, T. Yi, and J. Doyle, "Some protein interaction data do not exhibit power law statistics," *FEBS Letters*, vol. 579, no. 23, pp. 5140–5144, 2005.
- [54] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [55] J. Kennedy and W. M. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," in *IEEE International Conference on Evolutionary Computation (ICEC'98)*, 1998, pp. 78–83.
- [56] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [57] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [58] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*. Springer, 1998, pp. 591–600.
- [59] M. Dorigo, V. Maniezzo, and A. Colnori, "The ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, no. 1, pp. 29–41, 1996.
- [60] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97–116, 1999.
- [61] R. Albert and A. L. Barabási, "Topology of evolving networks: local events and universality," *Physical Review Letters*, vol. 85, no. 24, pp. 5234–5237, 2000.
- [62] P. L. Krapivsky and S. Redner, "Organization of growing random networks," *Physical Review E*, vol. 63, no. 6, 2001.
- [63] G. Bianconi and A. L. Barabási, "Competition and multiscaling in evolving networks," *Europhysics Letters*, vol. 54, no. 4, pp. 436–442, 2001.
- [64] 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*, vol. 16, no. 2, pp. 229–239, 2009.
- [65] 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*, vol. 1115, pp. 1–22, 2007.
- [66] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference," *PNAS*, vol. 107, no. 14, pp. 6286–6291, 2010.
- [67] B. Michel, "After 30 years of study, the bacterial SOS response still surprises us," *PLoS Biology*, vol. 3, no. 7, 2005.
- [68] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10 555–10 560, 2002.
- [69] D.-Y. Cho, K.-H. Cho, and B.-T. Zhang, "Identification of biochemical networks by s-tree based genetic programming," *Bioinformatics*, vol. 22, no. 13, pp. 1631–1640, 2006.
- [70] S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura, and M. Hatakeyama, "Function approximation approach to the inference of reduced nget models of genetic networks," *BMC Bioinformatics*, vol. 9, no. 1, p. 23, 2008.
- [71] S. Kimura, S. Nakayama, and M. Hatakeyama, "Genetic network inference as a series of discrimination tasks," *Bioinformatics*, vol. 25, no. 7, pp. 918–925, 2009.
- [72] M. Kabir, N. Noman, and H. Iba, "Reverse engineering gene regulatory network from microarray data using linear time-variant model," *BMC Bioinformatics*, vol. 11, no. Suppl 1, p. S56, 2010.
- [73] N. Noman and H. Iba, "Reverse engineering genetic networks using evolutionary computation," *Genome Informatics*, vol. 16, no. 2, pp. 205–214, 2005.
- [74] Y. Wei, J. M. Lee, D. R. Smulski, and R. A. LaRossa, "Global impact of sdiA amplification revealed by comprehensive gene expression profiling of *Escherichia coli*," *Journal of Bacteriology*, vol. 183, no. 7, pp. 2265–2272, 2001.

Kyriakos Kentzoglanakis obtained his BSc (Hons) and PhD degrees in Computer Science from the University of Portsmouth. His research interests include nature-inspired computing, swarm intelligence, systems biology and agent-based modelling of evolutionary processes.



Matthew Poole received the BSc (Hons) and PhD degrees in computer science from Swansea University in 1990 and 1995, respectively. He has held positions at the University of Leeds, Swansea University and Coventry University. His research interests lie at the interface between computer science and computational science, with applications in modeling and simulation of physiological systems such as electrical activity in cardiac tissue. Current work also includes the application of computational intelligence techniques to problems in bioinformatics and systems biology. He has been a senior lecturer at the University of Portsmouth since 2003.