# Special Session on Swarm Intelligence

Maite Termenon[1]

[1]Computational Intelligence Group

2012 February 10

# Article to Present

IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 9, NO. 2, MARCH/APRIL 2012

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

Kyriakos Kentzoglanakis and Matthew Poole

# Outline

# Outline

# Description

- RNN for modeling the dynamical behaviour of gene regulatory systems.
- ACO for generating biologically plausible candidate architectures.
- PSO for training the RNN models.

# Difficulties

- Information contained in a gene expression data set is polluted by considerable amounts of biological and experimental noise.
- 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 ("curse of dimensionality").

# Proposal

- A novel solution construction process for artificial ants:
  - It is used for the generation of candidate solutions.
  - It consists of:
    - extending a stochastic graph generation model proposed by Bolloba's et al.
    - adding stigmergic pheromone-based information.

# Outline

# Outline

# Structure of a Gene Network

- Can be represented as a **direct graph**: $G = (V, E)$, where each vertex $v_i \in V$ represents a gene and each edge $e_{ij} \in E$ correspondes to the regulatory influence of genes $v_j$ (regulator) to $v_i$ (target).
- Equivalently, network can be represented as an **adjacency matrix** $M = [m_{ij}]_{NxN}$, where N is the fixed number of nodes and $m_{ij}$ is a binary value that determines whether a directed edge exists from nodes $v_j$ to $v_i$.

# Recurrent Neural Network

- Dynamics of the system are expresed by RNN formalism.
- The expression level $x_i$ of the $i$th gene, varies temporally as:

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

- $b_i$ is the bias term (basal expression level of the $i$th gene).
- $c_i$ is a time constant (scaling factor).
- $f$ is a sigmoidal function (logistic function).

# Specification of a RNN instance

- Consists of:
  - weight matrix $W = [w_{ij}]_{NxN}$
  - bias vector $B = [b_i]_{1xN}$
  - time constant vector $C = [c_i]_{1xN}$

# Outline

# Problem decomposition strategy

- Quality of a candidate network architecture is evaluated by estimating the parameters of the corresponding RNN model that minimize the error between the actual and simulated time series.
- We apply problem decomposition strategy to the global problem of estimating the full set $N(N+2)$ RNN parameters, splitting to N independent subproblems, each associated with estimating the parameters of an individual target gene.

# Problem decomposition strategy

- For the $i$th subproblem, 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 $c_i$.
- Objective: minimize the prediction error $\varepsilon_i$ according to:

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

- 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.

# Problem decomposition strategy

- Quality of the candidate architecture under consideration is determined by an error vector $\mathscr{E} = [\varepsilon_i]_{1xN}$, where $\varepsilon_i$ represents the minimum achieved prediction error for the temporal expression pattern of gene $i$.

# Particle Swarm Optimization

- PSO is applied separately to each independent subproblem $i$ for estimating the corresponding model parameters.
- 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 $\overrightarrow{x}_n$, a velocity vector $\overrightarrow{v}_n$, and a vector $\overrightarrow{p}_n$ that serves as memory of the best position in terms of fitness that the particle has encountered.

# Particle Swarm Optimization

- Particle position vectors encode the RNN parameters associated with the current ($i$th) subproblem under consideration.
- Quality 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 $i$.

# Particle Interaction

- Particles interact by communicating their best position $\overrightarrow{p}_n$ to other particles within a neighborhood to determine the neighborhood's best position $\overrightarrow{p}_b$.
- Each particle randomly selects $K = 3$ particles to share its $\overrightarrow{p}_n$.
- Each particle $n$ moves within the search space iteratively updating its velocity and position vectors, attracted by its $\overrightarrow{p}_n$ and its neighborhood's $\overrightarrow{p}_b$:

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

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

# Particle Interaction

- where $w$ is the inertia weight parameter and $\phi_1$ and $\phi_2$ are the particle's acceleration coefficients that control the magnitude of stochastic attraction toward $\overrightarrow{p}_n$ and $\overrightarrow{p}_b$.
- Each vector $\overrightarrow{U}(0, \phi_i)$ contains random numbers drawn from a uniform distribution in $[0, \phi_i]$.

# Outline

# Ant Colony Optimization

- It is a class of metaheuristics that provide a generic framework of communication between simple agents, whose task is to construct candidate solutions to the optimization problem under consideration.
- Two sources of information:
  - **Stigmergic information**: represented by pheromone matrix $T = [\tau_{ij}]_{N \times N}$, where each $\tau_{ij}$ is associated with the corresponding directed edge $e_{ij}$ in the network architecture.
  - **Heuristic information**: each solution component is associated with a heuristic value $\eta_{ij}$ representing the desirability of adding edge $e_{ij}$ to the solution under construction.

# Stochastic Generation of Candidate Solution

- Graph-theoretic approaches generate topologies that exhibit the scale-free property.
- A parametric, generative process is 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.

# Extended DSF (eDSF)

- eDSF 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 has a degree $k(u) = k_{in}(u) + k_{out}(u)$. If $k(u) > 0$, node $u$ is considered to be existing (connected).

# Extended DSF (eDSF)

- Let $G(t_0)$ be the initial graph, where $t_0$ is the number of edges and $n_0$, the number of nodes.
- At each discrete time t, $N_{new}(t) \cap N_{old}(t) = 0$, $N_{new}(t) \cup N_{old}(t) = N(t)$ and $\mid N_{old}(t) \mid = n(t)$.
- Let $\alpha, \beta, \gamma, \delta_{in}$ and $\delta_{out}$ be nonnegative, real numbers, with $\alpha + \beta + \gamma = 1$ and $\alpha + \gamma > 0$.
- Let also $\tau_{ij} \in \Re^+$ be the pheromone value associated with edge $e_{ij}$.
- In order to obtain $G(t+1)$ from $G(t)$, exactly one of the three following operations is performed:

# Possible Operations I

- With probability $\alpha$, an edge is added from a new node $u$ to an existing node $w$. Node $u$ is selected 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 $j, \kappa \in \mathcal{N}_{\text{new}}(t)$.

- Node $w$ is selected according to $k_{in} + \delta_{in}$ and the pheromone value corresponding to its incoming edge from node $u_j$, with probability:

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

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

# Possible Operations II

- With probability $\beta$, an edge is added from an existing node $u$ to an existing node $w$.

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

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

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

# Possible Operations III

- With probability $\gamma$, a new edge is added from an existing node $u$ to a new node $w$.

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

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

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

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

# Extended DSF (eDSF)

- If for all edges $e_{ij}$, $\tau_{ij} = c$, with $c \in \Re^+$, 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.

# ACO algorithm

**Algorithm 4** ACO algorithm

Initialize pheromone matrix $T = [\tau_{ij}]_{N \times N}$
Initialize global adjacency matrix $\widetilde{M} = [0]_{N \times N}$
Initialize global error vector $\widetilde{\mathcal{E}} = [\infty]_{1 \times N}$
**for** each ACO step **do**
  Initialize local adjacency matrix $\widehat{M} = [0]_{N \times N}$
  Initialize local error vector $\widehat{\mathcal{E}} = [\infty]_{1 \times N}$
  **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 $\widehat{M}$, $\widehat{\mathcal{E}}$ with $M_k$, $\mathcal{E}_k$ (see Algorithm 1)
  **end for**
  Update pheromone matrix $T$ with $\widehat{M}$, $\widehat{\mathcal{E}}$ (see Algorithm 2)
  Update $\widetilde{M}$, $\widetilde{\mathcal{E}}$ with $\widehat{M}$, $\widehat{\mathcal{E}}$ (see Algorithm 1)
  Update pheromone matrix $T$ with $\widetilde{M}$, $\widetilde{\mathcal{E}}$ (see Algorithm 2)
  Perform pheromone evaporation (see Algorithm 3)
**end for**
**return** solution $(\widetilde{M}, \widetilde{\mathcal{E}})$

# ACO algorithm

**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 ε'_i < ε_i then
    for each regulator j do
      m_ij ← m'_ij
    end for
  end if
end for
```

**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
      τ_ij ← τ_ij + log ε_i/(log ε_i − 1)
    end if
  end for
end for
```

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

```
for each target i do
  for each regulator j do
    τ_ij ← (1 − ρ)τ_ij
  end for
end for
```

# Outline

# Outline

# Two Experiments with Artificial Data

- Experiment 1: Reconstructing a small ANN.
- Experiment 2: Reconstructing a Real-World Network.

# Exp1 - PHERO model

- The alternative **pheromone-based** (PHERO) model ignores heuristic and operates solely on the basis of stigmergic information represented by the pheromone matrix $T$.
- It 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 $\tau_{ij}$.

# Exp1 - Reconstructing a small Artificial Network

- Small artificial gene network comprising 4 nodes and 8 edges.
- Network dynamics were generated using an RNN model.

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].*

- Training dynamics comprise 50 time points and were assembled by sampling from a time series of 300 time points generated by setting $\Delta t = 0.1$.

# Exp1 - Reconstruction Experiment

- $L = 10$ independent ACO trials of 50 steps each using a population of 5 artificial ants.

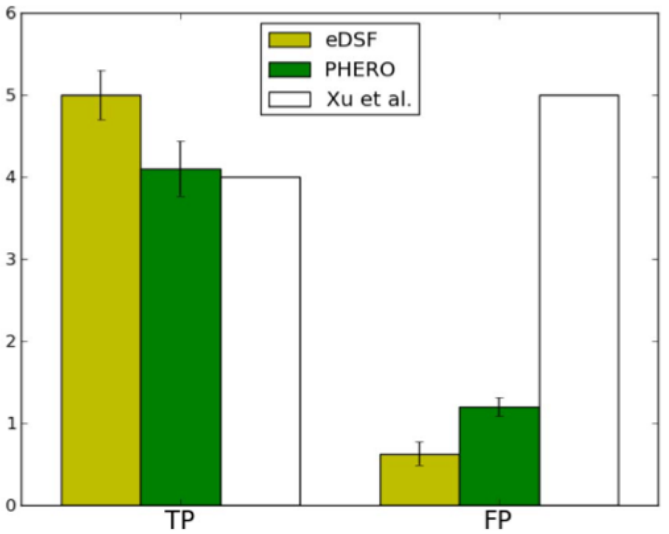



Fig. 3. The statistical properties (true and false positives) of the solutions acquired from $L = 10$ independent ACO trials, for each experimental setting (eDSF and PHERO) of Section 4.1.1. The error bars correspond to the standard error of the sample. The white bars represent the results of Xu et al.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

# Exp1 - Nature of inferred relationship

- 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 TP edges whose corresponding RNN weight sign (+ or -) matched the nature of the relationship (activatory or repressive) in the actual network (Table1).
- 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).

# Exp2 - Reconstructing a Real-World Network

- Artificial data with added noise in order to compare the performance of the eDSF and PHERO models in the context of ACO/PSO.
- GeneNet-Weaver (GNW), an open source software package for the generation of network topologies.
- GNW generates network topologies extracting modules from known biological networks and simulates the system dynamics using differential equation models.

# Exp2 - Target Network

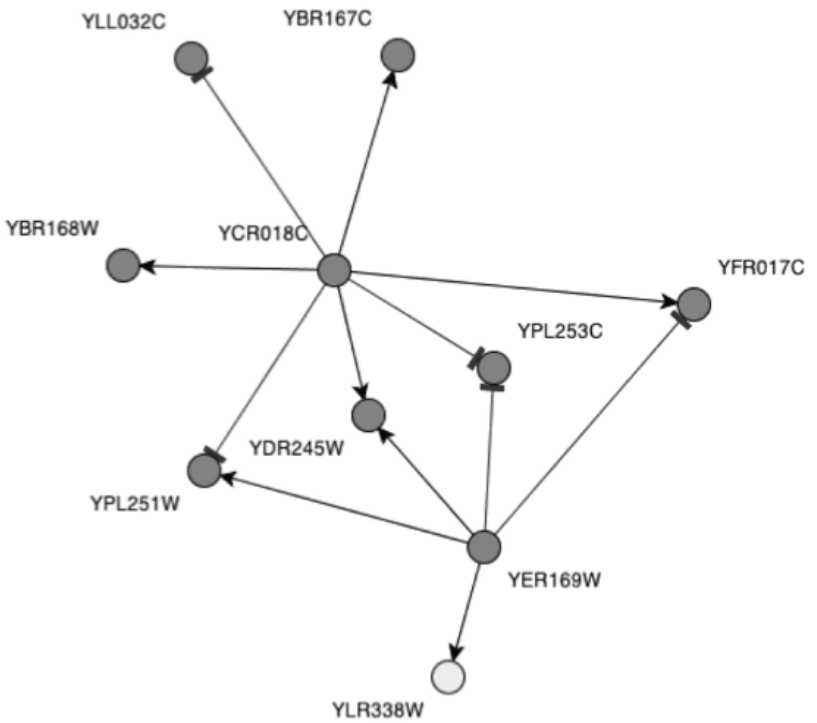



Fig. 4. The topology of the artificial network that was extracted from the yeast genetic interaction network using GNW. Normal arrow heads denote activation, while T-shaped arrow heads denote repression.

# Exp2 - Description

- The generated dynamics consisted of 21 time points.
- Number of independent ACO trials is $L = 10$.
- Each ACO run consists of 100 steps with a population size of five artificial ants and a pheromone evaporation rate set to $\rho = 0.1$.
- Number of iterations for each PSO subproblem is set 2000.
- Swarm population size is calculated by $n_i = [10 + 2\sqrt{D_i}]$, where $D_i = 2 + \sum_j m_{ij}$.

# Exp2 - Results

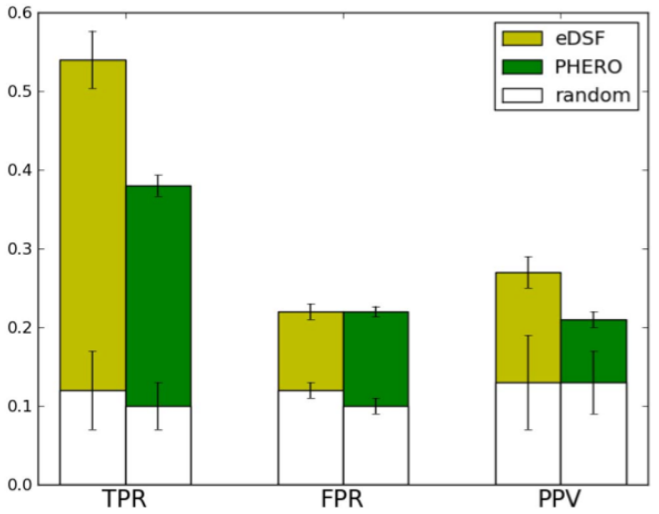
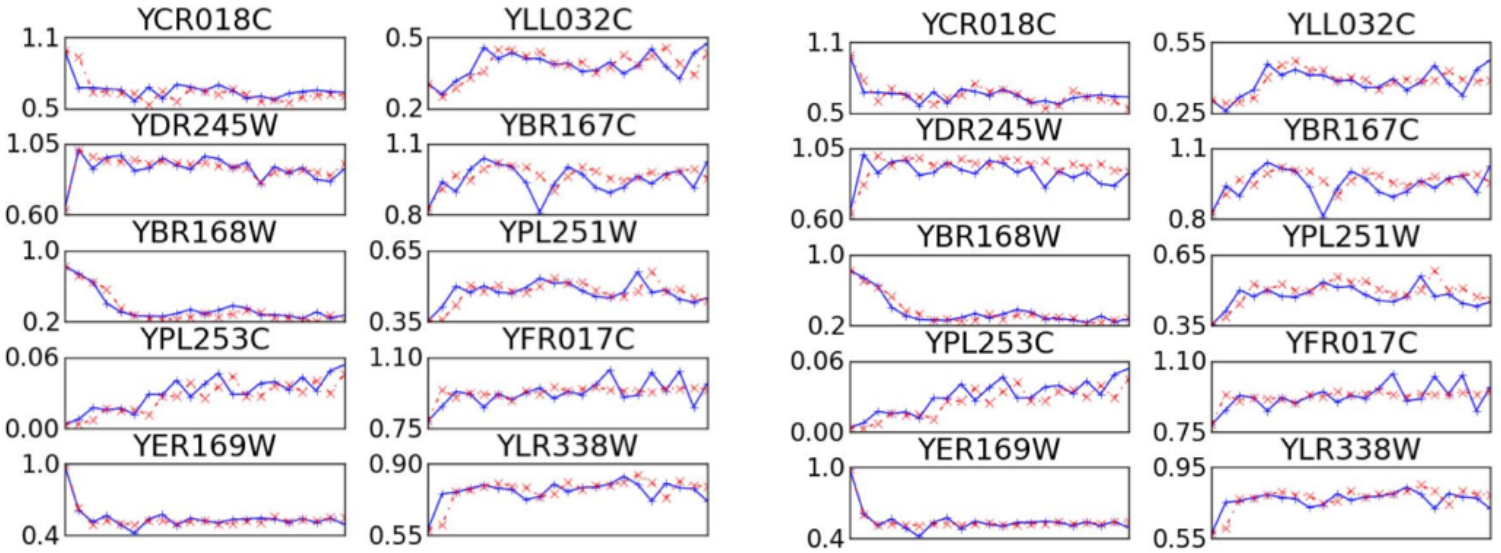



Fig. 6. The statistical properties of the solutions acquired from $L = 10$ independent ACO trials, for each experimental setting (eDSF and PHERO) of Section 4.1.2. The error bars correspond to the standard error of the sample. The *random* values were determined by running the corresponding model (eDSF 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 $\tau_{ij} = 1$.

# Time series




(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 $\check{M}$ 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 $2.4 \times 10^{-3}$. (b) The PHERO experiment with prediction MSE $2.5 \times 10^{-3}$.

# Exp2 - Nature of inferred relationship

- For each experiment (eDSF and PHERO), we trained 100 RNN instances that corresponded to the inferred topology $\tilde{M}$ (with $\sigma = 0.5$) using PSO.
- 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).

# Outline

# Description

- ACO/PSO framework, incorporating the eDSF model of generating candidate architectures, is applied to SOS response system of *E. coli.*
- It is a transcriptional network consisting of proteins that are involved in DNA repair activities.
- DNA repair is regulated by the interplay between two proteins: LexA and RecA.
- ACO/PSO framework settings used were the same as in the artificial data experiments.

# Inferred network topologies

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 $\check{M}$ with a strict threshold value $\sigma = 0.9$ for the four SOS data sets.*

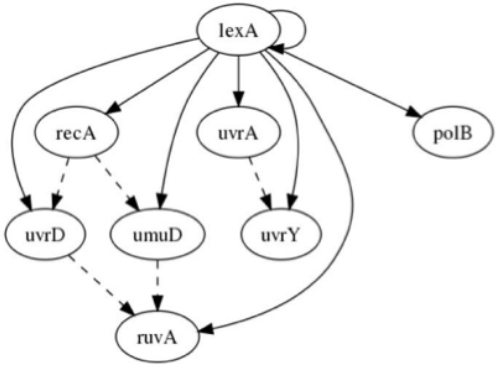



Fig. 9. The predicted topology $\check{M}$ of the SOS network, resulting from 10 independent ACO runs, with a strict threshold set at $\sigma = 0.9$. Correctly inferred edges (true positives) have been drawn with solid lines and falsely inferred edges (false positives) with dashed lines.

- Best prediction was achieved using the second time series, with an inferred topology consisting of 13 edges, 8 of which were TP and 5 FP.

# Results Comparison

TABLE 4
Comparison of Predictions for the SOS Data Set

| Known interaction | Predictions by | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | [12] | [73] | [69]† | [25]‡ | [70]† | [71]† | [72]† | ACO/PSO |
| 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).

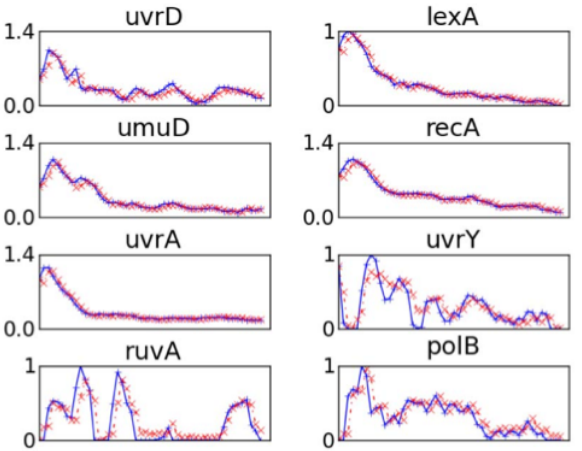



Fig. 10. Actual and predicted dynamics for the SOS experiment, using the second data set. The prediction MSE is $1.2 \times 10^{-2}$. The actual dynamics consist of the second time series in the data set of Ronen et al. [68]. The predicted dynamics were generated using a trained RNN model that corresponded to the inferred network topology $\check{M}$, 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.