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

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

Difficulties

- The analysis of gene expression data to identify the underlying relationships has important difficulties as:
  - 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 **directed 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.

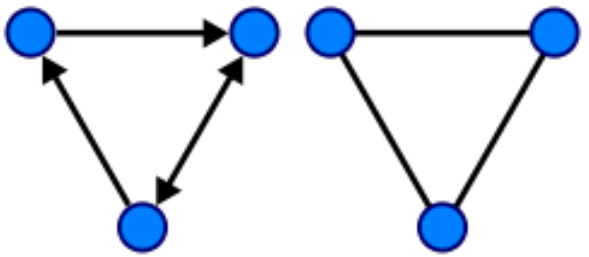

Figure: Left - Directed Graph. Right - Undirected Graph.

Recurrent Neural Network

- Dynamics of the system are expresed by RNN formalism.
- The expression level x_i of the ith 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 ith gene).
- c_i is a time constant (scaling factor).
- w_{ij} is the weight associated to ith and jth genes.
- f is a sigmoidal function (logistic function).

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 ith subproblem, the parameters under training include **only** 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 ε_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 ε_i represents the minimum achieved prediction error for the temporal expression pattern of gene i.
- PSO is applied separately to each independent subproblem i for estimating the corresponding model parameters.

Particle Swarm Optimization

- Particle position vectors encode the RNN parameters associated with the current (ith) subproblem under consideration.
- 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 $\overrightarrow{p}_n$.

Updating Particle Interaction

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

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

- where w is the inertia weight parameter that controls the scope of the search (balance between exploration and exploitation).
- ϕ_1 and ϕ_2 are the particle's acceleration coefficients that control the magnitude of stochastic attraction toward $\overrightarrow{p}_n$ and $\overrightarrow{p}_b$ ($\phi_1 = \phi_2 = 1.496$).
- Each vector $\overrightarrow{U}(0,\phi_i)$ contains random numbers drawn from a uniform distribution in $[0,\phi_i]$.

Outline

Ant Colony Optimization

- ACO is a **metaheuristic** optimization algorithm.
- Two sources of information:
  - **Stigmergic information**: represented by pheromone matrix $T = [\tau_{ij}]_{N \times N}$, where each τ_{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 η_{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 (eDSF) model that augments the heuristic degree-based preferential principle of the original model, with a stigmergic pheromone-based preferential principle.

Extended DSF (eDSF)

- eDSF describes a stochastic process where a graph (network) grows by adding a single directed edge (regulatory relationship) at each discrete time step.
- At each such step, three possible operations are possible:
  - an edge is added from a new node u to an existing node w.
  - an edge is added from an existing node u to an existing node w.
  - a new edge is added from an existing node u to a new node w.

Extended DSF (eDSF)

- A node u is considered to be existing (connected) if it has a degree $k(u) = k_{in}(u) + k_{out}(u) > 0$. Otherwise, it is a new (unconnected) node.
- ACO is introduced by T matrix, where $\tau_{ij} \in \Re^+$ is the pheromone value associated with edge e_{ij}.
- If for all edges, $\tau_{ij} = c$, with $c \in \Re^+$, the selection of any node is equiprobable with respect to the stigmergic information. So, eDSF model is equivalent to the original DSF model.

Example of Operations I

- Node u is selected according to the pheromone values corresponding to its outgoing edges, with probability:

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

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

- 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:

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

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

- Where δ_{in} is nonnegative, real number.

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 ρ

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

Outline

Three Experiments

- Experiment 1: Reconstructing a small ANN with artificial data.
- Experiment 2: Reconstructing a real-world Network with artificial data.
- Experiment 3: Reconstructing a real-world Network with real-world data.

Description

- ACO framework, incorporating the eDSF model of generating candidate architectures.
  - $L = 10$ ACO trials, population size of 5 artificial ants, pheromone evaporation rate set to $\rho = 0.1$
- Create a RNN from the candidate architectures inferred by ACO.
- Train 100 RNN instances that corresponded to the inferred topology $\tilde{M}$ using PSO.
  - Different time points (50, 21, 21).

Exp 3 - Inference using Real-World Data

- 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.

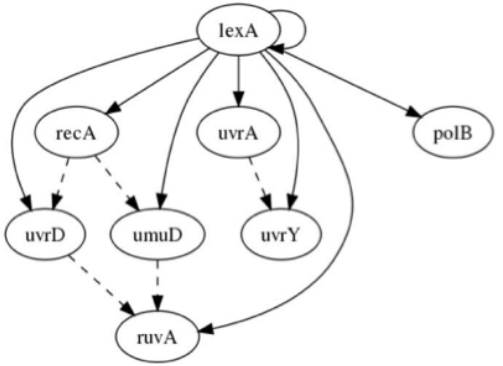

Fig. 9. The predicted topology $\check{M}$ of the SOS network, resulting from 10 independent ACO runs, with a strict threshold set at $\sigma = 0.9$. Correctly inferred edges (true positives) have been drawn with solid lines and falsely inferred edges (false positives) with dashed lines.

- 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.

Actual and Predicted Dynamics using PSO

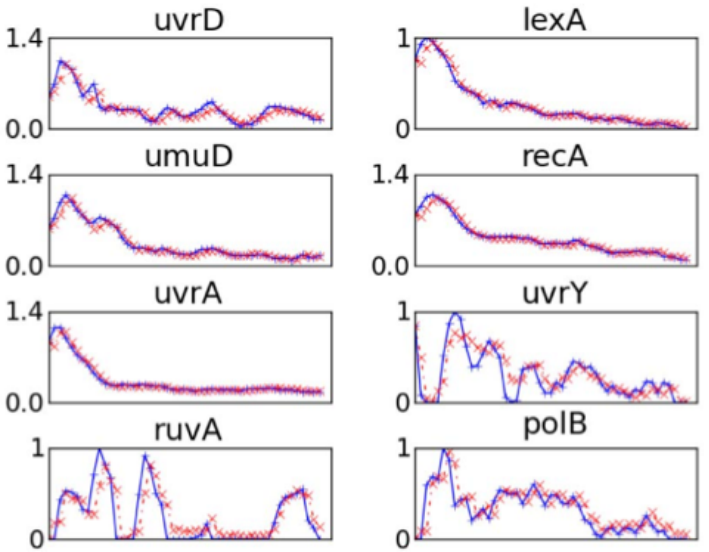

Fig. 10. Actual and predicted dynamics for the SOS experiment, using the second data set. The prediction MSE is 1.2×10^{-2}. The actual dynamics consist of the second time series in the data set of Ronen et al. [68]. The predicted dynamics were generated using a trained RNN model that corresponded to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

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).