

A Graph-theoretic Method for the Discretization of Gene Expression Measurements

Elena Dimitrova, John McGee, and Reinhard Laubenbacher
Virginia Bioinformatics Institute (0477), Blacksburg, VA 24061

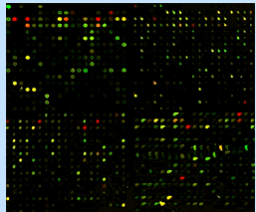
INTRODUCTION

Applications requiring **discrete** data:

- reverse engineering of gene regulatory networks,
 - machine learning algorithms,
 - Bayesian network applications, etc.
- Experimental data are often continuous and need to be discretized.

We study **gene regulatory networks (GRN)** – collections of genes that interact with each other, governing the rates at which genes in the network are expressed.

Gene expression measured on *microarrays* by the abundance of mRNA.



A microarray. Courtesy of www.mcbl.arizona.edu

We use **finite dynamical systems** to model GRN of n genes:

$$(f_1, \dots, f_n) : \mathbb{F}^n \rightarrow \mathbb{F}^n$$

- genes represented by variables
- f_i gives dynamics of gene i
- f_i a polynomial function over finite field \mathbb{F}

Experimental gene expression data are continuous and needs to be discretized over a finite field.

New discretization method employs **graph theory, clustering, and information theory** to discretize experimental data into a finite number of states and to maintain high information content. Algorithm determines number of discrete states.

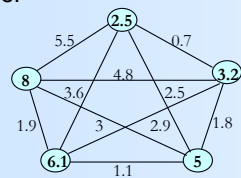
METHOD

1. Discretization of one variable

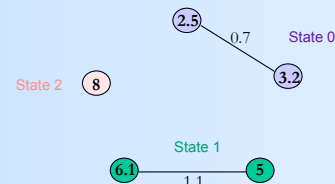
- Let T = set of time series of m distinct values for a given variable.
 - Construct complete weighted graph $G(V, E)$ on m vertices:
vertex label = value of entry in T
edge weight = Euclidean distance between entries
 - Consecutively delete heaviest edge(s) until graph is disconnected into components.
- Apply a(ii) to a component $C(V_c, E_c)$ if one of the following holds:
 - $2 * \text{avg_edge_wt}(C) \geq \text{avg_edge_wt}(G)$
 - $2 * | \max(V_c) - \min(V_c) | \geq | \max(V) - \min(V) |$
 - If b(i) and b(ii) fail and disconnecting C increases information content in discretized values.
- # components = # discrete states

Example

$T = \{2.5, 3.2, 5, 6.1, 8\}$ time series for a variable.



Delete heaviest edges until graph is disconnected and apply b(i, ii, and iii):



2. Discretization of several variables into the same number of states

To discretize N variables, apply 1 (left) to each variable and discretize them into m_1, \dots, m_N states. Let $m = \max\{m_i \mid 1 \leq i \leq N\}$. For every variable, sort its values in each cluster and split the one which contains the two most distant values between these values. Repeat until m clusters are obtained.

For discretization over finite fields, keep splitting clusters until smallest possible $p^n \geq m$ is reached.

VERIFICATION

Artificial gene network AGN [1]
time series

discretization

Discrete time series

reverse engineering [2]

Discrete model DM

Network and model dynamics match:

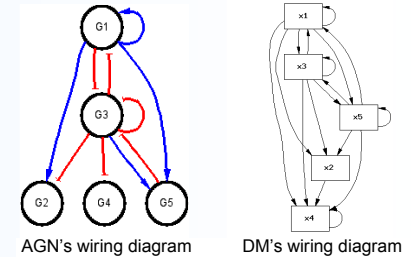
Matching fixed points:

Single steady state for AGN:
(1.99, 1.99, 0.000025, 0.998, 1.99)

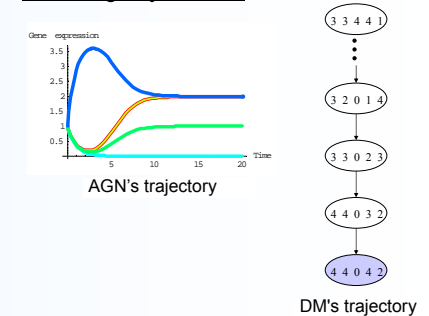
discretization

Single fixed point for DM:
(4, 4, 0, 4, 2)

Matching wiring diagrams:



Matching trajectories:



CONCLUSIONS

1. Method's effectiveness demonstrated in reverse engineering of GRN.
2. Method provides a tool for discretization of gene expression measurements into a finite number of states.

REFERENCES

- [1] Mendes, P., Sha, W., and Ye, K., 2003. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* **19**, 122–129.
- [2] Laubenbacher, R. and Stigler, B., 2004. A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology* **229**, 523–537.