An Optimization Algorithm for the Identification of Biochemical Network Models

www.math.vt.edu

M. Paola Vera-Licona, Luis David Garcia, John McGee, Miguel Colon-Velez and Reinhard Laubenbacher
mveralic@.math.vt.edu

www.vbi.vt.edu

INTRODUCTION

An important problem in computational biology is the modeling of several types of networks, ranging from gene regulatory networks and metabolic networks to neural response networks.

In [LS], Laubenbacher and Stigler presented an algorithm that takes as input time series of system measurements, including certain perturbation time series, and provides as output a discrete dynamical system over a finite field.

Since functions over finite fields can always be represented by polynomial functions, one can use tools from computational algebra for this purpose. The key step in the algorithm is an interpolation step, which leads to a model that fits the given data set exactly. Due to the fact that biological data sets tend to contain noise, the algorithm leads to over-fitting.

Here we present a genetic algorithm, that optimizes the model produced by the Laubenbacher-Stigler algorithm between model complexity and data fit. This algorithm (implemented in C++), uses tools from computational algebra in order to provide a computationally simple description of the mutation rules.

DISCRETE POLYNOMIAL MODELS

A **discrete polynomial model** is a vector function

$$f = (f_1,\ldots,f_n) : \text{☞}^n \rightarrow \text{☞}^n$$

where ☞ is a finite field and $f_i : \text{☞}^n \rightarrow \text{☞}$, are local update polynomials for each node $i = 1,\ldots,n$.

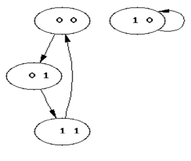
The phase space of a polynomial model, is given by the directed graph, where:

vertices := states of system
Edge from v_j to v_k iff $f(v_j)=v_k$

Therefore a time series is represented as a path in this graph.

Example of a model and phase space graph

$f = (f_1, f_2) : \{1,0,\}^2 \rightarrow \{1,0\}^2$
$f_1 = x_1 + x_2$
$f_2 = x_1 + 1$



GENETIC ALGORITHM

1. Elements of the Genetic Algorithm

Genome: A finite dynamical system model as a set of *d* polynomials over ☞$_2$ (finite field of 2 elements)

Fitness function: Hamming distance between time series generated by the model and the input time series along with a measure of model complexity

Mutated Initial States: Improve robustness of estimated model

Mutation: Addition or removal of one variable from one monomial

Crossover: Creation of a Polynomial as mixture of subsets of the monomials from two parent models.

2. Input Data

Time Series: A set of discrete time series over ☞$_2$, with different initial states

Knockout Data: Perturbed data from the knockout of some or all of the entities in the network

Candidate Model: Obtained by applying L-S algorithm to the available data.

3. Effectiveness of Fitness Score

We used empirical measurements to verify that if an estimated model is close to the true model, then the time series produced by the estimated model will have a small Hamming distance from that produced by the true model from the same initial state.

Figure 1 plots *h* for the values of *d* up to 30, where *d* is the number of mutations between two models, and h is the average Hamming distance between time series produced by the two models.

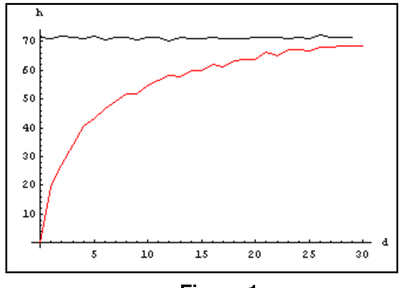


Figure 1

GENETIC ALGORITHM

3. Parameters of Genetic Algorithm

In order to control the evolutionary search for a satisfactory solution model, our genetic algorithm uses 12 parameters (e.g. population size, mutation rate, crossover rate, etc).

The tuning of this set of parameters is one main feature in the early stage of this project.

4. Computational Algebra Tools

We formulated an upper bound for the number of variables in every monomial, given the input time series samples. This formulation is based on the maximum support of a monomial member of any Gröbner basis of the ideal that vanishes on a time series *t*, which greatly restricts the genetic algorithm search space, improving the efficiency of the optimization.

Testing and Tuning

We employed an 8-variable model to generate time series and perturbed data, as input into the Laubenbacher-Stigler method and/or directly as input into the genetic algorithm.

The performance of the G.A. is significantly improved when the model from L-S was used as input in the G.A.

$f_1 = x_1 + x_2$
$f_2 = x_2 + x_3x_5$
$f_3 = x_3 + x_4x_6$
$f_4 = x_4 + x_8$
$f_5 = x_5 + x_3$
$f_6 = x_6$
$f_7 = x_7$
$f_8 = x_8 + x_1$

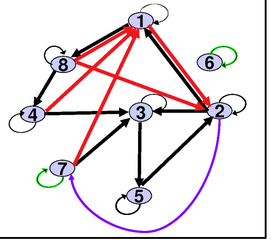
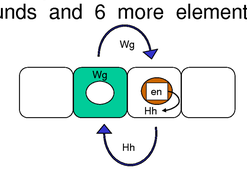


Figure 2. On the left, model used. On the right, Topology of the model used. In black is the intersection of the graphs produced by the original model, L-S model and that obtained from the genetic algorithm. In red are the extra links that the last two share and finally, the purple link is the extra link that L-S model has and that the genetic algorithm was able to delete.

APPLICATIONS

Network of the Segment Polarity Genes in the Fruit Fly *D. Melanogaster*

This network consists of the genes whose function is to define the segment boundaries and polarity in the *D. melanogaster* embryo. Each one of the four cells in the network consists of 5 genes and the products encoded by them, as well as two protein compounds and 6 more elements from neighbor cells.



We perform a series of experiments on the 21-variable model obtained in [LS] , adjusting our method by applying the genetic algorithm twice.

Label of Parameter Set	Corrected links	Loops deleted
1.2.	3	1
2.2.	6	2
3.3.	3	1
5.2.	2	0

Table 1. Considering [AO] model (21 nodes and 44 links), we compare our results with [LS]. The second column refers to the number of excessive links in [LS] topology that the G.A. corrected and the second column represents the number of loops that the G.A. deleted.

FINAL COMMENTS

1. Initial results indicate that our Genetic Algorithm approach to either discovery or refinement of a dynamical system model can be an effective element of a robust system for genetic network identification.
2. An extension of this algorithm for more general finite fields is being performed as part of a second phase of this project.

REFERENCES

[LS] R. Laubenbacher and B. Stigler, A Computational Algebra Approach to the Reverse Engineering of Gene Regulatory Networks

[BO] E. Babson, S. Onn. The Hilbert Zonotope and a Polynomial Time Algorithm for Universal Grobner Bases. 2003

[HC] Hochmuth Gregor, On the Genetic Evolution of a Perfect Tic-Tac-Toe Strategy

[AO] R. Albert and H. Othmer. The Topology of the Regulatory Interactions Predicts the Expression Pattern of the Segment Polarity Genes in *Drosophila melanogaster.*