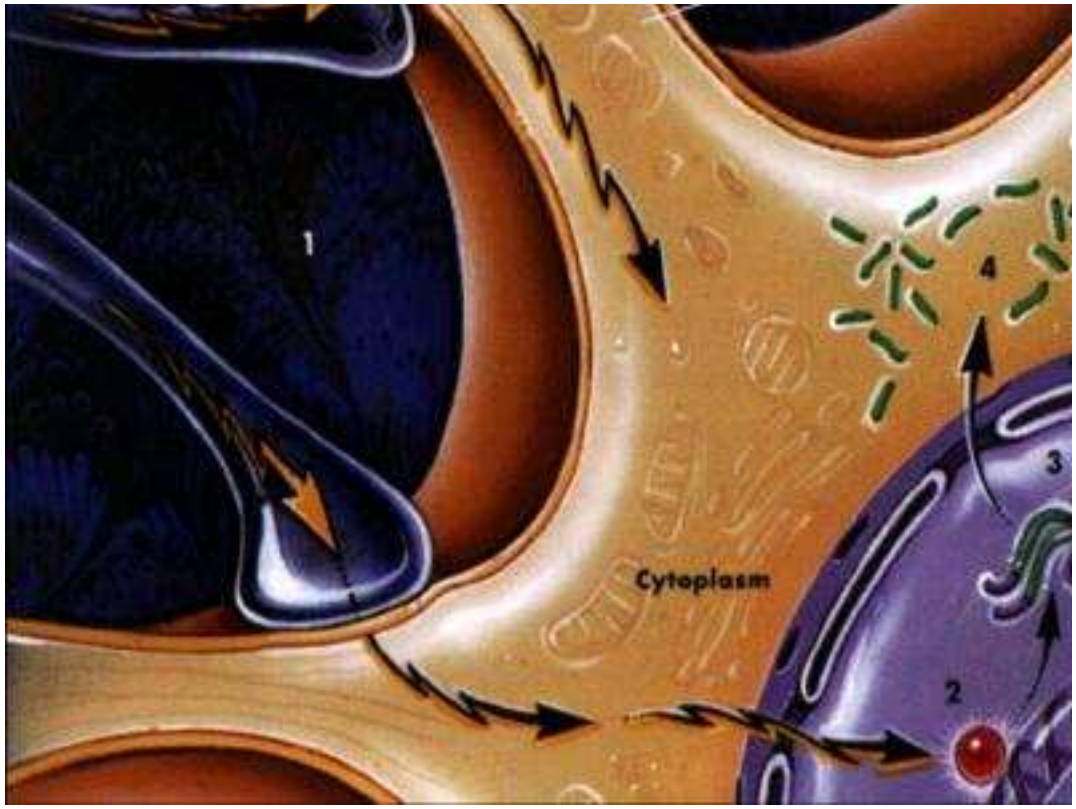
Analysis of Gene Regulation Networks Using Finite-Field Models

Humberto Ortiz Zuazaga

November 29, 2005

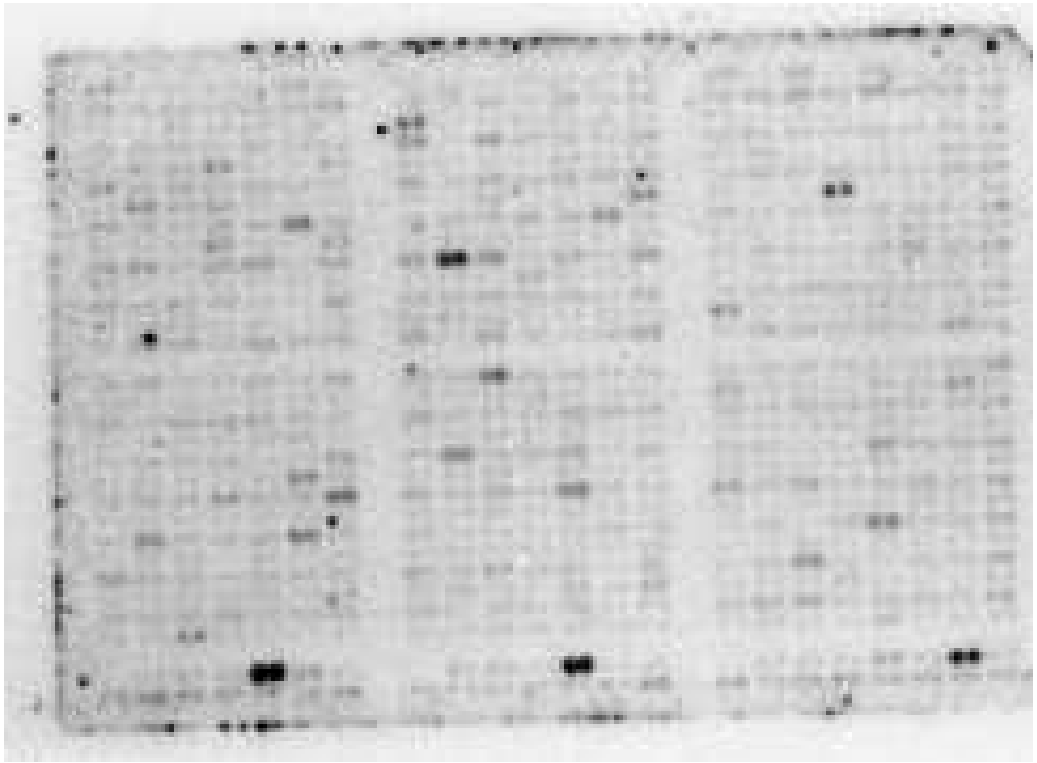
Background

A Model Cell



Post Genome Biology

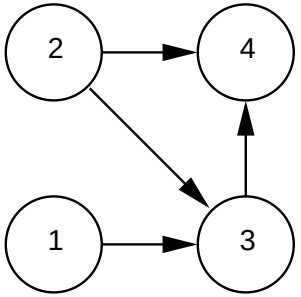
or, "I've got all the genes, now what do I do with them?"



Reverse Engineering Genetic Networks

- Input:
  - A set of genes
  - A set of gene expression measurements
- Output:
  - A set of *control functions* by which some genes control others

Boolean Genetic Networks



$$
\begin{aligned}
f_1 &= 1 \\
f_2 &= 1 \\
f_3 &= x_1 \wedge x_2 \\
f_4 &= x_2 \wedge \neg x_3
\end{aligned}
$$

Boolean Genetic Network Model

We define Boolean genetic network model (BGNM):

- A *Boolean variable* takes the values 0, 1.
- A *Boolean function* is a function of Boolean variables, using the operations $\wedge$, $\vee$, $\neg$.

A *Boolean genetic network model* (BGNM) is:

- An n-tuple of Boolean variables $(x_1, \ldots, x_n)$ associated with the genes
- An n-tuple of Boolean control functions $(f_1, \ldots, f_n)$, describing how the genes are regulated

Reverse Engineering Boolean Networks

- Akutsu, S. Kuahara, T. Maruyama, O. Miyano, S. 1998. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (SODA 98), H. Karloff, ed. ACM Press.

- Ideker, T.E., Thorsson, V., and Karp, R.M. 2000. Discovery of regulatory interactions through perturbation: inference and experimental design. Pacific Symposium on Biocomputing 5:302-313.

- S. Liang, S. Fuhrman and R. Somogyi. 1998. REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. Pacific Symposium on Biocomputing 3:18-29.

Boolean results

- Problem: Consistent assignment
- Input: a gene network and an assignment of True or False to each variable
- Output: True if the assignment is consistent with the rules of the network, False otherwise
- Result: Akutsu et al prove this problem is NP-complete (by reduction from 3-SAT)

Perturbation experiments

- Problem: how many experiments do I need to do?
- Input: a gene network with n genes
- Output: the number of gene knockdown (force gene to 0) or overexpression (force gene to 1) experiments needed to completely determine the genetic network
- Result: worst case, $2^{(n-1)/2}$
- Result: if the degree (number of genes that act on a gene) is limited to D, $O(n^{2D})$

Further work proceeds on the assumption that $D = 2$ or $D = 3$.

Boolean Bugs

- Boolean variables can only represent all-or-none effects
- Boolean models are deterministic
- Efficient algorithms for Boolean networks require indegree of genes to be limited to a small constant value (*i.e.*, at most 2 or 3 transcription factors act on any given gene)

Finite fields represent an alternative algebraic structure to substitute Booleans. Our research seeks to characterize genetic networks based on these fields.

Finite field models

- Each gene can be an element of a finite field
- Multivariate polynomial models
- Based on computing Gröebner bases and ideals

Laubenbacher, R. and Stigler, B. (2004), 'A computational algebra approach to the reverse engineering of gene regulatory networks', *J. Theor. Biol.* **229**, 523–537.

Finite Fields

A *finite field* $\{F, +, \cdot\}$ is a finite set F, and two operations $+$ and $\cdot$ that satisfy the following properties:

- $\forall a, b \in F,\ a + b \in F,\ a \cdot b \in F$
- $\forall a, b \in F,\ a + b = b + a,\ a \cdot b = b \cdot a$
- $\forall a, b, c \in F,\ a + (b + c) = (a + b) + c,\ (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- $\forall a, b, c \in F,\ a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- $\exists 0, 1 \in F,\ a + 0 = 0 + a = a,\ a \cdot 1 = 1 \cdot a = a$
- $\forall a \in F,\ \exists (-a) \in F$ s.t. $a + (-a) = (-a) + a = 0$
  $\forall a \neq 0 \in F, \exists a^{-1} \in F$ s.t. $a \cdot a^{-1} = a^{-1} \cdot a = 1$

The World's Smallest Finite Field

The integers 0 and 1, with integer addition and multiplication modulo 2 form the finite field $Z_2 = \{\{0, 1\}, +, \cdot\}$.

The operators $+$ and $\cdot$ are defined as follows:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Products of Sums and Sums of Products

We can realize any Boolean function as an expression over Z_2:

$$\begin{aligned} X \wedge Y &= X \cdot Y \\ X \vee Y &= X + Y + X \cdot Y \\ \neg X &= 1 + X \end{aligned}$$

This perspective unites the mathematical foundation of finite fields with the logic of Boolean networks, but remaining within the realm of communications science.

Probabilistic Boolean Networks

- Each gene may have many controlling functions, select among them by random process.
- Generate predictors by enumerating all k-input functions for each gene, tractability requires restricting k to a small integer (4)
- Selection probabilities proportional to *coefficient of determination* of the given gene by a predictor

Shmulevich, I., Dougherty, E. R., Kim, S. and Zhang, W. (2002), 'Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks', *Bioinformatics* **18**(2), 261–274.

Probabilistic Sequential Systems

- Generalize BPN to GF(p)
- Combine sequential dynamical systems and PBN

Aviñó, M. A., Bulancea, G. and Moreno, O. (2005), Probabilistic sequential systems, *in* 'Proceedings GENSISP'.

Conditioned taste aversion (CTA)

- associative aversive conditioning paradigm
- Animals are exposed to a novel taste, the *conditioned stimulus*
- An *unconditioned stimulus* induces malaise
- The animals develop a long lasting aversion to the conditioned stimulus

CTA Dataset

- two controls, the pre-treatment group and the one hour saline group
- four time points, 1, 3, 6, and 24 hours after conditioning
- 1185 genes on each spotted array
- 5 biological replicates of each array

Chiesa, R., Ortiz-Zuazaga, H. G., Ge, H. and Peña de Ortiz, S. (2000), Gene expression profiling in emotional learning with cDNA microarrays, *in* '40th meeting of the American Society for Cell Biology', San Francisco, California.

Objectives and Preliminary Results

Objectives

1. To develop new algorithms and heuristics for clustering and error correction, building on finite field models of gene expression networks, and majority logic decoding.

2. To develop new algorithms and heuristics for reverse engineering probabilistic models, extending univariate polynomial finite field models

Objective 1

To develop new algorithms and heuristics for clustering and error correction, building on finite field models of gene expression networks, and majority logic decoding

Finite Field Genetic Networks

Any BGNM can be converted into an equivalent model over Z_2 by realizing the boolean functions as sums-of-products and products-of-sums. We now have a *finite field genetic network* (FFGN):

- An n-tuple of variables over Z_2, $(x_1, \ldots, x_n)$ associated with the genes
- An n-tuple of functions over Z_2, $(f_1, \ldots, f_n)$, describing how the genes are regulated

Revrese engineering can be done using Lagrange interpolation of univariate polynomials from the time series data.

Moreno, O., Ortiz-Zuazaga, H., Corrada Bravo, C. J., Aviñó-Diaz, M. A. and Bollman, D. (2004), 'A finite field deterministic genetic network model', Preprint.

FFGN Models

- Finite field models are an improvement on Boolean network models
- Laubenbacher's multivariate polynomial representation of networks utilizes Gröebner bases, a somewhat esoteric area
- Bollman and Orozco have demonstrated that multivariate and univarite polynomial models are equivalent
- Our approach is to bring the tools of modern communications science to bear on the problem of analyzing regularoty networks

Bollman, D. and Orozco, E. (2005), Finite field models for genetic networks. Preprint.

Error correction

A01a glypican 1; HSPG M12; nervous system cell-surface heparan sulfate proteoglycan

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	0.172	0.099	0.176	0.142	0.062	0.152
2	0.274	0.168	0.126	0.114	0.104	0.276
3	0.003	0.119	0.552	0.178	0.193	0.114
4	0.114	0.139	0.6	0.311	0.179	0.181
5	0.04	0.006	0.172	0.103	0.036	-0.047
average	0.121	0.106	0.325	0.17	0.115	0.135
control	0.113					
epsilon						0.022
calls			+	+	0	0

Majority logic

Repetition	1 h	3 h	6 h	24h
1	+	0	−	0
2	−	−	−	+
3	+	+	+	+
4	+	+	+	+
5	+	+	0	−
consensus	+	+	?	+

Substituting averaged controls

Repetition	1 h	3 h	6 h	24h
1	+	+	−	+
2	0	0	0	+
3	+	+	+	0
4	+	+	+	+
5	+	0	−	−
cvac	+	+	?	+

Pruning extreme values

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	—	0.099	0.176	0.142	—	0.152
2	—	—	0.126	0.114	0.104	—
3	0.003	0.119	—	—	0.193	0.114
4	0.114	0.139	—	—	0.179	0.181
5	0.04	—	0.172	0.103	—	—
new average	0.052	0.119	0.158	0.12	0.159	0.149
new control	0.086					
new epsilon	0.063					
new calls			+	0	+	0

Consistent calls

1. at least two of the above set of calls agrees in the last 4 columns of data (1 h, 3 h, 6 h, and 24h)

2. either the 1 h or the 24 h columns is a "0"

3. across the last 4 columns of data, the column exhibits the consecutive zeros property (*i.e.*, values do not oscillate between "0" and "+" or "−")

A01a is not consistent

	1 h	3 h	6 h	24h
average calls	+	+	0	0
consensus	+	+	?	+
cvac	+	+	?	+
new calls	+	0	+	0

Clustering

- Categorizing each timepoint for each gene into coarse divisions yields a clustering of genes
- In our current experiment there are $3^4 = 81$ possible clusters that a gene may fall into
- Longer time series or larger fields will allow finer grained division of the genes into clusters

Results

- 127 consistent genes in CTA dataset
- Grouping genes with same calls in 1 h – 24 h timepoints yields 23 clusters
- Obtained upstream sequences for "000+" cluster (1020 bp, 800 bp before start of transcription) expression most similar to CREB
- Searched for transcription factor binding sites with TESS
- Found two very interesting genes: Pmch and Calca, both have CRE sites
- These genes were excluded from analysis using traditional microarray techniques, and thus would have been missed

Pmch

- Cyclic neuropeptide
- Affects appetite or metabolism
- Induces hippocampal synaptic transmission

Varas, M., Perez, M., Ramirez, O. and de Barioglio, S. (2002), 'Melanin concentrating hormone increase hippocampal synaptic transmission in the rat', *Peptides* **23**(1), 151–155.

Calca

- Vasodilator
- May be involved in axonal regeneration
- May be involved in synaptogenesis

Li, X. Q., Verge, V. M., Johnston, J. M. and Zochodne, D. W. (2004), 'CGRP peptide and regenerating sensory axons', *J. Neuropathol. Exp. Neurol.* **63**(10), 1092–1103.

Objective 2

To develop new algorithms and heuristics for reverse engineering probabilistic genetic network models, extending univariate polynomial finite field models

Probabilistic finite field network

- PFFN $A = A(V, F, C)$
- n nodes $V = \{x_1, x_2, \ldots, x_n\}$, representing the genes
- $x_i \in \mathsf{GF}(p^m)$
- a list for each gene $F = \{F_1, F_2, \ldots, F_n\}$ of sets
- the sets $F_i = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ contain functions
- each function $f_j^{(i)} : \mathsf{GF}(p^m)^n \to \mathsf{GF}(p^m)$ is called a predictor
- a list $C = \{c_j^{(i)}\}_{i \in I,\ j \in J}$, of selection probabilities.
- The selection probability that a given predictor $f_j^{(i)}$ is used to update the value of a gene x_i is $c_j^{(i)}$

PFFN Example

- PFFN $A = (V, F, C)$
- $V = \{X_0, X_1, X_2, X_3\}$, $X_i \in \mathsf{GF}(2^2)$
- $F = \{F_0, F_1, F_2, F_3\}$
  - $F_0 = \{f_0^{(0)} = 0, f_1^{(0)} = 1\}$
  - $F_1 = \{f_0^{(1)} = 0, f_1^{(1)} = 1\}$
  - $F_2 = \{f_0^{(2)} = X_0 \cdot X_1, f_1^{(2)} = X_0 + X_1\}$
  - $F_3 = \{f_0^{(3)} = X_1 \cdot (X_2 + 1), f_1^{(3)} = X_0 + X_1\}$
- $C = \{c_j^{(i)}\}_{i \in \{0,1,2,3\}, j \in \{0,1\}}$
- $c_j^{(i)} = 0.5$ for all $i \in \{0, 1, 2, 3\}, j \in \{0, 1\}$

Node (and predictor) splitting

- $X_0 = \alpha \cdot {}_0x_1 + 1 \cdot {}_0x_0$
- $X_1 = \alpha \cdot {}_1x_1 + 1 \cdot {}_1x_0$

$$
\begin{aligned}
f_0^{(2)} &= X_0 \cdot X_1 \\
&= (\alpha \cdot {}_0x_1 + 1 \cdot {}_0x_0) \cdot (\alpha \cdot {}_1x_1 + 1 \cdot {}_1x_0) \\
&= \alpha^2 \cdot {}_0x_1 \cdot {}_1x_1 + \alpha \cdot {}_0x_1 \cdot {}_1x_0 + \alpha \cdot {}_1x_1 \cdot {}_0x_0 + 1 \cdot {}_0x_0 \cdot {}_1x_0 \\
&= (\alpha + 1) \cdot {}_0x_1 \cdot {}_1x_1 + \alpha \cdot {}_0x_1 \cdot {}_1x_0 + \alpha \cdot {}_1x_1 \cdot {}_0x_0 + 1 \cdot {}_0x_0 \cdot {}_1x_0 \\
&= \alpha \cdot {}_0x_1 \cdot {}_1x_1 + 1 \cdot {}_0x_1 \cdot {}_1x_1 + \alpha \cdot {}_0x_1 \cdot {}_1x_0 + \alpha \cdot {}_1x_1 \cdot {}_0x_0 + 1 \cdot {}_0x_0 \cdot {}_1x_0 \\
&= \alpha \cdot ({}_0x_1 \cdot {}_1x_1 + {}_0x_1 \cdot {}_1x_0 + {}_1x_1 \cdot {}_0x_0) + 1 \cdot ({}_0x_1 \cdot {}_1x_1 + {}_0x_0 \cdot {}_1x_0)
\end{aligned}
$$

Future Directions

Objective 1

- Dr. Peña's lab is validating expression changes for Calca and Pmch
- We are working with Dr. Giray to apply our techniques to protein time series data from honeybee

Objective 2

- Design univariate polynomial interpolation routines to learn PFFN from data, given a data set with n genes, r repetitions of t time points or conditions
- Current Boolean and PBN techniques require enumerating $\binom{n}{k}$ input functions, with k representing the genes that may act on another gene, "reasonable" restrictions on k are unreasonable
- Interpolating r^t candidate functions from the data is cheaper if $r, t << n$ as is currently the case
- Each candidate function can be selected with a probability proportional to a correlation coefficient of the function to the time course data, analogous to PBN

Expected outcomes

- As predicted by our analysis, Pmch and Calca will be modulated by CTA training, and will be dependent on CREB. We expect our error correction and clustering techniques to result in a joint publication with Dr. Peña's lab in 2006.
- We expect our error correction and clustering techniques to yield insight into protein interaction networks
- We expect that PFFN will more accurately describe biological systems than PBN
- We expect that univariate polynomial interpolation will prove more efficient than partial enumeration techniques for the construction of PFFN from microarray data

Ethical issues

- Genetic testing: microarrays are used for diagnosis, can be used to test for errors in transcriptional regulation
- Genetic engineering: knowlege of the transcriptional control can be used to select for certain outcomes (bigger cows, prettier children, ...)
- Reverse engineering: algorithms for reverse engineering gene regulatory networks can also be applied to reverse engineer hardware or software
- Cracking electronic communications: our techniques could in principle be used to reverse engineer encryption systems and eavesdrop on confidential information.