Analysis of Gene Regulation Networks Using Finite-Field Models

Humberto Ortiz-Zuazaga

October 28, 2005

Contents

1 Rationale

Microarrays allow researchers to simultaneously measure the expression of thousands of genes. They give invaluable insight into the transcriptional state of biological systems, and can be important in understanding physiological as well as diseased conditions. However, the analysis of data from many thousands of genes, from only a few replications is very difficult.

The major goal of this proposal is to further develop information theoretic techniques for microarray analysis, and specifically, to develop procedures to cluster gene expression values and determine gene regulatory interactions.

We will use published microarray data sets, synthetic data, and a data set from learning and memory processes in rats to test our procedures. In our preliminary data, we have devised a novel method of correcting errors in microarray experiments, that also clusters genes into groups, and categorizes their measurements into coarse divisions, suitable for discrete techniques for reverse engineering. These techniques are based on finite fields and algebraic coding theory.

2 Background

cDNA microarrays are a technique for measuring the abundance of RNA from many thousands of genes simultaneously in an inexpensive experiment (Schena, Shalon, Davis & Brown 1995). They are used extensively for diagnostic purposes, and the data they allow researchers to collect have permitted the study of genome wide interactions among genes. The analysis of microarray data, however, is a difficult task, proving a fruitful area of research in numerous fields. An extensive review is available in (de Jong 2002). This section will attempt to review the literature most relevant to the proposed work.

2.1 Stages of microarray analysis

The analysis of microarray data is a complex, multi-stage process that typically involves the following steps:

1. microarray image analysis
2. normalization
3. detection of differential expression
4. clustering
5. biological network analysis

This proposal concentrates on the last two stages.

2.2 Clustering

Clustering of gene expression measurements is an important step in many analysis, most early microarray work performed hierarchal clustering, where genes are successively agglomerated into groups by selecting the two clusters whose average expression values are closest (Eisen, Spellman, Botstein & Brown 1998). It is typical to first cluster genes before trying to determine the gene

regulatory network by reverse engineering. Clustering helps reduce the computational resources required to analyze microarray data sets by grouping together many separate genes that demonstrate similar patterns of expression (Akutsu, Miyano & Kuhara 1999). It also can help in determining common functionality or common regulatory elements of genes which cluster together (D'haeseleer, Liang & Somogyi 2000).

2.3 Genetic network models

As early as 1969 Stuart A. Kauffman (Kauffman 1969) (see (Kauffman 1993) for a detailed review) proposed the far-reaching and important idea of using Boolean logic, the logic of computers, to produce and gain insight into the logic of genes. The invention of cDNA microarrays brought a resurgence of interest in these Boolean genetic network models.

2.4 Boolean models and the reverse engineering problem

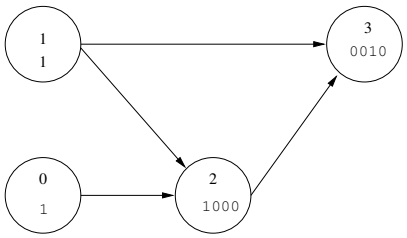
A series of papers in 1998, 1999 and 2000 defined Boolean network models, reverse engineering, and proved interesting results on the number of experiments required to completely define a Boolean network.

Taking the model definition from (Ideker, Thorsson & Karp 2000), for example, we can describe a genetic network as:

1. A graph consisting of N numbered nodes and, $1 \leq n \leq N$.
2. A set of directed edges between nodes.
3. A Boolean function f_n for each node.

An edge from a node to another represents an influence of the first node on the expression of the second.

Figure 1 reproduced from (Ideker et al. 2000) represents a small example genetic network with 4 nodes and 4 edges.



x_1	1	0	1	0
x_2	1	1	0	0
x_3	0	0	1	0

$$
\begin{aligned}
x_0 &:= 1 \\
x_1 &:= 1 \\
x_2 &:= x_0 \wedge x_1 \\
x_3 &:= x_1 \wedge \neg x_2
\end{aligned}
$$

Figure 1: Example of the Boolean steady-state network model: (a) a directed graph structure with numbered nodes connected by edges, (b) the truth table (shown for node 3 only) and (c) the logic equations for each node.

We understand that the following is a formalization of the model presented in (Ideker et al. 2000).

Definition 1 *A Boolean variable assumes the values 0,1.*

Definition 2 *A Boolean function is a function involving Boolean variables and the operations* $\wedge$, $\vee$, $\neg$ *with the following definitions:*

X	Y	$X \wedge Y$	$X \vee Y$	$\neg X$
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

Definition 3 *A dBnm is a set of n Boolean variables* $(x_1, \ldots, x_n)$ *which are inputs, and a set of n Boolean functions which are the outputs* $(f_1, \ldots, f_n)$*. The Boolean variables represent genes or stimuli, and the Boolean function* f_i *represents how the gene i is determined by all the other genes.*

Lemma 1 *Given n Boolean variables* $(x_1, \ldots, x_n)$ *and define* $f(x_1, \ldots, x_n)$ *for all possible values. Then there is a Boolean function that coincides with f as defined.*

Proof: See any book on computer architecture (c. f. Patterson & Hennessy (1997)) for realizing a Boolean function as sums of products and products of sums.

We will additionally define:

An *expression matrix* is a set of measurements (such as those which result from microarray experiments) over the genetic network. From this expression data, the challenge is to reconstruct or reverse engineer the genetic network.

A *gene perturbation experiment* is an expression matrix where some entries correspond to measurements taken when the value of one gene or more are forced to a known state.

Akutsu, Kuahara, Maruyama & Miyano (1998) proved lower and upper bounds on the number of gene perturbation experiments required to completely determine a gene network. The results are discouraging, since in the general case, the problem is shown to be NP-complete. However, in (Liang, Fuhrman & Somogyi 1998), an efficient algorithm for determining the gene network from a set of input-output pairs is developed, assuming that each gene has an indegree in the directed graph that is at most three. This restriction corresponds to saying that at most three genes have an influence on the expression of the target gene. Further research proceeds on the assumption that this indegree is bounded by a small constant. In (Akutsu et al. 1999) it is shown that a gene network will be recovered with high probability in only $O(\log n)$ experiments if the indegree is at most two. Ideker et al. (2000) provide an iterative procedure for selecting genes to perturb while determining a genetic network such that the uncertainty in the specification of the model is reduced. After this series of papers, work on these Boolean models was mostly discontinued, biologists objected to the simplicity of the Boolean representation of genes.

It is also important to note that all of these Boolean network papers leave unspecified the manner in which gene expression measurements are converted to Boolean values. For example, Ideker et al. (2000) simply says that gene values will be approximated as high or low and represented by the values 1 or 0.

2.5 Probabilistic models

Probabilistic Boolean networks, PBN, were developed in (Shmulevich, Dougherty, Kim & Zhang 2002, Shmulevich, Dougherty & Zhang 2002) to overcome problems encountered in the study of gene expression data with Boolean networks. The principle problem PBNs address is the inherent determinism of Boolean network models. PBN incorporate a stochastic process, to allow for uncertainty in the data, and in the produced models.

PBN can incorporate many Boolean functions for a single gene, selecting among the multiple functions according to a probability that corresponds to how well the function correlates to the data.

PBN models of genetic networks can capture uncertainty in the specification of the genetic networks, but are very computationally intensive (Suh, Dougherty, Kim, Bittner, Chen, Russ & Martino 2002). The principal computational obstacle in PBN is the enumeration of all possible functions controlling a gene. The usual solution is to partially enumerate all functions with a small number of inputs.

2.6 Partial enumeration

In both the Boolean network models and PBN, reverse engineering via partial enumeration of functions as described in (Shmulevich, Dougherty, Kim & Zhang 2002, Liang et al. 1998, Akutsu et al. 1998) requires limiting the number of inputs to each genetic function, usually assuming that between 2 to 4 genes affect the expression of a given gene. This requirement for computational tractability directly conflicts with the evidence that transcriptional networks for higher organisms are significantly more complex (Lemon & Tjian 2000, Merika & Thanos 2001), with even yeast having up to 10 or more transcription factors influencing the expression of a single gene (Lee, Rinaldi, Robert, Odom, Bar-Joseph, Gerber, Hannett, Harbison, Thompson, Simon, Zeitlinger, Jennings, Murray, Gordon, Ren, Wyrick, Tagne, Volkert, Fraenkel, Gifford, & Young 2002).

2.7 Finite field genetic network models

Moreno has shown that any function over Boolean variables can be realized instead as a function over the finite field Z_2. In a natural manner, we can extend this to sets of Boolean functions, such as those contained in a Boolean network model. The advantage of finite fields and vector spaces is that they allow tools developed for cryptanalysis and communications theory to be applied to microarray data. One such tool, the Finite Dynamical System (Moreno, Bollman & Aviñó 2002, Laubenbacher & Pareigis 2001) (FDS) is of particular interest.

In Moreno, Ortiz-Zuazaga, Corrada Bravo, Aviñó-Diaz & Bollman (2004) we demonstrate how to extend Boolean genetic networks into finite field deterministic genetic network models, first, define a finite field over Z_2:

Definition 4 *$+$ and $\cdot$ in Z_2 are defined as follows:*

X	Y	$X \cdot Y$	$X + Y$
1	1	1	0
0	1	0	1
1	0	0	1
0	0	0	0

Remark 1 *Z_2 is a field with those two operations.*

Definition 5 *A polynomial function in the variables $X_1, \ldots, X_n$ over Z_2 is a multi-variable polynomial in the variables $X_1, \ldots, X_n$.*

Lemma 2 *Given n variables over Z_2 and define $f(x_1, \ldots, x_n)$ for all possible values then there is a function over Z_2 that coincides with f as defined.*

Proof: note that for any two Boolean variables X, Y we have:

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

Now if we are given the function f we first invoke Lemma 1 and realize it as a Boolean function. Now using the above it is easy to see how we can realize it as a polynomial function over Z_2.

Example 1 *To illustrate let us work the example of Figure 1, that was given in (Ideker et al. 2000). We have there $f_0 = 1, f_1 = 1, f_2 = x_0 \cdot x_1, f_3 = x_1 \cdot (x_2 + 1)$. Note now they are multi-variable polynomials over the finite field Z_2.*

Lemma 3 *The set of Boolean functions coincides with the set of functions over Z_2.*

Proof: This follows from Lemma 1 and Lemma 2.

Definition 6 *A finite dynamical system (FDS) is a pair (V, f) where V is the set of vectors over a finite field $GF(p^n)$ and $f : V \to V$.*

Definition 7 *A FDS (V, f) is* linear *if the function f is a linear function.*

Theorem 1 *The dBnm defined in (Ideker et al. 2000) is a Finite Dynamical System.*

Proof: Note first, that the dBnm can be seen as a set of n functions over Z_2, by Lemma 3. The FDS over Z_2 can also be seen as a set of n functions (in the n variables $x_1, \ldots, x_n$) from Z_2^n to Z_2. Then, it is easy to see that they are in fact equivalent.

Example 2 *Let us illustrate our theorem in the case of the example of Figure 1 (from (Ideker et al. 2000)) and we will see that it is a finite dynamical system. Let $V = Z_2^4$ in our definition of FDS and let $f = (f_0, f_1, f_2, f_3)$ where f_i, $i = 0, 1, 2, 3$ are as it was given in the previous example. Then it should be clear that $f : Z_2^4 \to Z_2^4$ and that the example of Figure 1 is a FDS. The proof of the theorem follows the same method.*

Boolean networks and PBN then share 2 limitations: they can only represent genes as "on" or "off", and they limit the nature of the gene interaction network to ensure computational tractability.

Both these problems have been addressed by the formulation of polynomial models over finite fields (Laubenbacher & Stigler 2001, Laubenbacher & Stigler 2003). These models allow for a richer variation of gene expression levels, and remove the restrictions on the degree of the genes. These

polynomial models, however, are more akin to Boolean network models than to PBN, as they are deterministic, and cannot represent uncertainty in the data or network models.

Several alternative representations and techniques for polynomial models over finite fields have been developed (Aviñó, Green & Moreno 2004, Green 2004, Moreno et al. 2002), and Bollman & Orozco (2005) demonstrates that these polynomial models are equivalent to those described in (Laubenbacher & Stigler 2003, Laubenbacher & Stigler 2004).

This research lead to a series of techniques for error-correction, clustering, and reverse engineering based on finite fields. The current proposal seeks to extend these models, and produce new biological insight from microarray data.

2.8 Microarray experiments

Microarray experiments were performed in the laboratory of Dr. Sandra Peña de Ortiz. Her lab has kindly provided us with data sets for collaborative analysis. The methods described in this proposal were developed for the purpose of analyzing these data sets, but are sufficiently general to analyze any equivalent data set.

The studies described here focus on one cognitive task, conditioned taste aversion (CTA), as a model system for gene expression profiling. CTA, is an associative aversive conditioning paradigm in which pairing gastrointestinal malaise (induced by lithium chloride, LiCl, the unconditioned stimulus) with prior exposure to a novel taste (the conditioned stimulus) may create a strong and long lasting aversion to the novel taste.

CTA lends itself as an excellent model system to study the dynamics of gene regulation in learning and memory because it is a single trial associative learning paradigm, which involves discrete regions in the brain, including selected amygdala nuclei (Yamamoto, Shimura, Sako, Yasoshima & Sakai 1994, Yasoshima, Shimura & Yamamoto 1995).

2.8.1 Behavioral training

Behavioral training of rats in the CTA task prior to collection of the microarray data used for our experiments was done as described in (Ge, Chiesa & Peña de Ortiz 2003).

2.8.2 Microarray measurements

The gene profiling experiment was replicated five times. Four animals were used per condition for each replicate. Thus, a total of sixteen rats were used per condition. Animals were sacrificed by decapitation at 1, 3, 6, and 24 hours after conditioning. Hybridization, image capture and analysis was similar to the procedures described in (Robles, Vivas, Ortiz-Zuazaga, Felix & Peña de Ortiz 2003). The data set thus obtained (CTA data set) is described in (Chiesa, Ortiz-Zuazaga, Ge & Peña de Ortiz 2000). In summary, the data has two controls, the pre-treatment group and the one hour saline group, and four time points, 1, 3, 6, and 24 hours after conditioning. Each array has 1185 genes, and we have 5 replicates of the arrays.

3 Objectives

As stated in Section 1, our principal goal is to develop new techniques for analyzing microarray data utilizing tools from information theory. These tools have been shown to be applicable to the

analysis of microarray expression data. To accomplish our goal we propose the following objectives:

1. Previous Boolean network models assume the values for each gene have been discretized, usually by thresholding, and no errors are present in the discretization. We have seen that using multiple repetitions of an experiment, we can discretize a gene into several values, and use majority logic decoding and other techniques to correct for errors in the microarray image analysis and discretization procedures. Thus we propose **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. We have seen that the computational tractability of Boolean and probabilistic Boolean approaches to the reverse engineering problem depend on the assumption that each gene is influenced by a small number of other genes. This assumption is flawed, except perhaps in the simplest of organisms. Multivariate finite field models of gene networks overcome this restriction. We have seen that univariate finite field models are equivalent to multivariate models, and may be simpler to manipulate. This thesis seeks to **develop new algorithms and heuristics for reverse engineering, extending univariate polynomial finite field models to probabilistic models.**

4 Methods and Preliminary Results

To accomplish our general goal of developing new techniques for the analysis of microarray expression data, we propose the following methods and experiments.

4.1 Error Correction and Clustering

We have devised a scheme for detecting and correcting errors using discretized data.

Here we apply our technique to data from one gene in the CTA data set described in Section 2, to illustrate the method:

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

Each row is a repetition of the microarray experiment. Columns represent the measurements of the genes. Pre and Sal are the pretreatment (time 0) and injection with saline solution controls.

4.1.1 Averaging

The first step in the analysis is to average the expression across repetitions.

average 0.121 0.106 0.325 0.17 0.115 0.135

We also average our control columns to obtain a control value of 0.113.

We compute an epsilon value, such that either the 1 h or 24 h columns are within the range of control +/- epsilon. In this case, the epsilon is 0.022.

4.1.2 Discretization

We proceed to discretize each repetition by comparing each column to the control +/- epsilon. We illustrate for repetition 1:

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	0.172	0.099	0.176	0.142	0.062	0.152

The control for this repetition is $(0.172 + 0.099)/2 = 0.1355$, epsilon is fixed for all our tests at 0.022. We now call a column "+" if its value is greater than the control + epsilon, "-" if is is less than control - epsilon, and "0" otherwise.

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	+	-	+	0	-	0

Repeating for the remaining repetitions yields;

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	+	-	+	0	-	0
2	+	-	-	-	-	+
3	-	+	+	+	+	+
4	0	0	+	+	+	+
5	0	0	+	+	0	-

4.1.3 Majority logic decoding

We now obtain a consensus for each column by majority logic decoding, 3 or more occurrences of the same symbol in a column indicate that symbol is the consensus. If no consensus is obtained, we indicate "?".

	Pre	Sal	1 h	3 h	6 h	24h
consensus	?	?	+	+	?	+

4.1.4 Discretizing versus averaged controls

The above procedure is very sensitive to the value of the controls. Errors in the controls can skew the entire set of calls. We devised an alternate method of discretization that replaces the control value for each row by the average of the control value for all the rows. In our case this average control is 0.113. The discretization of the repetitions using this average control yields the following values, which we summarize with this consensus versus average control (cvac):

Repetition	Pre	Sal	1 h	3 h	6 h	24h
1	+	0	+	+	-	+
2	+	+	0	0	0	+
3	-	0	+	+	+	0
4	0	+	+	+	+	+
5	-	-	+	0	-	-
cvac	?	?	+	+	?	+

4.1.5 Discretizing the average

We also compute the discretization of the average values of each column, using the control 0.113 and the epsilon 0.022:

	Pre	Sal	1 h	3 h	6 h	24h
average	0.121	0.106	0.325	0.17	0.115	0.135
calls	0	0	+	+	0	0

4.1.6 Error correction

We now enter an error correction phase, we seek out outliers in the data of the columns and remove them, and recompute the average, controls, and epsilon.

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	—	—

With these outliers deleted from our data we now have new averages, control and epsilon values:

	Pre	Sal	1 h	3 h	6 h	24h
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	+	0	+	0

4.1.7 Consistent calls

We are now ready to produce a consistent set of calls for the gene. A set of calls is consistent if the following conditions are met:

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 "-")

As an example, the set of calls for A01a are:

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

These calls are not consistent, and this gene is removed from further examination. Together, the procedures we developed and the consistency criteria try to capture biologist's intuitions on the nature of gene expression changes.

4.1.8 Results of analyzing the CTA data set

We have performed the analysis described above on the CTA data set described in Section 2.8.2. In this data set, there are 127 consistent genes, which we divide into clusters by grouping together the genes that have the same set of calls in the 1 - 24 hour timepoints. This results in

23 clusters. We focus on the cluster labeled "000+". The consensus of the calls for these genes represents no change over the 1, 3, and 6 hour time points, followed by upregulation at the 24 hour timepoint. This cluster consists of genes whose expression most closely matches the expression profile of CREB. CREB is a transcription factor which we know to be required for long-term memory (Lamprecht, Hazvi & Dudai 1997). We investigated the genes in this cluster in depth, retrieving the gene information and sequence from the Ensembl Genome Browser version 32 (Hubbard, Andrews, Caccamo, Cameron, Chen, Clamp, Clarke, Coates, Cox, Cunningham, Curwen, Cutts, Down, Durbin, Fernandez-Suarez, Gilbert, Hammond, Herrero, Hotz, Howe, Iyer, Jekosch, Kahari, Kasprzyk, Keefe, Keenan, Kokocinsci, London, Longden, McVicker, Melsopp, Meidl, Potter, Proctor, Rae, Rios, Schuster, Searle, Severin, Slater, Smedley, Smith, Spooner, Stabenau, Stalker, Storey, Trevanion, Ureta-Vidal, Vogel, White, Woodwark & Birney 2005).

From Ensembl we obtained genomic sequence for each gene, 1020 base pairs starting 800 base pairs upstream of the transcription start site. These sequences were then submitted to TESS (Schug & Overton 1997) to search for transcription factor binding sites.

Two genes in particular caught our interest: Pmch and Calca. Both genes have CRE elements in their upstream regions, meaning they are possible targets of CREB1 regulatory function. According to the Rat Genome Database (Rat Genome Database Web Site 2005), Pmch is a cyclic neuropeptide that induces hippocampal synaptic transmission. It seems to have an effect on appetite or metabolism (Pereira-da Silva, Torsoni, Nourani, Augusto, Souza, Gasparetti, Carvalheira, Ventrucci, Marcondes, Cruz-Neto, Saad, Boschero, Carneiro & Velloso 2003) and anxiety (Kela, Salmi, Rimondini-Giorgini, Heilig & Wahlestedt 2003), and promotes synaptic transmission in the hippocampus (Varas, Perez, Ramirez & de Barioglio 2002). Calca is principally a vasodilator, but seems to have a role in axonal regeneration or synaptogenesis (Li, Verge, Johnston & Zochodne 2004). Thus these genes exhibit a pattern of expression consistent with the expression of Creb1, have CRE elements upstream of their transcription start site, and seem to have a role in strengthening or creating new synapses. Thus they are strongly implicated as important genes for the formation of memories. Our collaborator, Dr. Sandra Peña de Ortiz, and her students are actively seeking confirmation of these genes' role in CTA. In collaboration with Dr. Moreno, we will confirm the changes in expression of these genes and investigate their role in memory.

4.2 Probabilistic finite field network models

A Probabilistic Finite Field Network (PFFN) is an extension of Probabilistic Boolean Networks (PBN) (Shmulevich, Dougherty, Kim & Zhang 2002, Shmulevich, Dougherty & Zhang 2002) to work over values in finite fields, similar to how finite dynamical systems, as defined in (Laubenbacher & Pareigis 2001) generalize Boolean dynamical systems.

A PFFN $A = A(V, F)$ is defined by a set of n nodes $V = \{x_1, x_2, \ldots, x_n\}$ with values over some arbitrary finite field $\mathrm{GF}(p^m)$, and a list $F = \{F_1, F_2, \ldots, F_n\}$ of sets $F_i = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ of functions over $\mathrm{GF}(p^m)$. Each function $f_j^{(i)} : \mathrm{GF}(p^m)^n \to \mathrm{GF}(p^m)$ is called a predictor. Each node $x_i \in \mathrm{GF}(p^m)$ represents the state of gene x_i. The set F_i contains the possible regulatory interactions for gene x_i. All genes are updated synchronously. At each time step, one of the predictors for gene x_i is selected randomly from the set F_i according to a predetermined probability distribution.

A realization of the PFFN at a given time is determined by a vector of field valued functions $f_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)})$ for $i \leq k_i \leq l(i)$ and $f_{k_i}^{(i)} \in F_i$, $f_k : \mathrm{GF}(p^m)^n \to \mathrm{GF}(p^m)^n$ acts as a transition function, specifying the new value of each gene x_i, and thus defining the dynamics of the

system. The selection probability that a given predictor $f_j^{(i)}$ is used to update the value of a gene x_i is equal to $c_j^{(i)} = P\{f^{(i)} = f_j^{(i)}\} = \Sigma_{k_i=j} P\{f = f_k\}$.

Two particular examples of finite fields are of particular interest, when the genes $x_i \in \mathrm{GF}(3)$, known as the ternary model. In the ternary model, we can capture the biological intuition of genes being expressed, repressed, or unchanged. Another interesting variant is when $x_i \in \mathrm{GF}(2^m)$, as then each gene can be modeled as a vector of m binary variables. Since in (Moreno et al. 2002, Moreno et al. 2004) we demonstrate a mapping from models over GF(2) to Boolean networks, we can immediately apply this technique to the study of Probabilistic Boolean Networks. For example, a gene in a model with $\mathrm{GF}(p^m) = \mathrm{GF}(2^2)$ can be modeled as a pair of nodes $(x_i, x_{i'})$, with $x_i, x_{i'} \in \{0, 1\}$.

4.2.1 PFFN Example

We show an example constructing a PFFN over $\mathrm{GF}(2^2)$ and showing how to convert this PFFN to a PBN.

Our PFFN $A = (V, F)$, with $V = \{X_0, X_1, X_2, X_3\}$ the values of $X_i \in \mathrm{GF}(2^2)$, and $F = \{F_0, F_1, F_2, F_3\}$, for our example, we have two functions in each set, and choose between them with equal probability. Thus $F_0 = \{f_0^{(0)} = 0, f_1^{(0)} = 1\}$, $F_1 = \{f_0^{(1)} = 0, f_1^{(1)} = 1\}$, $F_2 = \{f_0^{(2)} = X_0 \cdot X_1, f_1^{(2)} = X_0 + X_1\}$, $F_3 = \{f_0^{(3)} = X_1 \cdot (X_2 + 1), f_1^{(3)} = X_0 + X_1\}$. Each predictor $f_j^{(i)}$ has an associated probability of being selected, $c_j^{(i)}$, in this simple example, $c_0^{(i)} = c_1^{(i)} = 0.5$ for $i = 0, 1, 2, 3$.

Note that $\mathrm{GF}(2^2)$ can be represented as $\mathrm{GF}(2^2) = \{0, 1, \alpha, \alpha^2\}$ where α is a root of the polynomial $z^2 + z + 1$ (with coefficients in Z_2). By taking $1, \alpha$ as a basis, we can say that $0 = (0,0), 1 = (0,1), \alpha = (1,0), \alpha^2 = (1,1)$. Note also that $\alpha^2 = \alpha + 1$ and $\alpha^3 = 1$. We will use these equivalences to simplify expressions in the conversion.

We will split each node of V into a pair of nodes with values in GF(2), so X_i splits into ${}_ix_1$, and ${}_ix_0$, or equivalently $X_i = \alpha \cdot {}_ix_1 + 1 \cdot {}_ix_0$. Each of the predictors will also be split into two functions over GF(2). Take for example $F_2 = \{f_0^{(2)}, f_1^{(2)}\}$ recall that $f_0^{(2)} = X_0 \cdot X_1$, we will substitute $X_0 = \alpha \cdot {}_0x_1 + 1 \cdot {}_0x_0$ and $X_1 = \alpha \cdot {}_1x_1 + 1 \cdot {}_1x_0$. Thus

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

If we let ${}_0f_0^{(2)} = {}_0x_1 \cdot {}_1x_1 + {}_0x_1 \cdot {}_1x_0 + {}_1x_1 \cdot {}_0x_0$ and ${}_1f_0^{(2)} = {}_0x_1 \cdot {}_1x_1 + {}_0x_0 \cdot {}_1x_0$, then we have transformed the predictor $f_0^{(2)}$ into the two predictors ${}_0f_0^{(2)}$ and ${}_1f_0^{(2)}$. In the PFFN, when $f_0^{(2)}$ is selected as the predictor $X_2 = f_0^{(2)}$ in our new PBN, ${}_0f_0^{(2)}$ will produce the value of ${}_2x_0$, and ${}_1f_0^{(2)}$ produces a value for ${}_2x_1$. The probability of each of these new predictors is equal to the probability of the original predictor, so $P({}_0f_0^{(2)}) = P({}_1f_0^{(2)}) = 0.5$

In the same manner, we can convert the other predictor, $f_1^{(2)} = X_0 + X_1$ into two functions

${}_0f_1^{(2)} = {}_0x_0 + {}_1x_0$, and ${}_1f_1^{(2)} = {}_0x_1 + {}_1x_1$. The predictors for each variable can be collected into sets ${}_0F_2 = \{ {}_0f_0^{(2)} = {}_0x_1 \cdot {}_1x_1 + {}_0x_1 \cdot {}_1x_0 + {}_1x_1 \cdot {}_0x_0, {}_0f_1^{(2)} = {}_0x_0 + {}_1x_0\}$. Thus, for each variable X_i we will have two variables ${}_0x_i$ and ${}_1x_i$, from each set of predictors F_i we will have two sets, ${}_0F_i$ and ${}_1F_i$ of predictors.

Our completed example will thus be: $A = (V, F)$, we now have 8 nodes

$$V = \{ {}_0x_0, {}_0x_1, {}_1x_0, {}_1x_1, {}_2x_0, {}_2x_1, {}_3x_0, {}_3x_1\}$$

, with 8 sets of predictors

$$F = \{ {}_0F_0, {}_1F_0, {}_0F_1, {}_1F_1, {}_0F_2, {}_1F_2, {}_0F_3, {}_1F_3\}$$

, the predictors will be:

$$\begin{aligned}
{}_0F_0 &= \{ {}_0f_0^{(0)} = 0, {}_0f_1^{(0)} = 1\} \\
{}_1F_0 &= \{ {}_1f_0^{(0)} = 0, {}_1f_1^{(0)} = 0\} \\
{}_0F_1 &= \{ {}_0f_0^{(1)} = 0, {}_0f_1^{(1)} = 1\} \\
{}_1F_1 &= \{ {}_1f_0^{(1)} = 0, {}_1f_1^{(1)} = 1\} \\
{}_0F_2 &= \{ {}_0f_0^{(2)} = {}_0x_1 \cdot {}_1x_1 + {}_0x_1 \cdot {}_1x_0 + {}_1x_1 \cdot {}_0x_0, {}_0f_1^{(2)} = {}_0x_0 + {}_1x_0\} \\
{}_1F_2 &= \{ {}_1f_0^{(2)} = {}_0x_1 \cdot {}_1x_1 + {}_0x_0 \cdot {}_1x_0, {}_1f_1^{(2)} = {}_0x_1 + {}_1x_1\} \\
{}_0F_3 &= \{ {}_0f_0^{(3)} = {}_1x_0 \cdot {}_2x_0 + {}_1x_1 \cdot {}_2x_1 + {}_1x_0, {}_0f_1^{(3)} = {}_0x_0 + {}_1x_0\} \\
{}_1F_3 &= \{ {}_1f_0^{(3)} = {}_1x_0 \cdot {}_2x_1 + {}_1x_1 \cdot {}_2x_0 + {}_1x_1 \cdot {}_2x_1 + {}_1x_1, {}_1f_1^{(3)} = {}_0x_1 + {}_1x_1\}
\end{aligned}$$

We will select the jth predictor for the bit b, of gene i, with probability $P({}_bf_j^{(i)}) = {}_bc_j^{(i)} = 0.5$, as before. We have in a sense, split each gene into two independent bits.

4.2.2 Transforming the general case

In general, we can take a PFFN over GF(p^i) and split each node into i separate nodes. In the same manner, each predictor may be split into i component parts by taking a basis, and establishing a recurrence (such as $\alpha^2 = \alpha + 1$ which we used in the above example). This kind of recurrence is known to exist for any field, but the exact form depends on the field.

4.2.3 Restrictions on inputs

When building transcriptional networks, we may wish to place restrictions on the interactions between genes. For example, we will allow a transcription factor t to act on a gene g only if g has a transcriptional site for t. These types of restrictions can be imposed by restricting the form of allowed predictor functions. Since the available information on transcriptional regulation is incomplete, it is a challenge to incorporate information on allowed, prohibited, and mandatory regulatory interactions, and to do so in an efficient manner.

4.2.4 Remaining work

We will develop tools to perform reverse engineering of PFFN using the model outlined above, and test those tools on the CTA data set.

5 Ethical Issues

This research, like all research raises ethical issues. Two particularly controversial topics related to the research above are genetic testing and genetic engineering. Microarrays are already used or proposed to diagnose certain conditions. The accumulation of more microarray data and advances in analysis techniques would allow for screening for susceptibility to certain conditions. Access to private genetic information needs to be strictly protected, and analysis routines have to be designed to neither cause undue distress through false positives, nor a false sense of security due to false negative results. The reverse engineering problem in microarrays pretends to allow us to model cell processes leading to a condition. This knowledge could be used to target specific points in a regulatory network to prevent disease, or to produce a desired outcome in the cell.

The other major ethical issue is that efficient algorithms for reverse engineering genetic networks would likely be equally efficient at reverse engineering binary networks. Thus reverse engineering digital hardware or "black-box" software could be sped up by using our techniques. Similarly, many cryptographic protocols such as those protecting online commercial transactions are based on the difficulty of determining a specific key used in a cryptographic function to transform inputs into outputs. An efficient reverse engineering algorithm could be used to decrypt sensitive communications.

References

Akutsu, T., Kuahara, S., 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* .

Akutsu, T., Miyano, S. & Kuhara, S. (1999), 'Identification of genetic networks from a small number of gene expression patterns under the Boolean network model', *Pacific Symposium on Biocomputing* **4**, 17–28.

Aviñó, M., Green, E. & Moreno, O. (2004), 'Applications of finite fields to dynamical systems and reverse engineering problems', *Proceedings of the 19th ACM Symposium on Applied Computing - SAC* .

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

Chiesa, R., Ortiz-Zuazaga, H. G., Ge, H. & 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.

de Jong, H. (2002), 'Modeling and simulation of genetic regulatory systems: A literature review', *Journal of Computational Biology* **9**(1), 67–103.

D'haeseleer, P., Liang, S. & Somogyi, R. (2000), 'Genetic network inference: from co-expression clustering to reverse engineering.', *Bioinformatics* **16**(8), 707–726.

Eisen, M., Spellman, P., Botstein, D. & Brown, P. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proceedings of National Academy of Science* **95**, 14863–14867.

Ge, H., Chiesa, R. & Peña de Ortiz, S. (2003), 'Hzf-3 expression in the amygdala after establishment of conditioned taste aversion', *Neuroscience* **120**(1), 1–4.

Green, E. L. (2004), On polynomial solutions to reverse engineering problems. Pre-print.

Hubbard, T., Andrews, D., Caccamo, M., Cameron, G., Chen, Y., Clamp, M., Clarke, L., Coates, G., Cox, T., Cunningham, F., Curwen, V., Cutts, T., Down, T., Durbin, R., Fernandez-Suarez, X. M., Gilbert, J., Hammond, M., Herrero, J., Hotz, H., Howe, K., Iyer, V., Jekosch, K., Kahari, A., Kasprzyk, A., Keefe, D., Keenan, S., Kokocinsci, F., London, D., Longden, I., McVicker, G., Melsopp, C., Meidl, P., Potter, S., Proctor, G., Rae, M., Rios, D., Schuster, M., Searle, S., Severin, J., Slater, G., Smedley, D., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Storey, R., Trevanion, S., Ureta-Vidal, A., Vogel, J., White, S., Woodwark, C. & Birney, E. (2005), 'Ensembl 2005', *Nucleic Acids Res.* **33**, D447–D453. Database issue.

Ideker, T. E., Thorsson, V. & Karp, R. M. (2000), 'Discovery of regulatory interactions through perturbation: Inference and experimental design', *Pacific Symposium on Biocomputing* **5**, 302–313.

Kauffman, S. A. (1969), 'Metabolic stability and epigenesis in randomly constructed genetic nets', *J. Theor. Biol.* **22**, 437–467.

Kauffman, S. A. (1993), *The Origins of Order*, Oxford University Press, New York, Oxford.

Kela, J., Salmi, P., Rimondini-Giorgini, R., Heilig, M. & Wahlestedt, C. (2003), 'Behavioural analysis of melanin-concentrating hormone in rats: evidence for orexigenic and anxiolytic properties', *Regul. Pept.* **114**(2–3), 109–114.

Lamprecht, R., Hazvi, S. & Dudai, Y. (1997), 'cAMP response element-binding protein in the amygdala is required for long- but not short-term conditioned taste aversion memory', *J. Neurosci.* **17**, "8443–8450".

Laubenbacher, R. & Pareigis, B. (2001), 'Equivalence relations on finite dynamical systems', *Advances in Applied Mathematics* **26**, 237–251.

Laubenbacher, R. & Stigler, B. (2001), 'Dynamic networks', *Adv. in Al. Math.* **26**, 237–251.

Laubenbacher, R. & Stigler, B. (2003), A computational algebra approach to the reverse engineering of gene regulatory networks. http://arxiv.org/pdf/q-bio.QM/0312026.

Laubenbacher, R. & Stigler, B. (2004), Biochemical networks. Pre-print.

Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., Simon, I., Zeitlinger, J., Jennings, E., Murray, H., Gordon, D., Ren, B., Wyrick, J., Tagne, J., Volkert, T., Fraenkel, E., Gifford, D., & Young, R. (2002), 'Transcriptional regulatory networks in saccharomyces cerevisiae', *Science* **298**, 799–804.

Lemon, B. & Tjian, R. (2000), 'Orchestrated response: a symphony of transcription factors for gene control', *Genes and Development* **14**(20), 2551–2569.

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

Liang, S., Fuhrman, S. & Somogyi, R. (1998), 'REVEAL, a general reverse engineering algorithm for inference of genetic network architectures', *Pacific Symposium on Biocomputing* **3**, 18–29.

Merika, M. & Thanos, D. (2001), 'Enhanceosomes', *Curr Opin Genet Dev* **11**(2), 205–208.

Moreno, O., Bollman, D. & Aviñó, M. (2002), 'Finite dynamical systems, linear automata and finite fields', *2002 WSEAS Int. Conf. on System Science Alied Mathematics & Computer Science and Power Engineering Systems* pp. 1481–1483. Also to appear in the International Journal of Computer Research.

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

Patterson, D. A. & Hennessy, J. L. (1997), *Computer Organization and Design*, Morgan Kaufmann Publishers, San Francisco.

Pereira-da Silva, M., Torsoni, M., Nourani, H., Augusto, V., Souza, C., Gasparetti, A., Carvalheira, J., Ventrucci, G., Marcondes, M., Cruz-Neto, A., Saad, M., Boschero, A., Carneiro, E. & Velloso, L. (2003), 'Hypothalamic melanin-concentrating hormone is induced by cold exposure and participates in the control of energy expenditure in rats.', *Endocrinology* **144**(11), 4831–4840.

Rat Genome Database Web Site (2005), Rat genome data. Medical College of Wisconsin, Milwaukee, Wisconsin. World Wide Web (URL: `http://rgd.mcw.edu/`).

Robles, Y., Vivas, P. E., Ortiz-Zuazaga, H. G., Felix, Y. & Peña de Ortiz, S. (2003), 'Hippocampal gene expression profiling in spatial learning', *Neurobiology of Learning and Memory* **80**(1), 80–95.

Schena, M., Shalon, D., Davis, R. W. & Brown, P. O. (1995), 'Quantitative monitoring of gene expression patterns with a complementary DNA microarray', *Science* **270**(5235), 467–470.

Schug, J. & Overton, G. C. (1997), Tess: Transcription element search software on the www, Technical report, Computational Biology and Informatics Laboratory, School of Medicine, University of Pennsylvania. Technical Report CBIL-TR-1997-1001-v0.0.

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

Shmulevich, I., Dougherty, E. R. & Zhang, W. (2002), 'Gene perturbation and intervention in probabilistic boolean networks', *Bioinformatics* **18**(10), 1319–1331.

Suh, E. B., Dougherty, E. R., Kim, S., Bittner, M. L., Chen, Y., Russ, D. E. & Martino, R. (2002), Parallel computation and visualization tools for codetermination analysis of multivariate gene-expression relations, *in* W. Zhang & I. Shmulevich, eds, 'Computational and Statistical Approaches to Genomics', Kluwer, Boston, MA.

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

Yamamoto, T., Shimura, T., Sako, N., Yasoshima, Y. & Sakai, N. (1994), 'Neural substrates for conditioned taste aversion in the rat', *Behav. Brain Res.* **65**, 1231–137.

Yasoshima, Y., Shimura, T. & Yamamoto, T. (1995), 'Single unit responses of the amygdala after conditioned taste aversion in conscious rats', *Neuroreport* **6**, 2424–2428.