Inference of Gene Predictor Set Using Boolean Satisfiability

Pey-Chang Kent Lin, Sunil P Khatri
Texas A&M University
Department of Electrical & Computer Engineering
College Station TX 77843

***Abstract*— The inference of gene predictors in the gene regulatory network (GRN) has become an important research area in the genomics and medical disciplines. Accurate predicators are necessary for constructing the GRN model and to enable targeted biological experiments that attempt to validate or control the regulation process. In this paper, we implement a SAT-based algorithm to determine the gene predictor set from steady state gene expression data (attractor states). Using the attractor states as input, the states are ordered into attractor cycles. For each attractor cycle ordering, all possible predictors are enumerated and a conjunctive normal form (CNF) expression is generated which encodes these predictors and their biological constraints. Each CNF is solved using a SAT solver to find candidate predictor sets. Statistical analysis of the resulting predictor sets selects the most likely predictor set of the GRN, corresponding to the attractor data. We demonstrate our algorithm on attractor state data from a melanoma study [1] and present our predictor set results.**

I. Introduction

With increasing availability of gene expression data, the focus in computational biology has shifted to the understanding of gene regulation and its inter-relation with the biological system. The use of genome information has given rise to the possibility of "personalized medicine" – targeted and specific disease prevention and treatment based on individual gene information [2], [3]. The urgent applications to cancer and gene-related diseases calls for the genomics field to significantly improve the algorithms used for accurate inference of the gene regulatory network (GRN).

In an organism, the genome is a highly complex control system wherein proteins and RNA produced by genes and their products interact with and regulate the activity of other genes [4]. A *predictor* for a target gene g_i is the collection of genes directly participating in the regulation of gene g_i. As such, the predictor does not consider the type of regulation (repression versus activation), and is analogous to the *support* of a function in logic synthesis. Each gene has a single predictor (which is a collection of genes) and the *predictor set* is the set consisting of predictors of each gene in the GRN.

There are several observations that impact the formulation of our GRN model and predictor inference algorithm. First, the activity level (i.e. activation or repression) of all genes at a particular time t represents the *state* of the GRN at that time t. From our knowledge of biological systems, we observe that over time, cellular processes converge to sequences of stable *attractor* states. Some of these attractor states represent normal cellular phenomena in biology (i.e. cell cycle and division), while other attractor states are consistent with disease (i.e. metastasis of cancer). Second, the GRN is often inferred by observing microarray-based experimental data though which the activity level of genes is measured. Both observations of gene activity (or state) can be used to infer the gene regulation network. The disadvantage of using microarray data is that such studies do not involve controlled time-series experimental data. Hence the measurements are assumed to arise from cyclic sequences of gene expressions (attractor states) in steady state. Such a sequence is referred to as an *attractor cycle*. The GRN is then inferred from this data, using methods traditionally based on probabilistic transition models [5], [6].

As previously mentioned, it is necessary to determine the predictor set in order to reconstruct the GRN. However, there may exist many possible predictors for any gene, based on the attractor cycle data. Furthermore, only certain combinations of predictors may form a valid predictor set, due to biological constraints. The issue addressed in this paper is how to efficiently and deterministically select the predictors that form the predictor set. We have implemented a Boolean satisfiability (SAT) based algorithm for the inference of gene predictor sets. Satisfiability is a decision problem of determining whether the variables in a Boolean formula (expressed in Conjunctive Normal Form or CNF) can be assigned to make the formula evaluate to *true*. Although SAT is NP-complete, many SAT solvers have been developed to quickly and efficiently solve large SAT problems. Our algorithm takes advantage of a recent SAT solver to find the predictor set.

The basic outline of our SAT-based algorithm for predictor set inference is described briefly below. First, all possible orderings of attractor states are enumerated, yielding all possible attractor cycles. For each ordering, we enumerate all predictors that are logically valid, and create a CNF expression which encodes all these predictors and biological constraints (such as cardinality bounds on the predictors). A SAT solver is then used to find the valid candidate predictor sets. After this process is done iteratively for all attractor cycle (orderings), statistical analysis provides the most likely predictor set. Note that this paper does not claim to extract the GRN. Using the predictor set inferred by this paper, we plan to infer the GRN in a subsequent research effort.

The key contributions of this paper are:

- We develop a Boolean Satisfiability based approach to realize the gene predictor set from attractor state data.
- We modify an existing SAT-solver (MiniSat [7]) for efficient all-SAT computation, and further optimize the decision engine of MiniSat for improved predictor set inference.
- On gene expression data from a melanoma study [1], we apply our SAT-based algorithm and present the predictor set, including the predictor for the cancer gene WNT5a.
- Our approach can be used to find the predictor set for any gene related disease, provided attractor state data is available. The predictor set information obtained from our algorithm can be used by biologists to fine tune their gene expression experiments.

The remainder of this paper is organized as follows. Section II describes previous work in modeling the gene regulatory network and inference of gene predictors. Section III presents our FSM model and Boolean SAT approach. Section IV reports experimental results. Concluding comments and future work are discussed in Section V.

II. Previous Work

Several models have been proposed for modeling the GRN such as Markov Chains [8], [9], Coupled ODEs (ordinary differential equations), Boolean Networks [10], [11], Continuous Networks [12], and Stochastic Gene Networks [13]. This paper utilizes the Boolean Network (BN) model that was proposed by Kauffman in 1969 [10]. In a Boolean Network, the expression activity of a gene is represented as a binary value, where 1 indicates the gene is ON (active) and producing gene-products, while 0 indicates it is OFF. Such a model cannot capture the continuous and stochastic biochemical properties of protein and RNA production. However, genes can typically be modeled as ON or OFF in any particular biochemical pathway.

In [14], [15], the probabilistic modeling framework is represented using dynamic Bayesian networks and probabilistic Boolean networks (PBNs). The method proposed considers gene prediction using multinomial probit regression with Bayesian variable selection. Genes are selected which satisfy multiple regression equations, of which the strongest genes are used to construct the predictor set. The target gene is predicted based on the strongest genes, using the coefficient of determination to measure predictor accuracy.

Another method proposed by [16] also assumes a PBN model. A partial state transition table is constructed based on available attractor state data. From this state transition table, predictors with 3 or less regulating genes are selected for each target gene. All unknown values in the table are randomly set. The Boolean network is simulated for several

iterations using different starting states, observing whether the states eventually transition to an attractor cycle. If the simulation successfully transitions to an attractor cycle, the selected predictors are considered as a valid predictor set. This process is repeated, to build a collection of Boolean Networks which are combined to form a Probabilistic Boolean Network (PBN).

Our larger goal is to find a small number of *deterministic* GRNs, rather than a PBN. Towards this, we need to first find ways to accurately find the predictor set. This is the focus of this paper. Philosophically, our aim is to invest effort into accurate predictor set determination, so that the results can be used to find high quality deterministic GRNs.

III. Our Approach

This section describes our model and algorithm for inference of predictor sets using SAT. We begin with some logic synthesis definitions which are useful in understanding the application of SAT to GRNs and predictor selection. We then describe a simple example to explain the algorithm. Lastly, we generalize the algorithm for larger problem sets and comment on the complexity of the approach.

A. Definitions

Definition 1: A **literal** or a **literal function** is a binary variable x or its negation $\overline{x}$.

Definition 2: A **cube** is a product of a set of literal functions (example $xy\overline{z}$).

Definition 3: A **clause** is a disjunction (logical OR) containing literals (example $x + \overline{y} + \overline{z}$).

Definition 4: A **Conjunctive Normal Form (CNF)** expression consists of a conjunction (AND) of m clauses $c_1 \dots c_m$. Each clause c_i consists of disjunction (OR) of k_i literals.

A CNF formula is also referred to as a logical product of sums. Thus, to satisfy the formula (i.e. make it evaluate to *true*), each clause must have at least one literal evaluate to true.

Definition 5: **Boolean satisfiability (SAT)**. Given a Boolean formula S (on a set of binary variables X) expressed in Conjunctive Normal Form (CNF), the objective of SAT is to identify an assignment of the binary variables in X that satisfies S, if such an assignment exists.
For example, consider the formula $S(a, b, c) = (a + \overline{b}) \cdot (a + b + c)$. This formula consists of 3 variables, 2 clauses, and 4 literals. This particular formula is satisfiable, and a satisfying assignment is $(a, b, c) = (0, 0, 1)$, which can be expressed as the satisfying cube $\overline{a}\overline{b}c$.

There may exist many satisfying assignments for the formula in question. An extension of the SAT problem is to find *all* satisfying assignments (or All-SAT). One simple method to accomplish All-SAT is to run SAT on the formula S, express the satisfying assignment as a cube k, complement k to get a clause c, add c as a new clause of the formula, and run SAT again repeatedly. The inclusion of c in S ensures that the same cube k cannot be found as a satisfying assignment again. The process continues until no new solutions can be found. In the previous example, the satisfying cube $\overline{a}\overline{b}c$ is complemented and added as a new clause $(a + b + \overline{c})$ to the original formula. The new CNF is solved by SAT again (this is repeated until no new satisfying assignments are found).

Definition 6: A **predictor** $f_i = \{g_j, g_k, \cdots\}$ lists the set $\{g_j, g_k, \cdots\}$ of genes which regulate the activity of gene g_i.

Definition 7: The **predictor set** is the complete set of predictors $\{f_1, f_2, \cdots, f_n\}$ for the GRN with n genes $g_1, g_2, \cdots, g_n$.

B. Implementation and Example

Given gene expression data (a set of unordered attractor states) as input, we would like to determine the best predictor set. We first present an outline of our SAT-based algorithm, and then explain the steps through a simple example.

The algorithm has three main steps.

- First, attractor states are ordered into attractor cycles in all possible ways. For each possible ordering of attractor states into attractor cycles, all possible predictors are found and a CNF is generated encoding valid predictor sets.
- Second, the CNF is solved for All-SAT, recording all satisfying cubes. Each cube corresponds to a predictor set. The first two steps are repeated for all attractor cycle orderings.
- Finally, statistical analysis on the SAT results determines the most frequent (likely) predictor set for the GRN.

Present state			Next state		
x_1	x_2	x_3	y_1	y_2	y_3
0	1	0	1	1	0
1	1	0	0	1	0
1	1	1	1	1	1

TABLE I
EXAMPLE STATE TRANSITION TABLE

To illustrate the SAT-based algorithm, we apply it to a simple example with three genes (g_1, g_2, g_3) and gene expression data with three lines $(010, 110, 111)$. The present state of these genes is represented by the variables $< x_1, x_2, x_3 >$ and the next state is represented by the variables $< y_1, y_2, y_3 >$. We assume each line was measured in steady state and therefore is an attractor state.

Step 1: We order (or arrange) the attractor states into attractor cycles for which there are six possibilities. One ordering is with each attractor state transitioning to itself with a self-edge, resulting in three singleton attractor cycles. Two possible orderings result when all three attractor states form a single attractor cycle of length three. The last three possible orderings have two attractor cycles, one cycle with length two and the other cycle of length one. We focus our example on an ordering with two attractor cycles, as shown in Table I.

For each valid attractor cycle ordering, a *partial state transition table* is constructed, containing the attractor states. Table I shows the partial state transition table for the example attractor cycle ordering. To find all valid predictors of a gene, each next state column is checked against all combinations of the present state columns. For example, let us explore gene g_2 and g_3 as a predictor for gene g_1. For gene g_1, the next state bit is y_1, while for gene g_2 and g_3, the present state bits are x_2 and x_3. In the first two rows of TableI, $< x_2, x_3 >= 10$. However, in row 1, $y_1 = 1$, while in row 2, $y_1 = 0$, which forms a contradiction (since the same input cannot result in different outputs). Therefore, gene g_1 cannot be predicted by genes g_2 and g_3.

Now, consider genes g_1 and g_3 as a predictor for gene g_1. There is no contradiction, and the combination is logically valid. Thus one possible predictor for gene g_1 is $f_1 = \{x_1, x_3\}$. All valid predictors with P (user-defined) or less inputs are exhaustively searched and recorded for CNF formulation (which is done in the next step). In our example, gene g_1 has 2 possible predictors $\{x_1, x_3\}$, $\{x_1, x_2, x_3\}$ which we label v_1^1, v_2^1 respectively. We assume that a gene cannot self-regulate, so $\{x_1\}$ by itself is not a valid predictor.

Step 2: After all predictors are found for each gene, we generate the SAT formula which encodes logically valid predictor sets. The j^{th} predictor for gene i is assigned a variable v_j^i. Gene g_1 in our example will have two predictor variables $v_1^1 \equiv \{x_1, x_3\}$, $v_2^1 \equiv \{x_1, x_2, x_3\}$. Gene g_2 and g_3 will have their own corresponding predictor variables $v_1^2 \equiv \{x_1, x_2\}$, $v_2^2 \equiv \{x_1, x_3\}$, $v_3^2 \equiv \{x_2, x_3\}$, $v_4^2 \equiv \{x_1, x_2, x_3\}$ and $v_1^3 \equiv \{x_1, x_3\}$, $v_2^3 \equiv \{x_2, x_3\}$, $v_3^3 \equiv \{x_1, x_2, x_3\}$ respectively. There are three constraints that we incorporate while constructing the CNF that encodes valid predictor sets. The conjunction of these constraints forms our final CNF.

1) The first constraint (S_1) is that all genes in the GRN must have a predictor. In other words, we assume that all genes are highly correlated and are "participating" in the GRN. For gene i, all of its associated predictor variables are written in a single clause $c_i^1 = (v_1^i + \cdots + v_j^i)$. In our example, for g_1, $c_1^1 = (v_1^1 + v_2^1)$. For g_2 and g_3, we have $c_2^1 = (v_1^2 + v_2^2 + v_3^2 + v_4^2)$ and $c_3^1 = (v_1^3 + v_2^3 + v_3^3)$ respectively.
   To satisfy any c_i^1 clause, at least one predictor in the clause must be chosen. To ensure that at least one predictor is chosen for all genes, we write the conjunction of all c_i^1 clauses. $S_1 = c_1^1 \cdot c_2^1 \cdot c_3^1$
2) The second constraint (S_2) specifies that for each gene, exactly one predictor is chosen. The assumption is that a gene cannot have multiple predictors. To formulate the clauses c_i^2 for gene i, smaller clauses are formed from all pairs of combinations of its predictors $v_{1\dots j}^i$. In each of these clauses of pairs of variables, both predictor variables are complemented. Hence, $c_1^2 = (\overline{v_1^1} + \overline{v_2^1})$,

	PIRIN x_1	S100P x_2	RET1 x_3	MART1 x_4	HADHB x_5	STC2 x_6	WNT5A x_7
BAD	0	0	0	0	0	1	1
	0	0	1	1	1	1	1
	1	0	1	0	0	0	1
GOOD	0	1	0	0	0	0	0
	0	1	1	1	0	0	0
	1	0	1	1	1	1	0
	1	1	0	1	1	0	0

TABLE II

ATTRACTORS FOR MELANOMA NETWORK

$c_2^2 = (\overline{v_1^2}+\overline{v_2^2})\cdot(\overline{v_1^2}+\overline{v_3^2})\cdot(\overline{v_1^2}+\overline{v_4^2})\cdot(\overline{v_2^2}+\overline{v_3^2})\cdot(\overline{v_2^2}+\overline{v_4^2})\cdot(\overline{v_3^2}+\overline{v_4^2})$, and $c_3^2 = (\overline{v_1^3}+\overline{v_2^3})\cdot(\overline{v_1^3}+\overline{v_3^3})\cdot(\overline{v_2^3}+\overline{v_3^3})$

Any selection of two or more predictors for gene i will result in the clauses of c_i^2 becoming unsatisfiable. The c_i^1 clause ensures that at least one predictor will be chosen for gene i, and c_i^2 forces the selection of exactly one predictor for gene i. The conjunction of all c_i^2 clauses forms the constraint S_2, which forces SAT to choose only one predictor per gene. $S_2 = c_1^2 \cdot c_2^2 \cdot c_3^2$

3) The last constraint (S_3) requires that each gene must be used as a predictor for at least one other gene in the predictor set. A gene that is not used in any predictor does not perform any regulation function and could be removed from the GRN. S_3 ensures that this does not occur. To ensure that gene g_i is used in at least one predictor, we form clauses c_i^3 which include all predictors that use gene g_i as input. To specify that gene g_i must be used, we also include a single variable clause (x_i) to c_i^3. For gene g_1, $c_1^3 = (x_1)\cdot(\overline{x_1}+v_1^1+v_2^1+v_1^2+v_2^2+v_4^2+v_1^3+v_3^3)$. $c_2^3 = (x_2)\cdot(\overline{x_2}+v_2^1+v_1^2+v_3^2+v_4^2+v_2^3+v_3^3)$ and $c_3^3 = (x_3)\cdot(\overline{x_3}+v_1^1+v_2^1+v_2^2+v_3^2+v_4^2+v_1^3+v_2^3+v_3^3)$ for gene g_2 and g_3 respectively. To satisfy these clauses, x_i and at least one other predictor variable in the second clause of c_i^3 must be selected. Finally, $S_3 = c_1^3 \cdot c_2^3 \cdot c_3^3$

The final SAT formula S as a conjunction of the S_i formulas. $S = S_1 \cdot S_2 \cdot S_3$

Step 3: The SAT solver performs an All-SAT on S. The satisfying cubes (each cube encodes a candidate predictor set) from the All-SAT output are collected. The process is repeated for the remaining attractor cycle orderings. From the results, we find the most likely predictors based on the frequency of occurrence of the predictors across all orderings. Three methods are used to analyze the statistical results, which will be described in Section IV.

In general, the above algorithm can be applied to input data for N genes and A attractor states. The total number of attractor state orderings is $A!$. For each ordering, there can be up to $O(N^3)$ predictors per gene. The SAT search space per ordering is on the order of $O(2^{N^3})$, resulting in overall complexity of $O(A!2^{N^3})$. Typically, the number of attractor states A recorded through gene expression measurements is small. As such, $A!$ is thus much smaller than $2^{(N^3)}$, so the runtime complexity is dominated by the All-SAT operation. For pragmatic reasons, our algorithm stops each All-SAT after T minutes (or C cubes), where T or C is defined by the user.

IV. Experimental Results

To evaluate our SAT-based algorithm for inferring gene predictors, the algorithm was tested on gene-expression data from a melanoma study done by Bittner and Weeraratna [1]. In the melanoma study, it was observed that an abundance of RNA (expression) for gene WNT5A was associated with a high metastasis of melanoma. The study measured 587 genes with 31 gene expression patterns (lines). Seven genes are believed to be closely knit: PIRIN, S100P, RET1, MART1, HADHB, STC2, and WNT5A. There are 18 distinct patterns, which were reduced to seven using Hamming-distance of one, in Table II. These seven lines form the attractor states which are the input to our algorithm.

For the experiments, we assume two additional specifications. First, we divide attractor states into good and bad states, based on the presence of WNT5A. We allow good attractor states to cycle only to other good attractor states, and bad attractor states can only cycle to other bad attractor states. Second, we limit the maximum attractor cycle length L to 3, and the maximum number of predictor inputs P to 3, because long attractor cycles and large predictor inputs are highly complex and less likely to occur in biological systems [17], [4].

Our algorithm utilizes a modified open-source and highly efficient exact SAT-solver called MiniSAT v1.14 [18], [7]. All-SAT operations were limited to a 30 minute time-out. On average, each All-SAT run yielded 10K satisfying cubes in this duration. Our algorithm was implemented and run on a Pentium 4 Linux machine with 4GB RAM. MiniSat [7], was originally designed to find a single satisfying assignment. We modified MiniSat to perform All-SAT as described in Section III. We further modified MiniSat to always randomly select decision variables during the solving process to increase the activity of all variables. With the second modification, we find that a reduced runtime of 30 minutes is sufficient to achieve an average of $\leq$ 5% difference in the predictors' occurrence frequency compared to the full All-SAT results (without the modification yields 8/

The following presents our results after collection of All-SAT results from all valid attractor cycle orderings. In Figure 1, we display a histogram of all logically valid predictors and their frequency of occurrence, across all attractor orderings. In the sequel, a predictor label of 2367 means that gene g_2 is predicted by genes g_3, g_6, and g_7. From this chart, we can observe that certain predictors occur with significantly higher frequency than others. For example with gene g_1, the predictor $\{x_3, x_5, x_7\}$ (PIRIN predicted by RET1, HADHB, WNT5A) occurs with much higher frequency than all other predictors for gene g_1. This indicates that this predictor is most likely to be present in the final predictor set.

From this data, we propose three methods (A, B, AB) for selecting the predictor set. In **method A**, a predictor histogram is created as in Figure 1. From the histogram, for each gene g_i, we find its predictor p_j^i such that p_j^i is the most frequently occurring predictor of gene g_i and the *resolution ratio* R_i of this predictor (defined as the ratio of the occurrence frequency of p_j^i to the occurrence frequency of the next most frequently occurring predictor of gene g_i) is maximum. Among all genes, we choose the one with the highest resolution ratio, and select its most frequently occurring predictor as its final predictor. After selecting this final predictor, we regenerate the histogram, discarding any candidate predictor sets that do not contain the final predictor(s) that have been selected in previous steps. The process repeats until all genes have a single final predictor. The set of final predictors of all genes forms the predictor set. The advantage of method A is that at every iteration, we select real predictors that have a high overall occurrence in the solution. However the method may have problems selecting final predictors if the resolution ratio is low (i.e. when the frequencies of occurrence of the predictors are nearly identical).

As an alternative, **method B** is proposed, to determine for each gene i, how likely it is that gene g_i will predict the other genes in the GRN. In other words, we ask what is the occurrence frequency of x_i in the predictors of f_j. Table III shows in entry (i, j) how frequently a gene g_i is used to predict a gene g_j. This table is populated by summing the occurrence frequency of all predictors of g_j that have gene g_i as one of their inputs. As such, any entry can be $\geq$1, and is a measure of the usefulness of g_i as a predictor for g_j. The predictor of g_j is determined by finding, for each column j of Table III, the three largest entries and adding their values. Suppose we call this sum s_j (the resolution score of column j). We compute the resolution score for all columns and select the final predictor for the column with the highest resolution score. This final predictor is formed by listing the 3 input genes that correspond to the 3 entries that were used to compute the highest resolution score. Similar to method A, we reiterate the process by regenerating the table after discarding all predictor sets that do not contain predictors that were selected in previous steps. Method B has the advantage of being more robust when no *single* predictor has a significantly higher occurrence frequency than others. However, there is no guarantee that the predictor selected by method B is a valid predictor. If this happens, we select the column with the next highest resolution score.

In our experiments, we also use a hybrid **method AB** which works in the following manner. Both methods A and B are used to select their best predictor. If both methods produce the same predictor f_i, we select this predictor as a final predictor. If not, we list the best predictors for each gene, for both methods. If multiple predictors match for both methods, we choose the final predictor as the one with the highest weighted sum of the resolution ratio and resolution score. The resolution ratio is weighted by 0.3 and the resolution score is weighted by 0.7. The weighting factor for the resolution ratio is lower since the resolution ratio values of any gene are often close to 1. In such a situation, we

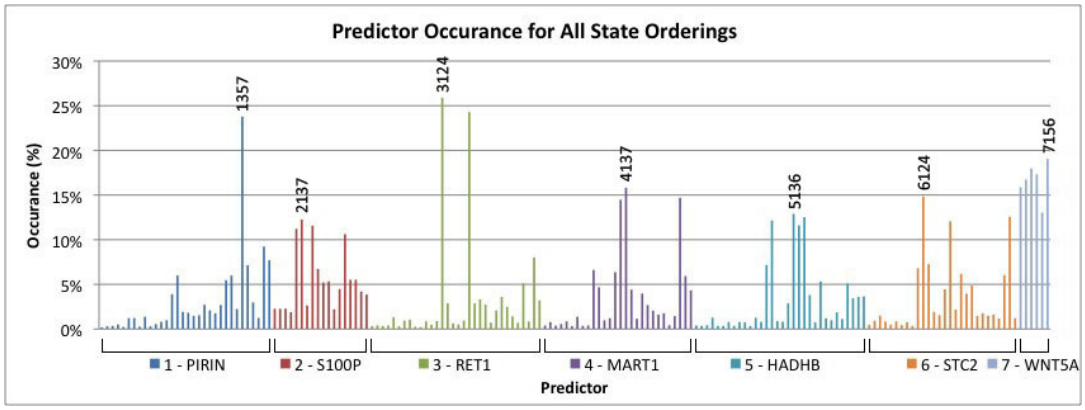

Fig. 1. Method A: Predictor occurrence for all valid attractor cycle orderings (first iteration: no predictor selected)

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
x_1		0.59	0.68	0.57	0.69	0.60	1.00
x_2	0.24		0.41	0.29	0.33	0.49	0.51
x_3	0.65	0.48		0.76	0.58	0.56	0.17
x_4	0.39	0.40	0.78		0.54	0.44	0.29
x_5	0.56	0.30	0.27	0.44		0.39	0.36
x_6	0.42	0.54	0.52	0.41	0.44		0.67
x_7	0.64	0.63	0.24	0.48	0.32	0.45	

TABLE III

METHOD B: GENE OCCURRENCE FOR ALL PREDICTORS (FIRST ITERATION)

		PIRIN x_1	S100P x_2	RET1 x_3	MART1 x_4	HADHB x_5	STC2 x_6	WNT5A x_7
A	Predictor set	1357	2137	3146	4357	5124	6124	7124
	Resolution ratio	2.57	1.41	1.34	1.30	1.41	1.66	1.31
B	Predictor set	1357	2137	3146	4137	5134	6137	7126
	Resolution score	1.78	1.77	1.84	1.97	1.99	1.98	2.56
AB	Predictor set	1357	2367	3146	4137	5137	6357	7124
	Weighted sum	2.06	1.57	1.75	1.61	1.45	1.39	1.88

TABLE IV

PREDICTOR SET SELECTION

would like to favor method B. If no predictor is produced by the previous step, we look at the top five predictors of method A for each gene and calculate the weighted sum of their resolution ratio and resolution score. The predictor with the highest weighted sum is selected as the final predictor. The process is reiterated, regenerating the histogram and table at each step, discarding any predictor sets that do not contain any of the previously selected final predictors. With this combined approach, we are able to select predictors with a higher degree of confidence and robustness.

We process our All-SAT data from melanoma attractor data of [1] using methods A, B, and AB. Results are shown in Table IV and shows what predictor was selected for each gene and the accompanying resolution ratio, resolution score, or weighted sum.

From the results, we can draw several conclusions:

- The iterative steps in regenerating the histogram (or table) retain only cubes (predictor sets) that contain previously selected final predictors. Hence the final predictor set from *each* method is a valid satisfying cube of the SAT formula S.
- The final predictor set is present in a select number of attractor cycle orderings. For example, the final predictor set selected by methods A, B, and AB are found in respectively 8, 4, and 6 attractor cycle orderings out of the total 5040 possible orderings. Hence the algorithm will enable us to generate a few deterministic GRNs.
- Some predictors are common among the predictor sets between the three methods. For example, all three methods select $f_1 = \{g_3, g_5, g_7\}$ (PIRIN predicted by RET1, HADHB, WNT5A) as well as $f_3 = \{g_1, g_4, g_6\}$. We can conclude this predictor is highly likely to be a final predictor in the GRN. Also, a majority of the predictors selected by the three method share common input genes. For example, the predictor selected by all methods for gene g_2 (S100P) contain 2 common genes $\{g_3, g_7\}$ (RET1, WNT5A), indicating these 2 genes are likely to be contained in the final predictor of f_2. Similarly f_7 has w common genes g_1 and g_2 for all methods.
- Using the above results, biologists can target their research on gene regulation and control, focusing on the gene relationships determined by the predictor set results.

V. Conclusions

Determining the predictor set for a gene regulatory network is important in many applications, particularly inference and control of the GRN. In this paper, we formulate gene predictor set inference as an instance of Boolean satisfiability. In our approach, we determine all possible orderings of attractor state data, generate the CNF encapsulating predictor and biological constraints, and apply a highly-efficient and modified SAT solver to find candidate predictor sets. The SAT results are analyzed using three selection methods to produce the final predictor set. We have tested our algorithm on attractor state data from a melanoma study, and determined the predictor sets for this GRN.

References

[1] M. Bittner et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 3, pp. 536–540, 2000.

[2] W. Burke and B. M. Psaty, "Personalized Medicine in the Era of Genomics," *JAMA*, vol. 298, no. 14, pp. 1682–1684, 2007.

[3] M. Teutsch et al., "The evaluation of genomic applications in practice and prevention (EGAPP) initiative: methods of the EGAPP working group," *Genetics in Medicine*, vol. 11, no. 1, pp. 3–14, 2009.

[4] N. Guelzim et al., "Topological and causal structure of the yeast transcriptional regulatory network," *Nature Genetics*, vol. 31, pp. 60–63, 2002.

[5] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219 – 2235, 2000.

[6] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information-theoretic criteria," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 5, no. 2, pp. 262–274, 2008.

[7] "Minisat." http://minisat.se/.

[8] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?," *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.

[9] G. Vahedi, B. Faryabi, J.-F. Chamberland, A. Datta, and E. Dougherty, "Intervention in gene regulatory networks via a stationary mean-first-passage-time control policy," *Biomedical Engineering, IEEE Transactions on*, vol. 55, pp. 2319 –2331, oct. 2008.

[10] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437 – 467, 1969.

[11] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. Philadelphia, PA: SIAM – Society for Industrial and Applied Mathematics, 2009.

[12] N. Geard and J. Wiles, "A gene network model for developing cell lineages," *Artif. Life*, vol. 11, no. 3, pp. 249–268, 2005.

[13] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells," *Genetics*, vol. 149, pp. 1633–1648, 1998.

[14] X. Zhou, X. Wang, and E. R. Dougherty, "Gene prediction using multinomial probit regression with Bayesian gene selection," *EURASIP Journal on Applied Signal Processing*, pp. 115–124, 2004.

[15] W. Zhou, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics*, vol. 17, pp. 2129–2135, 2006.

[16] R. Pal, I. Ivanov, A. Datta, M. L. Bittner, and E. R. Dougherty, "Generating Boolean networks with a prescribed attractor structure," *Bioinformatics*, vol. 21, no. 21, pp. 4021–4025, 2005.

[17] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 ed., June 1993.

[18] N. Een and N. Sorensson, *An Extensible SAT-solver*. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2004.