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 predictors 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 regulations. The CNF expression is then solved for the possible 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 of 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 is the collection of genes directly participating in the regulation of gene. 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 represents the state of the GRN at that time. 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 logical 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 $x + \overline{y} + z$).

**Definition 4:** A Conjunctive Normal Form (CNF) expression consists of a conjunction (AND) of $m$ clauses $c_1 \ldots c_m$. 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 $S(a, b, c) = (a + b) \cdot (a + b + c)$. 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 $abc$.

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 $abc$ 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 $f_i = (g_{i1}, g_{i2}, \ldots)$ lists the set $\{g_{i1}, g_{i2}, \ldots\}$ of genes which regulate the activity of gene $g_i$.

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

#### 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.

#### Example State Transition Table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table I**

- 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.

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 Table I, $< x_2, x_3 > =$ 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 $g_1$, $f_1 = \{x_1, x_3\}$. 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 $v_{ij}$. Gene $g_i$ 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 $v_1^2 \equiv \{x_1, x_2\}, v_2^2 \equiv \{x_1, x_3\}, v_3^2 \equiv \{x_2, x_3\}, v_1^3 \equiv \{x_1, x_2, x_3\}$ and $v_2^3 \equiv \{x_1, x_2\}, v_3^3 \equiv \{x_1, x_3\}, v_1^4 \equiv \{x_1, x_2, x_3\}$ 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 $c_i^1 = v_{i1}^1 + v_{i2}^1$. In our example, for $g_1$, $c_1^1 = v_{11}^1 + v_{12}^1$, for $g_2$ and $g_3$, we have $c_2^1 = v_{21}^2 + v_{22}^2 + v_{23}^2$ and $c_3^1 = v_{31}^3 + v_{32}^3$. 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 $v_{ij}^k$. In each of these clauses of pairs of variables, both predictor variables are complemented. Hence, $c_i^2 = (v_{ij}^k + v_{ij}^k)$. 3. The third constraint ($S_3$) ensures that each predictor is chosen for each gene, exactly once.
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 the following 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 ≤ 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 g2 is predicted by genes g3, g0, and g7. From this chart, we can observe that certain predictors occur with significantly higher frequency than others. For example with gene g1, the predictor \((x_2, x_3, x_7)\) (PIRIN predicted by RET1, HADHB, WNT5A) occurs with much higher frequency than all other predictors for gene g1. 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 gi, we find its predictor \(p_i\) such that \(p_i\) 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 \(p_i\) to the occurrence frequency of the next most frequently occurring predictor of gene \(g_i\)). 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 the final predictors, we generate 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 it is as the frequency of occurrence of the predictors is 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 \(f_i\). 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, each entry can be ≥ 1 and gives us an indication 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 method 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
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 repeated, 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.
- The final predictor set is present in a select number of attractor cycle orderings out of the total 5040 possible orderings. Hence the process instructed, 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.

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


