Chapter 2
Predictor Set Inference using SAT

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 chapter, 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 [1], [2] on attractor state data from a melanoma study [3] and present our predictor set results.\(^1\)

2.1 Background

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 [4], [5]. 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 [6]. A predictor for a target gene \(g_i\) is the collection of genes directly

\(^1\) Part of the data reported in this chapter is reprinted with permission from “Inference of Gene Predictor Set Using Boolean Satisfiability” by Pey-Chang Kent Lin, Sunil P. Khatri. *IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS) 2010*, Nov. 2010, pp. 1–4, Copyright 2010 by IEEE.
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. the metastasis of cancer). Second, the GRN is often inferred by observing microarray-based experimental data though which the activity level of genes is measured. The 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 [7], [8].

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 chapter 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 the algorithm in this chapter does not claim to extract the GRN. Using the predictor set inferred by this chapter, we can infer the GRN in a subsequent step.
The key contributions of this chapter 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 [9]) for efficient all-SAT computation, and further optimize the decision engine of MiniSat for improved predictor set inference.
- Using the gene expression data from a melanoma study [3], 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.

2.2 Previous Work

In the context of predictor set inference, [10], [11] use dynamic Bayesian networks and probabilistic Boolean networks (PBNs). The GRN is then inferred from this data, using methods traditionally based on probabilistic transition models [7], [8]. 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 [12] 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. The key reason for this is that combining BNs into PBNs admits new behavior, which may be incorrect. For example, the BNs in Fig. 2.1a and b yield a PBN in Fig. 2.1 which admits a new behavior $S_0 \rightarrow S_1 \rightarrow S_0 \rightarrow S_1$. Towards this, we need

![Fig. 2.1 Combination of BNs into PBN](image-url)
to first find ways to accurately find the predictor set. This is the focus of this chapter. 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.

2.3 Background

This section describes our background and problem definition for inference of predictor sets using SAT. We begin with some GRN definitions and then explain some of the biological constraints that will be used in our formation for the next section.

**Definition II.1:** 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 II.2:** The **predictor set** is the complete set of predictors \( \{f_1, f_2, \cdots, f_n\} \) for the GRN with \( n \) genes \( g_1, g_2, \cdots, g_n \).

Based on the gene products of one or more genes in a set \( f_i \), a gene \( g_i \) can become repressed or activated. In this case \( f_i \) is said to be predictor of gene \( g_i \). A predictor for 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. Each gene has a single predictor and the predictor set is the set consisting of predictors of each gene.

Note, we can relate these terms to logic synthesis: the predictor is identical to the logical support of a logic function, while the predictor set is akin to the set representing the support of all variables. The Boolean network GRN then is the complete logic circuit including function for each node.

**Definition II.3:** Given a starting state, within a finite number of steps, the network will transition as determined by the gene functions into a cycle of states, called an **attractor cycle**. States in an attractor cycle are called **attractor states**. The attractor cycle represents the long term behavior of the network, and absent perturbation, a network that has transitioned to an attractor cycle will continue to cycle thereafter.

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. the metastasis of cancer).

Second, the GRN is often inferred by observing microarray-based experimental data though which the activity level of genes is measured. The 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**.
Table 2.1 Example 3-Gene partial state transition table

<table>
<thead>
<tr>
<th>Current state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$ $x_2$ $x_3$</td>
<td>$y_1$ $y_2$ $y_3$</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

2.4 Problem Formulation and SAT Formulation

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.

1. **SAT Formula Construction for Predictor Set:** In this step, 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.

2. **All-SAT:** Each attractor ordering from step 1 generates a CNF which is solved for All-SAT. All satisfying cubes are recorded, where each satisfying cube corresponds to a predictor set. The first two steps are repeated for all attractor cycle orderings.

3. **Predictor Set Selection:** Statistical analysis on the All-SAT results determines the most frequent (likely) predictor set for the GRN. This step is explained in Sect. 2.5.

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.

We order (or arrange) the attractor states into attractor cycles for which there are six possibilities for our example. 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 ($010 \rightarrow 110 \rightarrow 010 \rightarrow \ldots$) and ($111 \rightarrow 111 \rightarrow \ldots$) as shown in Table 2.1.

### 2.4.1 Partial State Transition Table

For each valid attractor cycle ordering, a partial state transition table is constructed, containing the attractor states. Table 2.1 shows the partial state transition table for
the example attractor cycle orderings. To find all valid predictors of a gene, each next state column is checked against all combinations of the current (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 current (present) state bits are $x_2$ and $x_3$. In the first two rows of Table 2.1, $<x_2, x_3>=10$. 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 $v^1_1$, $v^1_2$ respectively. We assume that a gene cannot self-regulate, so $\{x_1\}$ by itself is not a valid predictor.

### 2.4.2 SAT Formulation and GRN Constraints

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 has two predictor variables $v^1_1\equiv\{x_1, x_3\}$, $v^1_2\equiv\{x_1, x_2, x_3\}$. Gene $g_2$ and $g_3$ will have their own corresponding predictor variables $v^2_1\equiv\{x_1, x_2\}$, $v^2_2\equiv\{x_1, x_3\}$, $v^3_2\equiv\{x_2, x_3\}$ and $v^3_3\equiv\{x_1, x_2, x_3\}$ 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

   \[ c^i_1 = (v^1_d + \cdots + v^j_d) \]

   In our example, for $g_1$, $c^1_1 = (v^1_1 + v^1_2)$. For $g_2$ and $g_3$, we have $c^2_1 = (v^2_1 + v^2_2 + v^2_3 + v^2_4)$ and $c^3_1 = (v^3_1 + v^3_2 + v^3_3)$ 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 as $S_1$ (Eq. 2.1).

   \[ S_1 = c^1_1 \cdot c^2_1 \cdot c^3_1 \]

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 $c^2_i$ for gene $i$, smaller clauses are formed from all pairs of combinations of its predictors $v^i_1 ... j$. In each of these clauses of pairs of variables, both predictor variables are complemented.

$$c^2_1 = (\overline{v^1_1} + \overline{v^1_2})$$

$$c^2_2 = (\overline{v^1_1} + \overline{v^2_2}) \cdot (\overline{v^1_2} + \overline{v^2_3}) \cdot (\overline{v^1_3} + \overline{v^2_4}) \cdot (\overline{v^2_3} + \overline{v^3_2})$$

$$c^2_3 = (\overline{v^3_1} + \overline{v^3_2}) \cdot (\overline{v^3_2} + v^3_3)$$

Any selection of two or more predictors for gene $i$ will result in the clauses of $c^2_i$ becoming unsatisfiable. The $c^1_i$ clause ensures that at least one predictor will be chosen for gene $i$, and $c^2_i$ forces the selection of exactly one predictor for gene $i$. The conjunction of all $c^2_i$ clauses forms the constraint $S_2$ (Eq. 2.2), which forces SAT to choose only one predictor per gene.

$$S_2 = c^1_i \cdot c^2_i \cdot c^3_i$$ \hspace{1cm} (2.2)

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 $c^3_i$ 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 $c^3_i$. For gene $g_1$, $g_2$, and $g_3$, we create the following clauses $c^3_1$, $c^3_2$, and $c^3_3$ respectively:

$$c^3_1 = (x_1) \cdot (\overline{x_1} + v^1_1 + v^1_2 + v^2_1 + v^2_2 + v^3_1 + v^3_2)$$

$$c^3_2 = (x_2) \cdot (\overline{x_2} + v^1_2 + v^1_1 + v^2_3 + v^2_4 + v^3_3)$$

$$c^3_3 = (x_3) \cdot (\overline{x_3} + v^1_1 + v^1_2 + v^2_3 + v^2_4 + v^3_1 + v^3_2 + v^3_3)$$

To satisfy these clauses, $x_i$ and at least one other predictor variable in the second clause of $c^3_i$ must be selected. $S_3$ is a conjunction of all the $c^3_i$ clauses (Eq. 2.3).

$$S_3 = c^3_1 \cdot c^3_2 \cdot c^3_3$$ \hspace{1cm} (2.3)

The final SAT formula $S$ as a conjunction of the $S_i$ formulas (Eq. 2.4).

$$S = S_1 \cdot S_2 \cdot S_3$$ \hspace{1cm} (2.4)
2.4.3 All-SAT

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 the next section.

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.

2.5 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 [3]. 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 2.2. 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 [13], [6].

Our algorithm utilizes a modified open-source and highly efficient exact SAT-solver called MiniSat v1.14 [14], [9]. All-SAT operations were limited to a 30 minute time-out. On average, each All-SAT run yielded 10 K satisfying cubes in this duration. Our algorithm was implemented and run on a Pentium 4 Linux machine with 4 GB RAM. MiniSat [9], was originally designed to find a single satisfying assignment. We modified MiniSat to perform All-SAT as MiniSat normally
2.5 Experimental Results

Table 2.2 Attractors for melanoma network

<table>
<thead>
<tr>
<th>PIRIN</th>
<th>S100P</th>
<th>RET1</th>
<th>MART1</th>
<th>HADHB</th>
<th>STC2</th>
<th>WNT5A</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_1</td>
<td>x_2</td>
<td>x_3</td>
<td>x_4</td>
<td>x_5</td>
<td>x_6</td>
<td>x_7</td>
</tr>
<tr>
<td>BAD</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GOOD</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 2.2 Average predictor error difference on melanoma attractor data using MiniSat without modification

only returns one SAT result. We further modified MiniSat to always randomly select decision variables during the solving process to increase the activity of all variables. The unaltered MiniSAT uses a heuristic for selecting the next decision variables. However, this heuristic results in many of the same variables being chosen over iterative runs of MiniSat. To increase the activity of all variables, we change the random variable frequency of MiniSat to 100% (from 2% in the unaltered MiniSat code). This forces MiniSAT to always choose a random variable on every variable-branch decision. A random variable frequency of $f\%$ means that MiniSat selects the next variable randomly $f\%$ of the time.

To demonstrate the quality of predictor selection using our modified All-SAT, our algorithm was run on four selected attractor cycle orderings (labeled 10, 721, 744, and 849) using melanoma data from [3]. The All-SAT operation was allowed to run for 12 h (which approximates a complete All-SAT run). In the case of attractor cycle order 721, all cubes were found. In Figs. 2.3 and 2.2, we compare the
average difference in all the predictors’ occurrence frequency in the complete All-SAT result with the results obtained with shorter All-SAT runtimes (10, 30, 60, and 120 min). Figure 2.3 shows the average error difference of all predictors’ frequency for the four orderings, using MiniSat with the random variable selection modification (100% random variable frequency), while Fig. 2.2 shows the same results without random variable selection (2% random variable frequency). Across the four orderings analyzed, the average error difference of all predictors’ occurrence frequency (shown in Figs. 2.3 and 2.2) is significantly lower using the random variable selection modification than without. Furthermore, the average error difference decreases with increasing runtime when using random variable selection. From this experiment, we determine that 30 minutes with random variable selection was sufficient to achieve an average of \( \leq 5 \% \) difference in the predictors’ occurrence frequency compared to the full All-SAT results.

The following presents our results after collection of All-SAT results from all valid attractor cycle orderings. In Fig. 2.4, 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.
2.5 Experimental Results

Fig. 2.4 Method A: Predictor occurrence for all valid attractor cycle orderings (first iteration: no predictor selected)

2.5.1 Method A

In method A, a predictor histogram is created as in Fig. 2.4. 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).

2.5.2 Method B

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 2.3 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 2.3, 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.

### 2.5.3 Method AB

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 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 [3] using methods A, B, and AB. Results are shown in Table 2.4 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) Gene histogram 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 \( S \).
Table 2.4 Melanoma network predictor set selection

<table>
<thead>
<tr>
<th></th>
<th>PIRIN</th>
<th>S100P</th>
<th>RET1</th>
<th>MART1</th>
<th>HADHB</th>
<th>STC2</th>
<th>WNT5A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>1357</td>
<td>2137</td>
<td>3146</td>
<td>4357</td>
<td>5124</td>
<td>6124</td>
<td>7124</td>
</tr>
<tr>
<td>x2</td>
<td>2.57</td>
<td>1.41</td>
<td>1.34</td>
<td>1.30</td>
<td>1.41</td>
<td>1.66</td>
<td>1.31</td>
</tr>
<tr>
<td>x3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>1357</td>
<td>2137</td>
<td>3146</td>
<td>4137</td>
<td>5134</td>
<td>6137</td>
<td>7126</td>
</tr>
<tr>
<td>x2</td>
<td>1.78</td>
<td>1.77</td>
<td>1.84</td>
<td>1.97</td>
<td>1.99</td>
<td>1.98</td>
<td>2.56</td>
</tr>
<tr>
<td>x3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>1357</td>
<td>2367</td>
<td>3146</td>
<td>4137</td>
<td>5137</td>
<td>6357</td>
<td>7124</td>
</tr>
<tr>
<td>x2</td>
<td>2.06</td>
<td>1.57</td>
<td>1.75</td>
<td>1.61</td>
<td>1.45</td>
<td>1.39</td>
<td>1.88</td>
</tr>
<tr>
<td>x3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The final predictor set is present in a small 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 $f_1 = \{x_3, x_5, x_7\}$ (PIRIN predicted by RET1, HADHB, WNT5A) as well as $f_3 = \{x_1, x_4, x_6\}$. 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 $x_2$ (S100P) contain 2 common genes $\{x_3, x_7\}$ (RET1, WNT5A), indicating these 2 genes are likely to be contained in the final predictor of $f_2$. Similarly $f_7$ has two common genes $x_1$ and $x_2$ 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.

2.6 Chapter Summary

Determining the predictor set for a gene regulatory network is important in many applications, particularly inference and control of the GRN which we discuss in subsequent chapters. In this research, 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.

The results of this research, however, only reveals the predictor set (topology) of the GRN. Our next step is to determine the gene regulating function (logic) of the genes in the GRN to fully define the BN. In the next chapter, we describe a logic synthesis method for determine gene functions for a GRN using a SAT-based logic synthesis approach.
References

Logic Synthesis for Genetic Diseases
Modeling Disease Behavior Using Boolean Networks
Lin, P.-C.K.; Khatri, S.P.
2014, XXI, 100 p. 28 illus., 8 illus. in color., Hardcover
ISBN: 978-1-4614-9428-7