A Boolean Logic Cell Model of Vesicular Trafficking

Ashish Baghudana

National Center for Biological Sciences, Bangalore

Guided by:

Dr. Mukund Thattai

Simon Center for the Study of Living Machines

Contents

Abstract

Eukaryotic cells contain multiple membrane-bound organelles or *compartments*. These compartments contain many proteins, which are frequently exchanged with other compartments through packets called *vesicles*. This process of exchange of molecules is called *vesicular trafficking*. Each compartment and vesicle is identified uniquely by its composition. Here, we represent each compartment and vesicle by a Boolean string and describe the dynamics of vesicular transport through functions on these Boolean strings.

Vesicular transport is a very specific process and is mediated through several sets of proteins – SNAREs and Coat Proteins. These not only dictate the specificity of transport, but also assist in the budding and fusion of vesicles. These proteins can either be activators or inhibitors of the fusion process. In this model, we abstract away the properties of each protein and represent them as Boolean variables, which are then used in molecular rules. These molecular rules describe which vesicles can fuse with and bud from a compartment.

We have studied properties of these molecular rules by studying their cognate Boolean formulae. We observed the satisfiability count of these formulae to changes in the number of activators, deletion and/or duplication of certain proteins and finally hybridization of sets of molecular rules.

We find that fusion molecular rules obey a strict Disjunctive Normal Form. Subsequently, the satisfiability count of these formulae depends on the total number of variables, the number of clauses and the number of terms per clause in the formula. The set of budding rules however, are more complicated and cannot be as easily represented as the fusion formulae.

1 Introduction to the Mechanisms of Vesicle Budding and Fusion

Newly synthesized proteins pass through a large number of membrane-bound organelles or compartments on their way to the extracellular space, plasma membrane, lysosomes, endosomes and so on. The main intermediate compartments in this transport chain are endoplasmic reticulum (ER) and the Golgi apparatus. The *Vesicular Transport Hypothesis* established that such secretory proteins are carried in vesicles[4]. These vesicles "bud" from compartments in a process that selectively picks only some proteins to be part of the vesicle. This selective incorporation is called "protein sorting". These vesicles are now targeted to other compartments that recognize the vesicle and unload its contents into themselves[1] [4] [8]. This process of budding and fusion continues till the cell reaches a dynamic equilibrium where no new proteins are being synthesized and the existing ones have reached their final destination. Broadly, the mechanism of vesicle transport can be classified into the following steps:

1. Protein Sorting
2. Vesicle Budding
3. Vesicle Tethering
4. Vesicle Fusion

1.1 Protein Sorting

Each vesicle contains a coat protein on the cytosolic surface. Three types of coat proteins are known as of now: **(a) Clathrin coat (b) COPII and (c) COPI**. These coat proteins are involved in both protein sorting and vesicle budding[6]. Coat subunit proteins polymerize at the vesicle-cytosol interface and ultimately help the vesicle to pinch off from the compartment. Each coat protein has two interfaces — adaptor side and cage side. The adaptor side of the coat protein helps choose molecules from amongst the compartment that form the cargo. Choosing occurs on the basis of protein sorting signals in the form of motifs or 3-D structures of the protein. Protein sorting signals are quite diverse and can consists of multiple motifs[5] [6].

1.2 Vesicle Budding

In addition to cargo selection, coat proteins are responsible for the change in shape of the membrane. The cage side helps stabilize the vesicle budding process. The coat protein destabilizes the membrane and causes it to deform. Some proteins in the coat protein complex also help stabilize the curvature required for vesicle budding. The vesicle budding process is energy-dependent. Therefore, each of these coat proteins are usually associated with some GTPase activity that allows vesicle formation.[4]

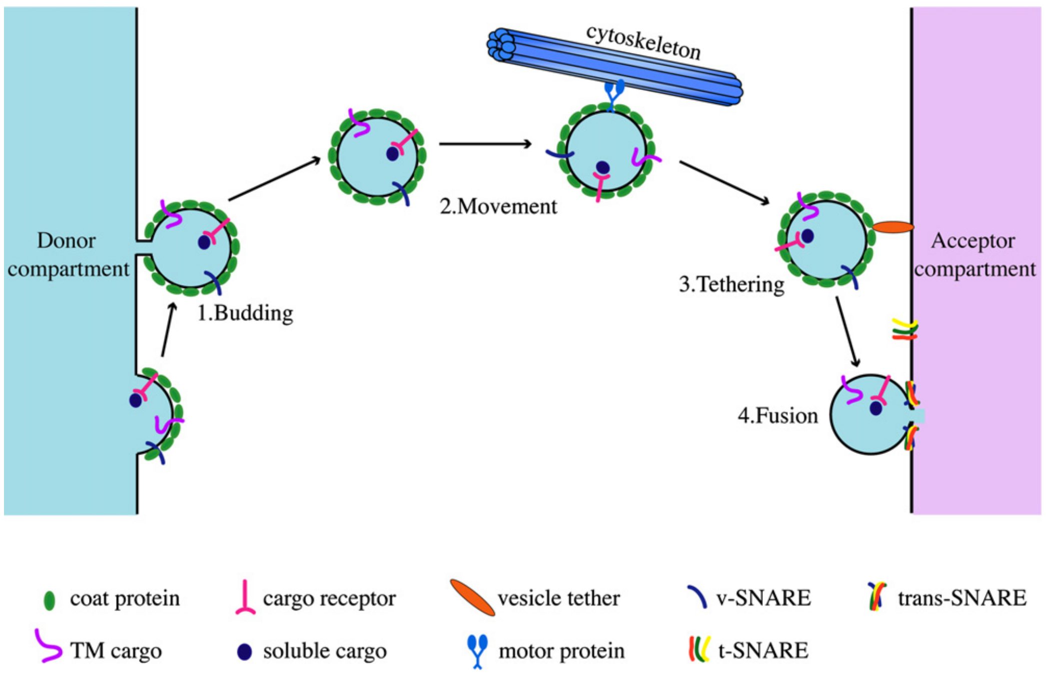

Figure 1: Vesicle-dependent transport consists of 4 steps: Cargo Selection and Vesicle Movement, Vesicle Budding, Vesicle Tethering and Vesicle Fusion. Sourced from Cai et. al., 2007[6]

1.3 Vesicle Tethering

Vesicle tethering refers to the initial interaction between the vesicle and its target compartment. Vesicle tethering is an important step in transport as it brings the vesicle and its cognate compartment in proximity. Even if a vesicle can fuse with a compartment, if tethers do not bring these two in proximity, fusion does not occur. Together with RABs, small GTPases of the Ras superfamily, tethers play an important role in determing the specifity of vesicle targeting.

1.4 Vesicle Fusion

The final step in vesicular transport is the fusion of a vesicle with its target compartment after they have been brought into close proximity. Fusion is regulated by SNARE proteins .SNARE proteins are particularly well studied and are related to three different neuronal proteins: synaptobrevin, syntaxin and SNAP-25. These are sub-classified into V-SNAREs and T-SNAREs. A V-SNARE on a transport vesicle pairs with its cognate T-SNARE on a compartment followed by fusion[7].

2 A Primer to Boolean Algebra

Boolean algebra is the field of mathematics which deals variables that can have only two possible values: *True* or *False*. Boolean expressions use such variables along with the following operators: *Conjuction* $\wedge$, *Disjuction* $\vee$, *Negation* $\neg$, *Implication* $\Rightarrow$ and *bi-Implication* $\Leftrightarrow$. Formally, these variables are called *Propositional Letters* and Boolean expressions are known as *Propositional Logic*[2].

2.1 Basics of Boolean Logic

Boolean expressions are generated from the following grammar.

$$t ::= x \mid 0 \mid 1 \mid \neg t \mid t \wedge t \mid t \vee t \mid t \Rightarrow t \mid t \Leftrightarrow t$$

where x ranges over a set of Boolean variables. As with algebra, Boolean operators also have an order of precedence. The priorities, with the highest first: $\neg$, $\wedge$, $\vee$, $\Leftrightarrow$, $\Rightarrow$.

Any Boolean expression can be represented using a truth table. Using the rules of precedence and truth tables given in Table 1, we can decide the value of any Boolean function.

	$\neg$
0	1
1	0

(a)

$\wedge$	0	1
0	0	0
1	0	0

(b)

$\vee$	0	1
0	0	1
1	1	1

(c)

$\Rightarrow$	0	1
0	1	1
1	0	1

(d)

$\Leftrightarrow$	0	1
0	1	0
1	0	1

(e)

Table 1: Truth tables for (a) Negation (b) Conjunction (c) Disjunction (d) Implication and (e) Bi-Implication

A few other definitions that we use during this report are *satisfiability* and *satisfiability count*. A Boolean formula is **satisfiable** if there exists at least one assignment of variables such that the formula evaluates to *True*. If a Boolean formula evalues to *True* for every possible assignment, it is called a **tautology**. The **satisfiability count** of a Boolean formula is the number of assignments of variables that evaluate to *True*.

2.2 Normal Forms of Boolean Expressions

A Boolean expression is in *Disjunctive Normal Form (DNF)* if it consists of a disjunction of conjunctions of variables and their negations, also known as a sum of products[3]. It is of the form

$$(t_1^1 \wedge t_2^1 \wedge \cdots \wedge t_{k_1}^1) \vee \cdots \vee (t_1^l \wedge t_2^l \wedge \cdots \wedge t_{k_l}^l)$$

Similarly, a *Conjunctive Normal Form (CNF)* consists of a conjunction of disjunctions of variables and their negations, also known as a product of sums. It is of the form

$$(t_1^1 \vee t_2^1 \vee \cdots \vee t_{k_1}^1) \wedge \cdots \wedge (t_1^l \vee t_2^l \vee \cdots \vee t_{k_l}^l)$$

Every Boolean expression can be represented a DNF or a CNF.

2.3 Binary Decision Diagrams

A Binary Decision Diagram (BDD) represents Boolean expressions in the *if-then-else* form, given by $x \to y_1, y_2$

$$x \to y_1, y_2 = (x \wedge y_1) \vee (\neg x \wedge y_2)$$

An expression $t \to t_1, t_2$ is true if t and t_1 are true or if t is false and t_2 is true. A Boolean expression is in the *If-then-else Normal Form (INF)* if it is built entirely from the if-then-else operator and the constants 0 and 1[2]. Thus, at each level, only single variables are evaluated and a tree is formed called a *binary decision tree.* At each level, a subexpression of the original Boolean formula is formed by substituting certain variables successively. A BDD is thus a complete binary tree which can be constructed as a rooted, directed acyclic graph with the following properties.

- one or two terminal nodes of out-degree zero labeled 0 or 1
- a set of variable nodes u of out-degree two. The two outgoing edges are given by two functions $low(u)$ and $high(u)$. A variable $var(u)$ is associated with each variable node which denotes the depth of the node.

A *Reduced Ordered Binary Decision Diagram* is a BDD which follows a distinct ordering of variables $x_1 < x_2 < \ldots < x_n$. It also has the following two properties.

- **(uniqueness)** no two distinct nodes u and v have the same variable name and low and high-successor i.e.

$$if\ var(u) = var(v), low(u) = low(v), high(u) = high(v), then\ u = v$$

- no variable node u has identical low and high-successor, i.e.,

$$low(u) \neq high(u)$$

To represent Boolean expressions, ROBDDs are constructed which are extremely amenable to manipulation and satisfiability count (SatCount) computation.

3 Modeling the Vesicular Trafficking System

The system of vesicular trafficking is mainly governed by the following factors.

1. Fusion Molecules
   (a) Fusion Activators
   (b) Fusion Inhibitors
2. Budding Molecules
   (a) Budding Activators
   (b) Budding Inhibitors
3. Coat Proteins

In our model, we placed a few restriction on the molecular rules that describe vesicular transport. A molecule can either be an activator or an inhibitor but not both. We have also assumed that a molecule can either be involved in the process of fusion or of budding, and not both. Given a n molecule system, where k molecules are fusion molecules and the remaining $n - k$ are budding molecules there are 2^n possible unique compartments and similarly the same number of vesicles. The set of molecular rules are represented as a fusion and budding matrix, labeled F-Matrix and G-Matrix respectively. The order of both matrices is $2^n \times 2^n$. As is immediately evident, increasing the number of molecules by even one leads to quadrupling of each matrix. At the level of 10 molecules, we are already dealing with approximately one million entries in each matrix.

3.1 Fusion Matrix

The fusion matrix describes which fusions are possible. We seek to represent each matrix as a Boolean expression where every variable indicates the presence or absence of the corresponding molecule. Since the same molecule can be present or absent in either the compartment or the vesicle, two different variables represent the presence or absence of the molecule in the (a) compartment, and (b) the vesicle separately.

Each matrix can be represented as a Boolean formula, and thereby converted to a DNF. With the restrictions that we placed on our molecular rules, we see that DNFs thus formed are monotonic in nature. A monotonic function is one that is either increasing or decreasing in each of its variables independently. This property arises because a variable in the Boolean formula appears either in its positive form or negated form only, and not both. Biologically, an activator molecule appears in the positive form, whereas an inhibitor appears negated. A simple example of a monotonic Boolean expression in its Disjunctive Normal Form is:

$$(a \wedge \neg d \wedge \neg e) \vee (b \wedge c \wedge k) \vee (b \wedge \neg n \wedge \neg o)$$

In this example, a, b, c and k are activators of fusion, whereas d, e, n and o are inhibitors.

3.1.1 Measuring SatCount and its Dependence on Activators, Clauses and Terms

We randomly generated many Boolean expressions in their DNFs by varying the number of molecules, number of activators, number of clauses and the number of terms per clause. We were mainly interested in the distribution of SatCount as a function of the number of activators.

We found that the **SatCount of Boolean expressions is independent of the number of activators**. The number of activators only shifts around the ones in the matrix and does not actually result in a change in the SatCount.

We also examined the effect of number of clauses and the number of terms per clause on the SatCount of a formula. We found that SatCount increases as the number of clauses increases and decreases as the number of terms decreases.

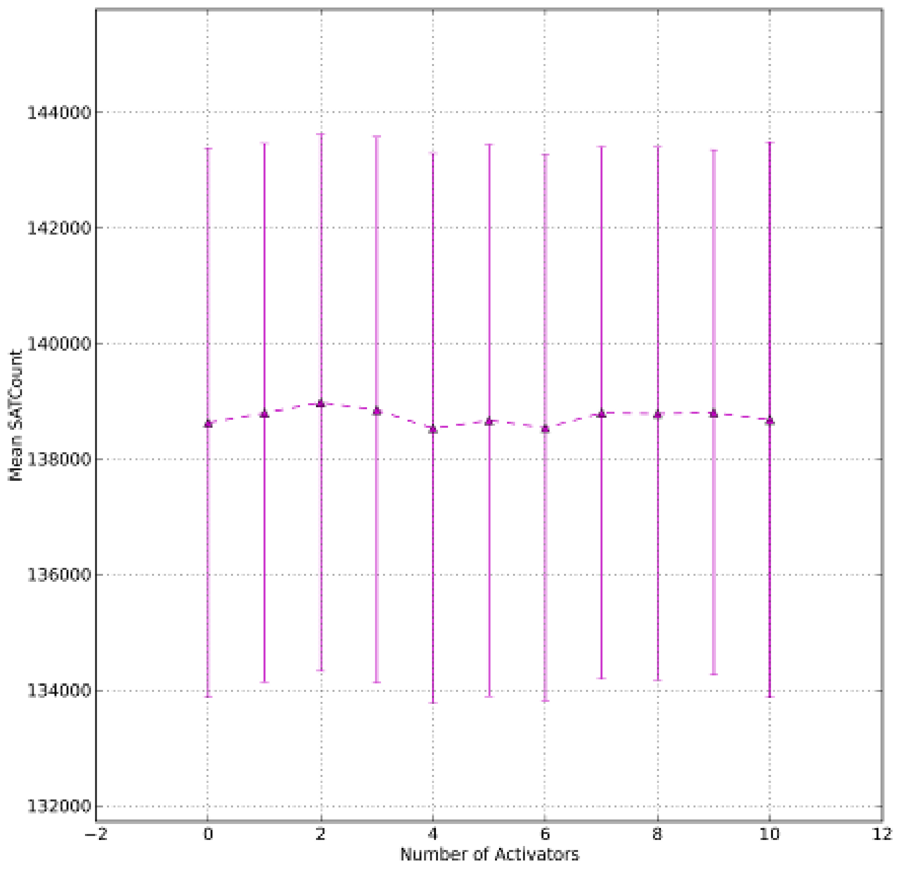

Figure 2: Mean SatCount for a 1000 randomly generated fusion Boolean expressions with 10 molecules, 5 clauses and 5 terms per clause. The SatCount is relatively steady despite the change in number of activators.

To summarize,

$$SatCount \not\propto Activators \tag{1}$$

$$SatCount \propto \frac{Clauses}{Terms} \tag{2}$$

3.1.2 Clustering of Ones and Zeros

We also compared our randomly generated Boolean formulae with randomly generated matrices of the same density to understand their similarities and

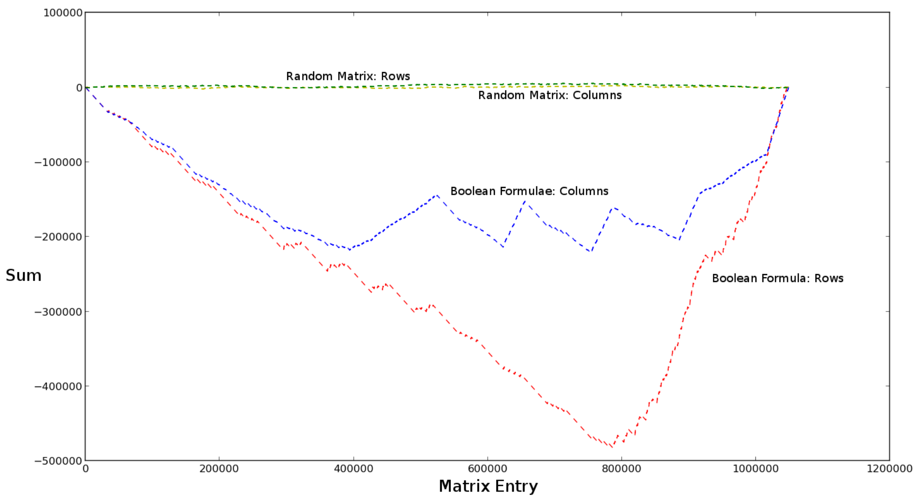

Figure 3: A Matrix Walk was done (along both rows and columns) for a randomly generated Boolean formula and it was compared with a randomly generated matrix of the same density. The nature of the Boolean formula is such that there is significant clustering of ones and zeros. Therefore, the plot for a Boolean formula has clear peaks. The randomly generated matrix however remains more or less steady and near 0.

differences. We first looked at a matrix walk through both the formula as well as the random matrix.

We define walking through a matrix as follows. Starting from the first entry of the matrix, we move either along a row or a column. Let the number of ones in the matrix be $ones$ and number of zeros be $zeros$. We initialize a variable sum. For every 0 we encounter, we arbitrarily add -1 to sum and for every 1 we encounter, we add a factor equalling $\frac{zeros}{ones}$ to sum. We obtain a plot as shown in Figure 3.

3.1.3 Number of Clauses and Terms in a Randomly Generated Matrix of a Given Density

As we have seen earlier, the SatCount or density of a matrix depended on the number of clauses and terms in a Boolean formula. We attempted to find out the average number of clauses and terms if we were to convert a random matrix to a Boolean formula in its DNF (Figure 4). We saw a blow up of the number of terms and clauses in each formula when compared to

Boolean formulae of the same densities. This is expected because of the declustering of ones and zeros in a random matrix. Because these ones and zeros are distributed evenly, no distinct pattern is found in their corresponding Boolean formulae.

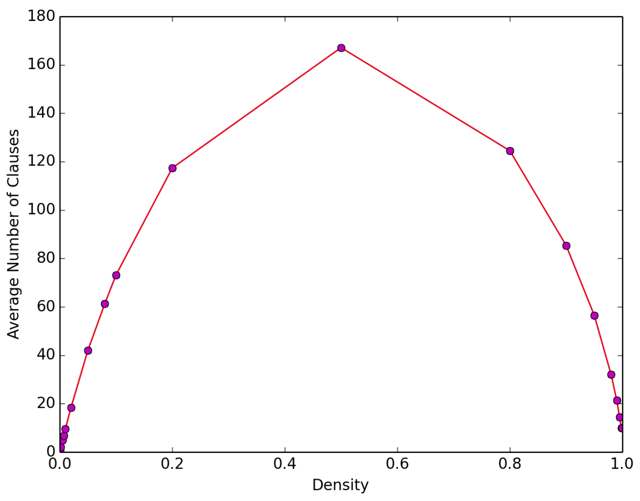

Figure 4: We chose a particular density for a matrix and generated 100 such random matrices. We converted each of them to a Boolean formula and simplified them using the QM method. The plot shows the number of clauses we obtained for different densities.

3.2 Budding Matrix

Apart from budding molecules, the budding formula depends on coat proteins as well. Each coat protein has two properties: (a) a recruitment DNF; and (b) a Nc String . The recruitment DNF tells the coat protein which compartments it can attach to, and the Nc String dictates which molecules it can pick from the compartment. All possible subsets of the Nc string are packed into a vesicles from each compartment. If multiple coats can attach themselves to a compartment, then each compartment will give away many different types of vesicles.

Each G-Matrix tells us which vesicles can be given away from each compartment. Together with the F-Matrix, we can trace the complete path of a cell state till it reaches a steady state.

Consider an n molecule system. Let the vesicle variables be $V_1, V_2, \ldots, V_n$

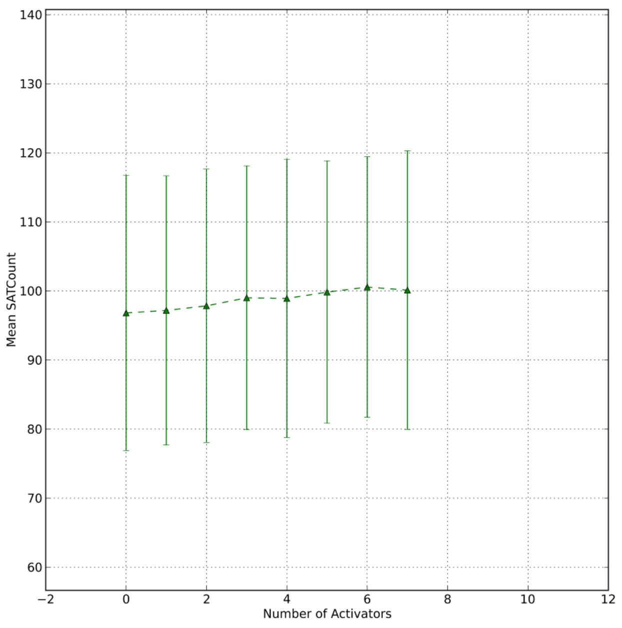

Figure 5: Mean SatCount for a 1000 randomly generated budding Boolean expressions with 7 molecules, 3 clauses, 3 terms per clause and 3 coat proteins. The SatCount is relatively steady (as with fusion formulae) despite the change in number of activators.

and the compartment variables to be $C_1, C_2, \ldots, C_n$. Let there be k coat proteins. Then, the Nc string for each coat is a binary string represented by $(Nc_1^1, Nc_2^1 \ldots Nc_n^1), \ldots, (Nc_1^k, Nc_2^k \ldots Nc_n^k)$ respectively. Each G-formula is of the form

$$\begin{gathered}
((V_1 \Leftrightarrow (C_1 \wedge Nc_1^1)) \wedge (V_2 \Leftrightarrow (C_2 \wedge Nc_2^1)) \wedge \cdots \wedge (V_n \Leftrightarrow (C_n \wedge Nc_n^1))) \\
\vee ((V_1 \Leftrightarrow (C_1 \wedge Nc_1^2)) \wedge (V_2 \Leftrightarrow (C_2 \wedge Nc_2^2)) \wedge \cdots \wedge (V_n \Leftrightarrow (C_n \wedge Nc_n^2))) \\
\vdots \\
\vee ((V_1 \Leftrightarrow (C_1 \wedge Nc_1^k)) \wedge (V_2 \Leftrightarrow (C_2 \wedge Nc_2^2)) \wedge \cdots \wedge (V_n \Leftrightarrow (C_n \wedge Nc_n^k)))
\end{gathered}$$

The recruitment DNFs are very similar in structure to the F-formulae. For instance,

$$(C_1 \wedge C_3 \wedge \neg C7) \vee (C_2 \wedge \neg C9 \wedge \neg C8)$$

G-Formulae consequently have a higher degree of complexity and are not simple DNFs as was the case with fusion formulae. However, we observe that SatCount even for the budding formulae does not depend on the number of activators. Thus,

$$SatCount \not\propto Activators \tag{3}$$

A sample plot of SatCount as a function of the number of activators is shown in Figure 5. The G-matrix however is much more sparse than the F-matrix at the same number of molecules.

4 Hybridization of Fusion Formulae

Very often, two different cells having similar molecular rules fuse together. This leads to a rather interesting phenomenon and allows us to understand the evolution of these rules. We represent the molecules in the two sets of cells as separate variables or entities. If the two molecules are identical or can be substituted for one another, there will be no perturbation of the molecular rules. However, if these two molecules are, in fact, independent, it will result in some perturbation of molecular rules.

We study these perturbations again through the SatCount of the hybridized

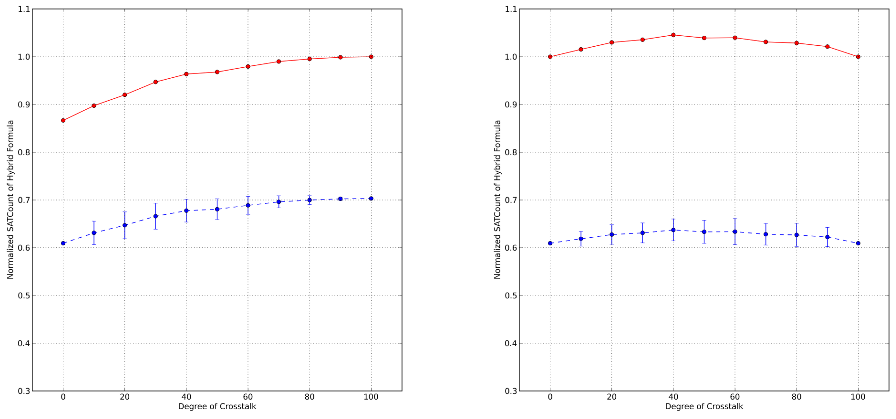

Figure 6: Normalized SatCount as a function of the Degree of Crosstalk. The blue line indicates the SatCount whereas the red line shows $\frac{SatCount}{SatCount\ at\ Crosstalk\ 100\%}$. In some cases, we observed that the SatCount was indeed maximum at a degree of crosstalk of 100%, whereas in other cases we did not find this to be the case.

formulae. Consider two formulae:

$$(a_1 \wedge b_1 \wedge c_1)$$

and

$$(a_2 \wedge b_2 \wedge c_2)$$

If these formulae are hybridized, and a_1 and a_2 are identical, we would get

$$((a_1 \vee a_2) \wedge b_1 \wedge c_1)$$

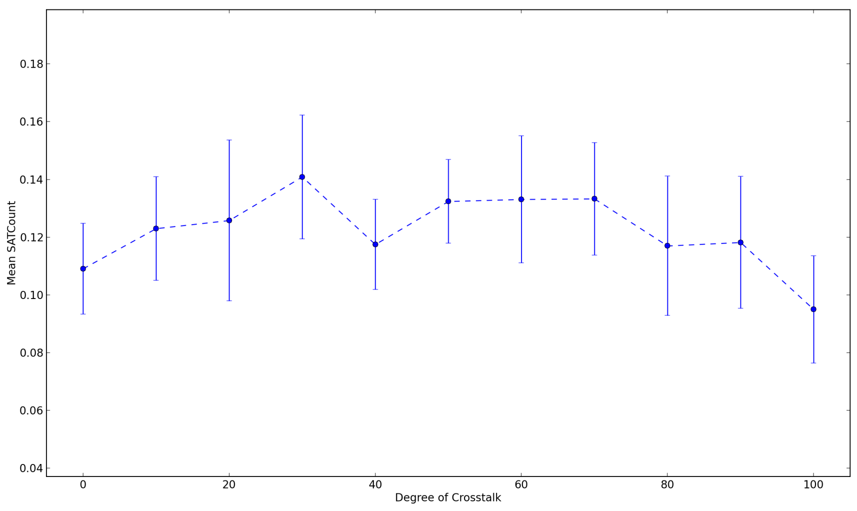

Figure 7: Normalized SatCount of a Budding formula as a function of the Degree of Crosstalk. These plots are much more haphazard than those of the Fusion formulae.

Figures 6 and 7 show plots of Normalized SatCount as a function of the degree of crosstalk for fusion formulae and budding formulae respectively. We find no clear pattern in either of these to predict a trend.

5 Ongoing Work

Having understood how Boolean formulae and their properties are affected by changes in parameters such as activators, clauses, terms, and coat proteins, we have turned our attention to Cell States. A cell state is described by the combination of compartments it has along with a specific F- and G-Matrix. With a n molecule system, there are 2^{2^n} possible cell states.

A layer in a Cell State space is defined by the number of unique molecules and the number of comparments. A n molecule, m compartment layer consists of all combinations of cell states with n unique molecules in m different compartments.

In any layer, we find three types of nodes - self nodes, exit nodes and transit nodes. Self nodes link back to themselves, exit nodes lead out of that layer (that is, the number of molecules or compartments decreases) and transit nodes lead to other nodes in the same layer. Using randomly generated matrices, we find that the percentage of self nodes and exit nodes is very huge. While the percentages vary from one set of matrices to another, we find that there are very few transit nodes in a layer. This is a slightly disturbing fact because it means that either most nodes in that layer lead out of it, or most cell states are trivial; they do not interact with other compartments and do not undergo any updates. We also found that the average path length to reach a self node was close to 1. Thus, we are now looking at generating more realistic matrices. One of the parameters we are investigating is the correlation between two compartments in a matrix.

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell, 4th edition*. Garland Science, New York, 2002.

[2] Henrik Reif Andersen. An introduction to binary decision diagrams. 1997.

[3] Mordechai Ben-Ari. *Mathematical Logic for Computer Science, 3rd edition*. Springer, London, 2012.

[4] Juan S. Bonifacino and Benjamin S. Glick. The mechanisms of vesicle budding and fusion. *Cell*, 116:153–166, 2004.

[5] Frances M. Brodsky, Chih-Ying Chen, Christine Knuehl, Mhairi C. Towler, and Diane E. Wakeham. Biological basket weaving: Formation and function of clathrin-coated vesicles. *Annual Review of Cell and Developmental Biology*, 17:517–568, 2001.

[6] Huaqing Cai, Karin Reinisch, and Susan Ferro-Novick. Coats, tethers, rabs, and snares work together to mediate the intracellular destination of a transport vesicle. *Developmental Cell*, 12:671–682, 2007.

[7] Yu A. Chen and Richard H Scheller. Snare-mediated membrane fusion. *Nature Reviews Molecular Cell Biology*, 2, 2001.

[8] Harvey Lodish, Arnold Berk, S. Lawrence Zipursky, Paul Matsudaira, David Baltimore, and James Darnell. *Molecular Mechanisms of Vesicular Traffic*. W. H. Freeman, New York, 2000.