Exploring Extreme Signaling Failures in Intracellular Molecular Networks

Mustafa Ozen[1], Effat S. Emamian[2], and Ali Abdi[1,3 *]

Mustafa Ozen, PhD
[1]Department of Biochemistry
Vanderbilt University, 2301 Vanderbilt Place, Nashville, TN 37240, USA
Email: mustafa.ozen@vanderbilt.edu

Effat S. Emamian, MD
[2] Advanced Technologies for Novel Therapeutics (ATNT)
116 Millburn Ave, #110, Millburn, NJ 07041, USA
Email: emame@atnt-usa.com

Ali Abdi, PhD
[1] Center for Wireless Information Processing
Department of Electrical and Computer Engineering
[3] Department of Biological Sciences
New Jersey Institute of Technology, 323 King Blvd, Newark, NJ 07102, USA
Email: ali.abdi@njit.edu
* Corresponding Author: Ali Abdi (ali.abdi@njit.edu)

DOI: 10.1016/j.compbiomed.2022.105692

Abstract: Developing novel methods for the analysis of intracellular signaling networks is essential for understanding interconnected biological processes that underlie complex human disorders. A fundamental goal of this research is to quantify the vulnerability of a signaling network to the dysfunction of one or multiple molecules, when the dysfunction is defined as an incorrect response to the input signals. In this study, we propose an efficient algorithm to identify the extreme signaling failures that can induce the most detrimental impact on the physiological function of a molecular network. The algorithm finds the molecules, or groups of molecules, with the maximum vulnerability, i.e., the highest probability of causing the network failure, when they are dysfunctional. We propose another algorithm that efficiently accounts for signaling feedbacks. The algorithms are tested on experimentally verified ERBB and T-cell signaling networks. Surprisingly, results reveal that as the number of concurrently dysfunctional molecules increases, the maximum vulnerability values quickly reach to a plateau following an initial increase. This suggests the specificity of vulnerable molecule(s) involved, as a specific number of faulty molecules cause the most detrimental damage to the function of the network. Increasing the number of simultaneously faulty molecules does not further deteriorate the network function. Such a group of specific molecules whose dysfunction causes the extreme signaling failures can better elucidate the molecular mechanisms underlying the pathogenesis of complex trait disorders, and can offer new insights for the development of novel therapeutics.

INTRODUCTION

Cellular functions are largely regulated by signaling events within the complex intracellular molecular networks [1-3]. Signals are typically transmitted from the cell membrane to the nucleus via intracellular signaling networks, to regulate some target molecules and alter different cellular functions. Intracellular signaling networks have been studied to address a variety of different questions [1-8].

An important application of the studies on intracellular signaling networks is in target discovery and drug development. In the area of targeted therapies in pharmaceutical industry, a wide variety of highly effective therapeutics have been successfully developed that can target the function of a selective set of molecules within the complex intracellular signaling networks. Fault diagnosis is a platform technology for finding such selective targets by using computational and systems biology techniques that have been developed and optimized over the past few years [9-12]. The main purpose of such technology developments is to understand how vulnerable the entire molecular network is to the dysfunction of each molecule, or a specific group of molecules. For the fault diagnosis of molecular networks, prior studies have focused on intracellular signaling networks with three main components of input molecules, intermediate signaling molecules, and output molecules. Input molecules of the network are typically ligands that bind to their receptors at the cell membrane. This ligand binding initiates a cascade of intracellular events that involve activation or inhibition of secondary messengers, G proteins, kinases, phosphatases, and other intracellular signaling molecules. Eventually through a cascade of signaling events initiated at the cell membrane, output molecules such as transcription factors are regulated, which ultimately alter cellular functions by changing the gene expression pattern [9-12].

In the realm of fault diagnosis technology development, the dysfunctional state of a molecule can be defined as the failure to respond correctly to the input signals, which may further propagate into incorrect responses at the output of the network. For this analysis, we have defined the vulnerability level of a molecule as the probability of having incorrect network responses when that specific molecule is dysfunctional. Vulnerability analysis can be performed for the dysfunction of a single as well as a group of molecules. The importance of the latter can be attributed to the widely known observations that the most common human disorders, such as cancer or schizophrenia, are reported to be associated with the dysfunction of multiple molecules rather than a specific single molecule [13,14]. This contrasts with some rare genetic disorder when a single molecule or a genetic mutation is sufficient to cause the pathology [15]. By computing the vulnerability level of a molecule or a group of molecules, one can identify and rank the key molecular components for gaining a desired response, and then compute how much they contribute to the failure of the network. From the drug development stand points, the highly vulnerable molecules can be used as the molecular targets of novel therapeutics.

To analyze a molecular network, first a biologically relevant model needs to be adopted. Here we consider the class of discrete models such as Boolean models [1-6,8,16] in which each molecule has a binary activity state of 0 or 1. In such models, the biological equivalent of the binary levels can be the activation of a molecule, i.e., the binary state of 1, or the inhibition of a molecule, i.e., the binary state of 0, specified by the states of its upstream molecules. They are particularly useful as they do not need detailed kinetic information and still provide certain biologically relevant insights and predictions. Compared to continuous differential equations models [7], discrete models do not require the knowledge of many mechanistic details and numerous kinetic parameters, and are more appropriate for our study in this paper.

The main goal of this paper is to develop a systematic method to explore the extreme failures of intracellular signaling networks. We define the extreme signaling failure as a pathological phenomenon that results in the highest probability of network failure, where the network failure is defined as the level of departure of the network response from its normal or expected response. The said pathological phenomenon is characterized to be emerged from the presence of one or more dysfunctional molecules in the network. It is conceivable that different individual dysfunctional molecules can result in different levels of probabilities of the network failure. It is not clear, however, what happens if two or more molecules are concurrently dysfunctional, and if the network failure probability increases with the number of simultaneously dysfunctional molecules or not. It is also of interest to have an efficient algorithm to determine the maximum possible network failure probability, over the large number of all possible groups of dysfunctional molecules. The computational complexity of an exhaustive search approach for a network with *K* molecules is extremely high, in the order of $K^{K/2}$, which is unmanageable as *K* increases. Here we introduce a computationally efficient algorithm with a much less running time in the order of K^3 , for identifying the extreme signaling failures, considering multiple dysfunctional molecules. This study is particularly important in the context of complex disorders with unknown molecular sources, where more than one molecule is observed to be involved in the pathology [13,14].

In this study we first present our extreme signaling failure analysis results on the ERBB signaling network of [17] and the T cell signaling network of [2]. Then, we analyze the computational complexity of the proposed algorithm and compare it with the exhaustive search. Afterwards, we provide the details of the vulnerability analysis equations and the proposed extreme signaling failure analysis algorithm in Methods, Sections A and B, respectively. Moreover, we propose another algorithm in Section S1 of

Supplementary Information, that determines the number of time points (clock cycles[1]) needed for network analysis and simulation while computing the vulnerability levels, so that we prevent running network simulations longer than what is needed. We present the results of this algorithm, when applied to the ERBB and the T cell signaling networks, in Sections S2 and S3, respectively, and finally conclude the paper with some concluding remarks.

RESULTS OF THE EXTREME SIGNALING FAILURE ANALYSIS

The extreme signaling failure algorithm was applied to two networks: a small signaling network that regulates the transmembrane tyrosine kinase ERBB [17], a therapeutic target in breast cancer, and a larger T cell signaling network [2] involved in a variety of immune system response. Results reveal the impact of the number of dysfunctional molecules in causing the extreme signaling failure.

A. Extreme Signaling Failure Study of ERBB Network

The ERBB network (Figure 1) has one input and one output. The biological equivalent for the input molecule is the Epidermal Growth Factor (EGF), and for the output molecule is the Retinoblastoma Protein (pRB). This network has been studied in the context of breast cancer and understanding the mechanism of action of few drug molecules [17]. Figure 1 shows the biological components of this network, including the ligand, its receptors, and the specific signaling molecules with their activation or inhibition states being changed after the binding of EGF to its receptors. Through the cascade of signaling events involving activation (green edges in Figure 1) or inhibition (red edges in Figure 1) of several intracellular molecules, eventually the output molecule pRB is activated. Equations that specify how the activity of each molecule is regulated by its inputs are listed in Table S1. To model a dysfunctional molecule, we assume its activity state is either stuck-at-0, SA0, or stuck-at-1, SA1, each with a probability of 1/2 [9].

Let *N* be the number of molecules that are simultaneously faulty, i.e., dysfunctional, in the network. According to the developed vulnerability analysis equations (Methods, Section A) and using the proposed extreme signaling failure algorithm (Methods, Section B), the network maximum vulnerability is computed for each *N* (Figure 2). Here *N* varies from 1 to 18, since the number of intermediate molecules in the network (Figure 1) is 18. For any *N*, the network maximum vulnerability is the highest probability of network failure, where the network failure is defined as the level of departure of the

[1] In the context of network simulation over time, the *clock* represents a time reference, i.e., a time axis, divided into intervals called *clock cycles*. Therefore, each clock cycle corresponds to a specific time point at which the network response is simulated.

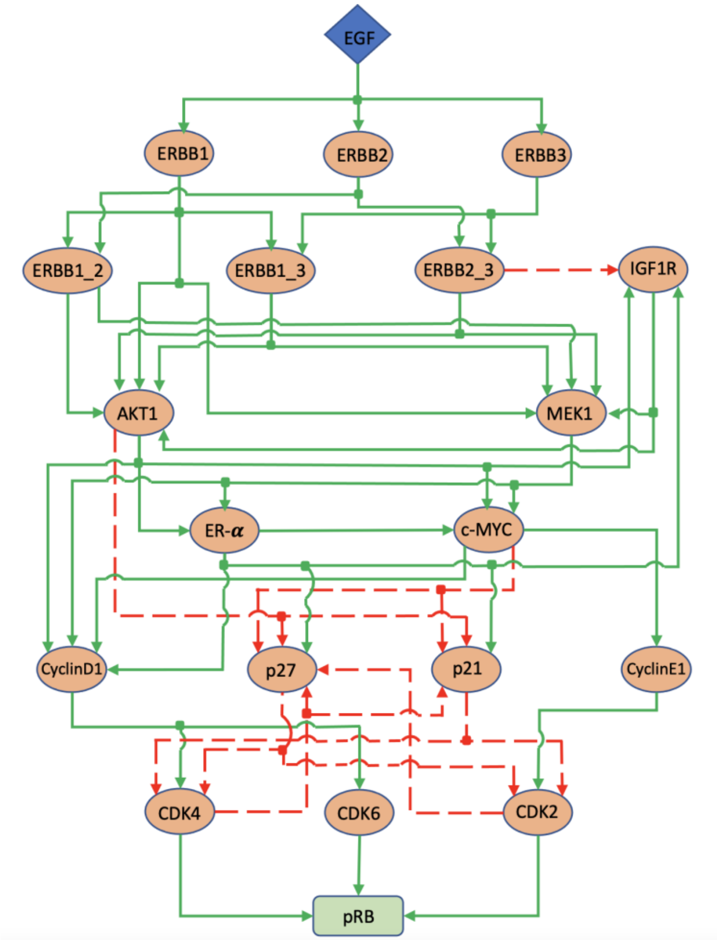


Figure 1. The experimentally verified ERBB signaling network reproduced from the study of Sahin *et al* [17]. The green solid edges represent activatory interactions and the red dashed edges represent inhibitory interactions. The green and red branching-out square markers pinpoint the crossing edges. The input and output nodes represent EGF and pRB, respectively.

network response from its normal values. In other words, the network maximum vulnerability for a given N is a parameter that quantifies the extreme possible signaling failure when there are N faulty molecules in the network.

An unexpected observation is that as the number of faulty molecules N increases, the maximum vulnerability values do not increase accordingly (Figure 2). While we see a maximum vulnerability increase going from single faults to double faults, i.e., $N = 1$ and 2 respectively, the maximum vulnerability quickly reaches a plateau and no longer increases further afterwards. Another interesting observation is that the smallest N for which we see the highest maximum vulnerability in this network is $N = 2$, i.e., double faults. This means there are some pairs of faulty molecules that cause the most

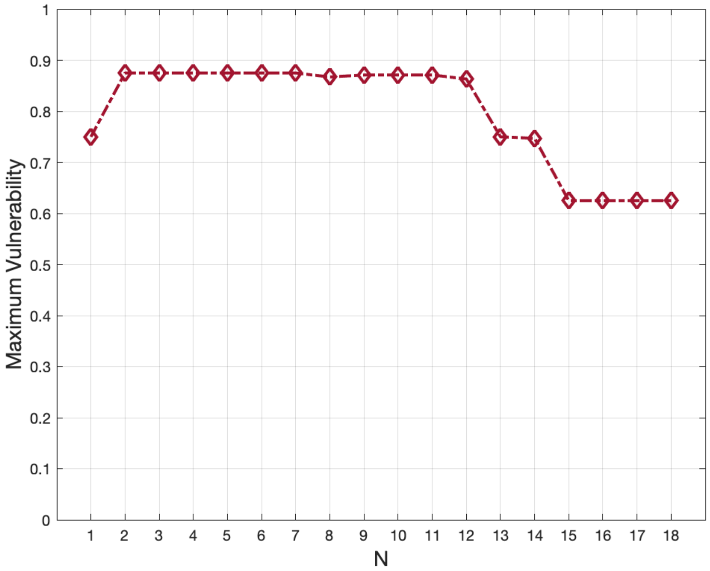


Figure 2. The ERBB signaling network maximum vulnerability levels, when there are *N* dysfunctional molecules in the network, computed using the proposed algorithm to study worst possible signaling failures.

detrimental damages to the signaling network, and increasing the number of faulty molecules can no further deteriorate the function of outputs. In other words, this analysis suggests that it may be enough to have only a few concurrently faulty molecules to disturb the whole network's functioning, which may eventually result in the development of the pathology.

In addition, we observe that when $N > 12$, the maximum vulnerability level starts decreasing. The decrease occurs because when the number of faulty molecules increases, erroneous signals of some "more harmful" faulty molecules exhibiting high vulnerability levels are blocked/masked by the additional "less harmful" faulty molecules (that have lower vulnerability levels). In other words, when the number of faulty molecules *N* is large, some pathways from the very harmful molecules to the network output may be cut off and disrupted because of the additional dysfunctional molecules in the pathways, that protect the network output from being affected by those very harmful molecules. That is why the maximum vulnerability level decreases after a certain *N* value, $N > 12$ in Figure 2. To exemplify, let us compare $N = 12$ and $N = 13$ cases, in which the maximum vulnerability levels are 0.86 and 0.75, respectively. When $N = 12$, the group of 12 molecules exhibiting the maximum vulnerability level of 0.86 contains ERBB1, ERBB2, ERBB3, ERBB1_2, ERBB1_3, ERBB2_3, IGF1R, AKT1, MEK1, ER-α, CyclinD1, and CyclinE1. The remaining fault-free molecules are c-MYC, p21, p27, CDK2, CDK4, and CDK6. All these six molecules are in the downstream of the above 12 faulty molecules, and they all have low single fault vulnerability levels, i.e., 0.25 for c-MYC, p21, p27, CDK4, CDK6, and 0 for CDK2. When $N = 13,$ the group of 13 molecules with maximum vulnerability will include one of these six molecules that have low individual vulnerabilities, and are also in the downstream of the group of 12 maximally vulnerable molecules. Therefore, when $N = 13,$ some effects of the more harmful signals are blocked by the downstream additional faulty molecule that has a low individual vulnerability. This reduces the maximum vulnerability when N changes from 12 to 13. As N increases beyond 13, the groups of faulty molecules will include even more of those six slightly vulnerable molecules, which can be the reason for the further decrease of the maximum vulnerability.

B. Extreme Failure Study of T cell Signaling Network

The T cell network is a much more complex real biological network that has three inputs and fourteen outputs. Figure 3 shows the biological components of this network, including the ligands, several receptors, and the specific signaling molecules with their activation or inhibition states being altered after the binding of the input molecules to their receptors. Through the cascades of several signaling events that involve activation (green edges in Figure 3) or inhibition (red edges in Figure 3) of numerous intracellular molecules, the output molecules are ultimately activated or inhibited. The biological equivalents of input molecules are cluster of differentiation 28 (cd28), cluster of differentiation 4 (cd4) and ligand-bound T-cell receptor (tcrlig) [2]. For the output molecules, biological equivalents are Src homology region 2 domain-containing phosphatase-2 (shp2), B-cell lymphoma-extra large (bclxl), p70s, activator protein 1 (ap1), serum response element (sre), branched-chain amino acid transaminase (bcat), cytochrome c1 (cyc1), p21c, p27k, forkhead transcription factor Foxo1 (fkhr), p38, cAMP - cyclic adenosine monophosphate - response elements (cre), nuclear factor of activated T cells (nfat), and nuclear factor kappa-light-chain-enhancer of activated B cells (nfkb) [2]. Equations that specify how the activity of each molecule is regulated by its inputs are listed in Table S2.

Using the developed vulnerability analysis equations (Methods, Section A) and the proposed extreme signaling failure analysis algorithm (Methods, Section B), the network maximum vulnerability is computed for each N (Figure 4), where N is the number of molecules that are simultaneously faulty. Similar to the ERBB network, for all the outputs, as the number of faulty molecules N increases, again maximum vulnerability values do not increase accordingly (Figure 4). Moreover, for the network outputs ap1, bcat and p70s, while we see a maximum vulnerability increase going from single faults to double faults, $N = 1$ and 2, respectively, the maximum vulnerability does not increase more afterwards. For these outputs, the smallest N for which we see the highest maximum vulnerability in this network is

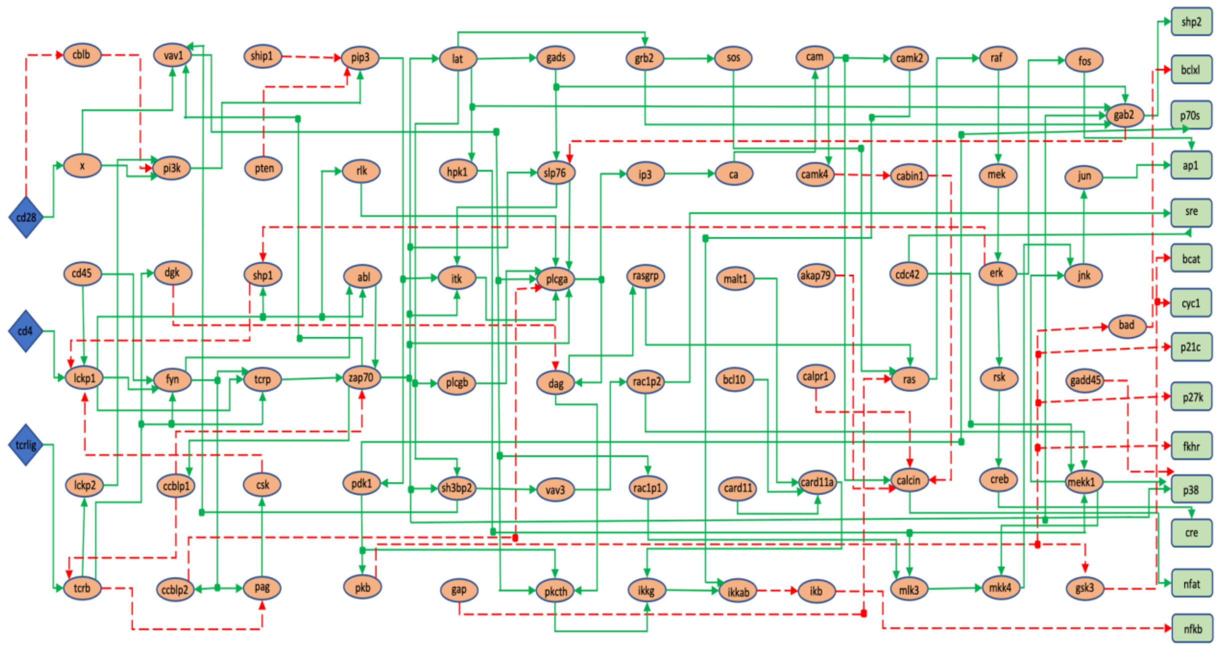


Figure 3. The experimentally verified T cell signaling network reproduced from the study of Saez-Rodriguez et al [2]. The green solid edges represent activatory interactions and the red dashed edges represent inhibitory interactions. The green and red branching-out square markers pinpoint the crossing edges.

$N = 2$, i.e., double faults. This means that there are some pairs of faulty molecules that cause the most detrimental damage for these outputs, and increasing the number of faulty molecules does not further deteriorate the function of these outputs. However, for some other network outputs, including cre, nfat, p38, shp2 and sre, the highest maximum vulnerability occurs when $N = 1$, which implies that there are some single faulty molecules that can individually create the extreme signaling failures at these specific outputs (Figure 4).

COMPUTATIONAL COMPLEXITY OF THE EXTREME SIGNALING FAILURE ALGORITHM

In this section, we determine the computational complexity of the proposed algorithm and compare it with the running time of exhaustive search. The extreme signaling failure analysis can be performed via an exhaustive search. This means that if one needs to find the maximum network vulnerability when there are N faulty molecules in the network, all possible groups of N faulty molecules must be considered one by one, and the vulnerability value for each group needs to be computed. For example, consider a network

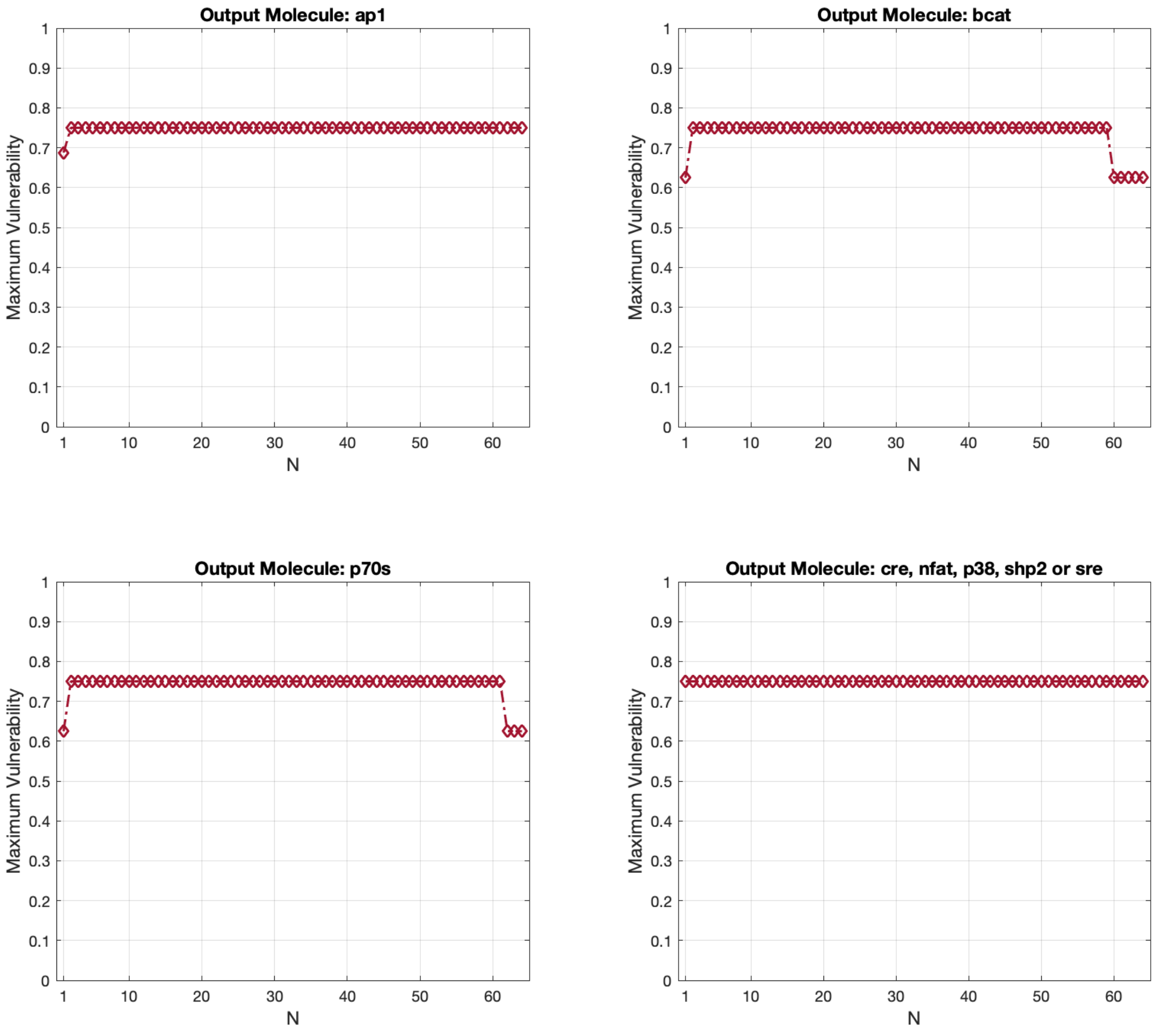


Figure 4. The T cell signaling network maximum vulnerability levels for the network outputs ap1, bcat, cre, nfat, p38, p70s, shp2 and sre, when there are *N* dysfunctional molecules in the network.

with $K = 50$ molecules. When $N = 2,$ the total number of *pairs* of faulty molecules that the exhaustive search has to examine can be shown to be 1,225 (see Equation (1)). For $N = 5,$ however, the total number of groups of *five* faulty molecules increases to 2,118,760. This computational complexity becomes highly prohibitive as the network size *K* increases. In what follows, we show that the proposed extreme signaling failure algorithm is much less complex than the exhaustive search.

To determine and compare the computational complexities, let *K* be the number of molecules in a network, and *N* be the number of molecules that are simultaneously faulty in the network. The total number of groups of *N* faulty molecules out of *K* molecules, $1 \le N \le K,$ is given in Equation (1), where $C(K,N)$ represents the number of possible combinations.

$$C(K,N) = \frac{K!}{\left(K-N\right)!N!} = \frac{K \times \left(K-1\right) \times \ldots \times \left(K-(N-1)\right)}{N!} = O\left(K^{N}\right) \tag{1}$$

Here the *O*-notation stands for the asymptotic upper bound [18]. The term $O\left(K^{N}\right)$ represents the computational complexity of $C(K,N)$ as a function of *K* and *N*.

The computational complexity of the exhaustive search $\alpha(K)$ is the number of all possible groups of *N* faulty molecules for which vulnerabilities have to be computed, $N = 1,...,K,$ i.e., $\alpha(K) = C(K,1) + ... + C(K,K)$. To simplify the notation and without loss of generality, assume *K* is even. We note that $C(K,N)$ has a maximum at $N = K/2$ [18], it is symmetric, i.e., $C(K,N) = C(K,K-N)$, $N = 1,...,(K/2)-1$, and $C(K,K) = 1.$ Therefore, the computational complexity of the exhaustive search simplifies to

$$\begin{aligned}\alpha(K) &= C(K,1) + ... + C(K,(K/2)-1) + C(K,K/2) + C(K,(K/2)+1) + ... + C(K,K), \\ &= 2\sum\nolimits_{N=1}^{(K/2)-1} C(K,N) + C(K,K/2) + 1.\end{aligned} \tag{2}$$

With $C(K,K/2)$ being the dominant term in Equation (2) when *K* is large, and also using Equation (1), the exhaustive search computational complexity can be finally written as

$$\alpha(K) = O\left(K^{K/2}\right) \tag{3}$$

To determine the computational complexity of the proposed extreme signaling failure algorithm (Methods, Section B), we note that initially all groups of one, two and three faulty molecules are considered, $N = 1,2,3.$ For the rest, $N = 4,...,K,$ only $K-(N-1)$ vulnerabilities are computed. This is inspired by the observation [11] that typically a molecule with a high vulnerability appears in larger groups of molecules with high vulnerabilities, and based on our experiments, $N \ge 4$ is large enough and provides good accuracy, as discussed in the paragraph immediately after Equation (5). Therefore, the computational complexity $\beta(K)$ of the algorithm can be written as

$$\beta(K) = \sum\nolimits_{N=1}^{3} C(K,N) + \sum\nolimits_{N=4}^{K} (K-(N-1)) \,. \tag{4}$$

It can be verified that $C(K,3)$ in the above expression is the dominant term, when K is large. This simplifies the proposed algorithm computational complexity to

$$\beta(K) = O\left(K^3\right). \tag{5}$$

Upon comparing Equations (3) and (5), we note that since $O\left(K^3\right) \ll O\left(K^{K/2}\right)$, the proposed algorithm is much simpler and therefore much faster than the exhaustive search. For example, for a network with $K = 50$ molecules, the proposed algorithm complexity is in the order of $50^3 \approx 1.3\times10^5$, which is much smaller than $50^{25} \approx 3\times10^{42}$, the exhaustive search complexity. Regarding the accuracy, we have observed that the differences between the results of the algorithm and the exhaustive search are 0.5% and 0%, for the ERBB and T cell networks, respectively, over all N values for which it was practical to perform the exhaustive search.

METHODS

A. Equations for Computing Vulnerability Levels

Computing the vulnerability level of a molecule or a group of molecules in a network can help to identify and rank the key components of the network, and discover appropriate therapeutics targets. Vulnerability of a molecule can be defined as the probability of having incorrect network responses when the molecule is dysfunctional. The dysfunction state of a molecule can be defined as failure to respond correctly to its input signals. In this paper, we consider stuck-at-0 (SA0) and stuck-at-1 (SA1) fault models to model the dysfunction of molecules, which are constantly 0, inactive, or 1, active, regardless of the activity state of their input signals.

Overall, we compute the vulnerability level of a specific molecule, or a group of molecules, as follows: First, we simulate the fault-free (normal) network and observe the responses at the network output molecule, by applying all input combinations. Then we simulate the abnormal network, where the specific molecule or group of molecules is rendered faulty, by fixing the activity states to 0 for SA0 or to 1 for SA1, and observe the network output responses for all input combinations. Afterwards, we compare the abnormal and normal network responses, to compute the probability of having incorrect network responses, which is the vulnerability level of that specific molecule or group of molecules. How the vulnerability levels are computed is further explained below with mathematical details.

To compute the vulnerability level of a molecule, one needs to first introduce the sample space associated with the correct and incorrect network responses at the output. Suppose K is the number of intermediate molecules in the network. Let *N* be the number of molecules that are simultaneously faulty, i.e., dysfunctional, *I* be the number of the network input combinations, CC be the number of clock cycles (time points) for which the network response is computed, and finally, let *l* be the subscript ranging from 1 to $C(K,N)$, indexing faulty molecules or groups of faulty molecules (an algorithm for determining the required CC is given in Section S1). Then, for each input combination stimulating the network in the presence of *N* faulty molecules, there will be CC number of output responses that may be correct, *c*, or erroneous, *e*, in each clock cycle. Therefore, the sample space *S* can be defined as the set of all possible output sequences of *c* and *e* responses over CC clock cycles, that is

$$S = \{c,e\}^{\mathrm{CC}}. \tag{6}$$

Moreover, for the *l*-th faulty molecule or the *l*-th group of faulty molecules, we define the event S_l, a subset of *S*, as the set of all output sequences of *c* and *e* responses over CC clock cycles for all input combinations, that is

$$S_l = \{v_1, v_2, ..., v_I\}. \tag{7}$$

Note that $v_i \in S$, $i = 1,...,I,$ is a sequence of *c* and *e* of length CC, in which having an *e* in the *t*-th element of v_i means that an erroneous response is observed in the *t*-th clock cycle. Depending on the possible network responses and that which molecule or group of molecules is faulty, v_is may have the same or different probabilities. Also note that some v_is may be identical, therefore, *I* is indeed the maximum number of elements of S_l. For $\mathrm{CC} = 2$ and $I = 2$, for example, we have $S_l = \{v_1, v_2\},$ where $v_1, v_2 \in S = \{c,e\}^2 = \{(c,c),(c,e),(e,c),(e,e)\}.$

In addition, we define CC number of events as follows: $E_1 =$ the event of having an erroneous network response at the output in the 1st clock cycle, …, and $E_{\mathrm{CC}} =$ the event of having an erroneous network response at the output in the CCth clock cycle. Note that E_1 is the set of those *v* elements in S_l in Equation (7) that have an *e* as the 1st entry, E_2 is the set of those *v* elements in S_l that have an *e* as the 2nd entry, and so on. We define the vulnerability level of the *l*-th molecule M_l, $\mathrm{Vul}(M_l)$, as the probability of having an erroneous network response in the 1st clock cycle, …, or in the CCth clock cycle, when M_l is dysfunctional. Therefore, $\mathrm{Vul}(M_l)$ can be written as

$$\mathrm{Vul}(M_l) = P\left(\bigcup_{t=1}^{\mathrm{CC}} E_t \ \& \ M_l \text{ is dysfunctional}\right). \tag{8}$$

Since we consider SA0 and SA1 as the fault models, Equation (8) can be expanded as follows

$$\mathrm{Vul}\left(M_l\right)=P\left(\bigcup_{t=1}^{\mathrm{CC}}E_t \ \& \ M_l \ \text{ is SA0}\right)+P\left(\bigcup_{t=1}^{\mathrm{CC}}E_t \ \& \ M_l \ \text{ is SA1}\right),$$

$$\mathrm{Vul}\left(M_l\right)=P\left(\bigcup_{t=1}^{\mathrm{CC}}E_t\middle|M_l \ \text{ is SA0}\right)P\left(M_l \ \text{ is SA0}\right)+P\left(\bigcup_{t=1}^{\mathrm{CC}}E_t\middle|M_l \ \text{ is SA1}\right)P\left(M_l \ \text{ is SA1}\right). \qquad (9)$$

While we assume equi-probable SA0 and SA1 faults for each molecule, i.e., $P\left(M_l \text{ is SA0}\right)=P\left(M_l \text{ is SA1}\right)=0.5$ in our computations, Equations (8) and (9) can be extended to other fault models and fault probabilities.

Equation (9) is provided for computing the vulnerability level of a single molecule. However, it is also of interest to study the abnormal network responses when multiple molecules are faulty at the same time. In the large T cell network (Figure 3) and for simplicity, in the extreme signaling failure analysis (Figure 4), we assume *N* molecules are either all SA0 or all SA1 at the same time, $N>1$. For this scenario, Equation (9) is still applicable, where M_l needs to be merely replaced with "$M_1, M_2, ..., M_N$". On the other hand, for the small ERBB network (Figure 1) and its extreme signaling failure analysis (Figure 2), we consider all possible SA0 and SA1 fault combinations for *N* dysfunctional molecules, $N>1$. In this scenario, we extend Equation (9) as follows

$$\mathrm{Vul}\left(M_1, M_2, ..., M_N\right)=\sum_{k=1}^{2^N}P\left(\bigcup_{t=1}^{\mathrm{CC}}E_t\middle|\left(M_1, M_2, ..., M_N\right)_k\right)P\left(\left(M_1, M_2, ..., M_N\right)_k\right), \qquad (10)$$

where $\left(M_1, M_2, ..., M_N\right)_k \in \left\{\text{SA0, SA1}\right\}^N$ is the *k*-th fault vector for the group of *N* dysfunctional molecules $M_1, M_2, ...,$ and M_N. An example of how the vulnerabilities are computed on the ERBB signaling network is provided in Section S4.

B. Proposed Algorithm for the Extreme Signaling Failure Analysis

In this section, we provide a detailed explanation of the proposed algorithm. The extreme signaling failure analysis can be performed by an exhaustive search. However, the time needed by the exhaustive search grows exponentially as the size of the network increases, as presented earlier in the paper. To avoid this high computational complexity, we propose the main algorithm with the following four steps

I. First, we compute an upper bound on the number of clock cycles needed for computing the vulnerability levels (Section S1), so that we prevent running network simulations longer than what is needed.

II. Next, we use Equation (10) to compute $\mathrm{Vul}\left(M_{l_1}, M_{l_2}, ..., M_{l_N}\right)$ for N=1,2, and 3. This is motivated by the observation [11] that typically a molecule with high vulnerability appears in larger groups of molecules with high vulnerabilities, and based our experiments, $N \geq 4$ is large enough and provides good accuracy, as described in the paragraph immediately after Equation (5). Thus far, $\max_{l_1} \text{Vul}\left(M_{l_1}\right)$, $\max_{l_1,l_2} \text{Vul}\left(M_{l_1}, M_{l_2}\right)$ and $\max_{l_1,l_2,l_3} \text{Vul}\left(M_{l_1}, M_{l_2}, M_{l_3}\right)$ represent the extreme signaling failures when there are single, double and triple faults, respectively.

III. To determine the extreme signaling failure when there are four simultaneously faulty molecules, $N = 4,$ we pick the molecular triplet, group of $N-1$ faulty molecules, having the highest vulnerability value, e.g., (a,b,c). Then we compute vulnerabilities only for those $K-(N-1)$ quadruplets, groups of N faulty molecules with $N = 4,$ that include (a,b,c), i.e., (a,b,c,M_l). This results in $\max_l \text{Vul}\left(a,b,c,M_l\right)$ as the extreme signaling failure when there are $N = 4$ simultaneous faults.

IV. Then, we repeat Step III for $N = 5,...,K,$ to complete the extreme signaling failure analysis.

Note that this algorithm is not limited to a specific molecular network. Furthermore, in addition to the vulnerability parameter used in this paper, other parameters that quantify and rank the importance of a molecule or a group of molecules can be used as well.

CONCLUSION

Signaling networks in cells are involved in controlling various cellular functions, through different signaling processes. Developing methods for the functional analysis of signaling networks is particularly important for understanding such complex normal and abnormal signaling processes, and can help for untangling the pathology of complex diseases. A fundamental question in systems biology is how vulnerable a signaling network is to the dysfunction of an individual or groups of molecules, where the dysfunction of a molecule is defined as failure to respond correctly to the input signals. The vulnerability levels associated with the dysfunction of molecules or groups of molecules can be measured by computing probabilities of having incorrect network responses in the presence of dysfunctional molecules, using the equations introduced and developed in Methods, Section A.

The focus of this study is to understand what molecule or group of molecules can result in the most detrimental failure in the function of a given network. To answer this question, here we propose a systematic method to identify extreme signaling failures in molecular networks (Methods, Section B). The extreme signaling failure is defined to describe a pathological phenomenon in which the failure of the network passes the physiological tolerable noise threshold, which is quantified as the maximum vulnerability level. The said pathological phenomenon is characterized to be emerged from the presence of one or more dysfunctional molecules in the network. While it is conceivable that different individual dysfunctional molecules may have different vulnerability levels, it is not clear what happens to the vulnerability levels, if two or more molecules are dysfunctional simultaneously.

The extreme signaling failure analysis is initially conducted on the ERBB signaling network (Figure 1). We observe that the maximum vulnerability values do not increase accordingly as the number of concurrently faulty molecules N increases (Figure 2). More precisely, we see a maximum vulnerability increase going from single faults to double faults, $N = 1$ and 2, respectively, and then the maximum vulnerability no longer increases when N increases. Moreover, we observe that the smallest N for which we see the highest maximum vulnerability in this network is $N = 2$, i.e., the double faults. This observation shows that in a network of 18 intermediate molecules, some selective pairs of faulty molecules are sufficient to create the most detrimental harm to the function of the output molecule, and increasing the number of simultaneously faulty molecules does not induce a worse scenario compared to having some specific dysfunctional pairs of molecules.

When extreme signaling failure analysis was done on the T cell signaling network (Figure 3), similar to the ERBB network results, we notice that the maximum vulnerability values do not increase accordingly as the number of simultaneously faulty molecules N increases (Figure 4), for all of T cell network outputs. However, the N value that gives the highest vulnerability varies depending on the output. For the network outputs ap1, bcat and p70s, we see the maximum vulnerability increase going from single faults to double faults, $N = 1$ and 2, respectively, and it reaches a plateau afterwards. For some other network outputs such as cre, nfat, p38, shp2 and sre, the highest maximum vulnerability occurs when $N = 1$. This implies that the functions of these outputs are more dependent on the functions of some specific individual faulty molecules which are sufficient to induce the extreme network failures at these outputs. These observations for both the ERBB and the T cell networks show that it can be enough to have only a few concurrently faulty molecules, to introduce the most detrimental damage to the whole network response, which may eventually result in the development of pathological conditions.

We also show that the computational complexity, i.e., the running time of the proposed extreme signaling failure analysis algorithm (Methods, Section B) is $O\left(K^3\right)$, where K is the number of intermediate molecules in the network. This efficient algorithm is in contrast with an exhaustive search having an exponential running time, $O\left(K^{K/2}\right)$, that quickly becomes impractical to implement, as K increases. For example, for a network with $K = 50$ molecules, the proposed algorithm complexity is in the order of $50^3 \approx 1.3\times10^5$, which is much smaller than $50^{25} \approx 3\times10^{42}$, the exhaustive search complexity.

The extreme signaling failure algorithm makes use of another proposed algorithm (Section S1) that properly incorporates the effects of signaling feedbacks in the extreme signaling failure analysis. Essentially it determines the number of time points (clock cycles) needed for network analysis and simulation while computing the vulnerability levels to find the extreme signaling failures, so that we prevent performing unnecessarily long network simulations. Usefulness of this algorithm is demonstrated by computing the requited number of clock cycles for vulnerability analysis of the ERBB and the T cell signaling networks (Sections S2 and S3, respectively).

Overall, the proposed algorithms have the potential to uncover certain aspects of abnormal signaling network behaviors that can contribute to the development of the pathology, and may suggest some new therapeutic strategies in the area of targeted therapy in pharmaceutical industry. This study is particularly important in the context of complex trait disorders with poorly understood molecular sources when more than one molecule is often reported to be involved in the pathogenesis of the disease.

REFERENCES

1. Saadatpour A, Albert R. Discrete dynamic modeling of signal transduction networks. Methods in Molecular Biology, 2012;880:255-272.
2. Saez-Rodriguez J, Simeoni L, Lindquist JA et al. A logical model provides insights into T cell receptor signaling. PLoS Comput Biol, 2007;3(8).
3. Helikar T, John K, Jack H et al. Emergent decision-making in biological signal transduction networks. Proc. Natl. Acad. Sci. U.S.A, 2008;105(6):1913-1918.
4. Saez-Rodriguez J, Alexopoulos LG, Epperlein J et al. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. Mol Syst Biol, 2009;5:331.
5. Wang RS, Albert R. Elementary signaling modes predict the essentiality of signal transduction network components. BMC Syst Biol, 2011;5:44.
6. Shmulevich I, Dougherty ER, Kim S et al. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics, 2002;18(2):261-274.
7. Ozen M, Lipniacki T, Levchenko A et al. Modeling and measurement of signaling outcomes affecting decision making in noisy intracellular networks using machine learning methods. Integrative Biology, 2020;12(5):122–138.
8. Habibi I, Emamian ES, Simeone O et al. Computation capacities of a broad class of signaling networks are higher than their communication capacities. Phys Biol, 2019;16(6).
9. Abdi A, Tahoori MB, Emamian ES. Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways. Science Signaling, 2008;1(42).
10. Abdi A, Emamian ES. Fault diagnosis engineering in molecular signaling networks: An overview and applications in target discovery. Chemistry and Biodiversity, 2010;7(5):1111–1123.
11. Habibi I, Emamian ES, Abdi A. Advanced fault diagnosis methods in molecular networks. PLoS ONE, 2014;9(10).
12. Habibi I, Emamian ES, Abdi A. Quantitative analysis of intracellular communication and signaling errors in signaling networks. BMC Syst Biol, 2014;8.
13. Emamian ES, Abdi A. Complex human disorders and molecular system engineering: Historical perspective and potential impacts," in Proc. 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, 2009;1083-1085.
14. Emamian ES. AKT/GSK3 signaling pathway and schizophrenia. Frontiers in Molecular Neuroscience, 2012;5.
15. Emamian ES, Kaytor MD, Duvick LA et al. Serine 776 of ataxin-1 is critical for polyglutamine-induced disease in SCA1 transgenic mice. Neuron, 2003;38(3):375-387.
16. Saadatpour A, Albert R. Boolean modeling of biological regulatory networks: A methodology tutorial. Methods, 2013;62(1):3-12.
17. Sahin O, Fröhlich H, Löbke C et al. Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. BMC Syst Biol, 2009;3(1).
18. Cormen TH, Leiserson CE, Rivest RL et al. Introduction to Algorithms. Cambridge: The MIT Press, 2009.

Supplementary Material for "Exploring Extreme Signaling Failures in Intracellular Molecular Networks"

Mustafa Ozen[1], Effat S. Emamian[2], and Ali Abdi[1,3 *]

Mustafa Ozen, PhD
[1]Department of Biochemistry
Vanderbilt University, 2301 Vanderbilt Place, Nashville, TN 37240, USA
Email: mustafa.ozen@vanderbilt.edu

Effat S. Emamian, MD
[2] Advanced Technologies for Novel Therapeutics (ATNT)
116 Millburn Ave, #110, Millburn, NJ 07041, USA
Email: emame@atnt-usa.com

Ali Abdi, PhD
[1] Center for Wireless Information Processing
Department of Electrical and Computer Engineering
[3] Department of Biological Sciences
New Jersey Institute of Technology, 323 King Blvd, Newark, NJ 07102, USA
Email: ali.abdi@njit.edu

* Corresponding Author: Ali Abdi (ali.abdi@njit.edu)

SUPPLEMENTARY SECTIONS

S1 Proposed Algorithm for Determining the Number of Required Clock Cycles to Compute the Vulnerabilities

Modeling and analysis of molecular networks become more challenging if there are positive or negative feedback paths. Due to the feedback mechanisms, network responses may change over time because of some internal compensatory or regulatory mechanisms [1,2]. Feedbacks can cause delays in propagation of signals to the network outputs, while passing through the feedback paths. Therefore, analysis of the effects of feedback in computing vulnerability levels of the network molecules is of interest. More precisely, in this paper we are interested in determining how many clock cycles are needed to compute the vulnerability level of a molecule or a group of molecules, when there are feedbacks in the network. For this purpose, in this section we propose an algorithm that computes an upper bound on the number of clock cycles needed to generate the network response, to calculate its molecular vulnerabilities. Using this algorithm, one can specify how many times the network needs to be simulated, for a normal or abnormal signal to complete its propagation to the network output. This is needed in the proposed main algorithm for the extreme signaling failure analysis (Methods, Section B), to minimize the overall simulation time.

The feedback paths in a network can be modeled by unit-delay memory elements called flip-flops [3]. In a network, if there is only one feedback path, then we intuitively need at most two clock cycles to see full effect of an error, i.e., the effects of an incorrect signal value of a faulty molecule on possibly other molecules and pathways, that collectively determine the network output response. This is because of the delayed response of the flip-flop in the feedback path. In fact, if after the 1st clock cycle there exists an erroneous signal value of a faulty molecule at the input of the feedback flip-flop, then the 2nd clock cycle may be needed for that error to show its full effect at the network output. This is because the feedback-delayed erroneous signal of the faulty molecule may affect some other molecules and pathways in the 2nd clock cycle, which may increase the probability of incorrect responses at the network output. In general, if there exist *F* feedback paths in the network, then we need to simulate the network for at most $F+1$ clock cycles, for an error to show its full effect at the network output. This maximum number of clock cycles is required, if these two conditions hold: (i) all the feedback paths are in the same pathway, connected in series and exhibiting *F* feedback flip-flops; and (ii) after the 1st clock cycle, an error appears at the input of the first feedback flip-flop.

The upper bound of $F+1$ clock cycles can be tightened, by finding the pathways within the network containing the highest number of feedback paths in series. For instance, if there exist *F* feedback paths in the network, with $L \le F$ being the largest number of feedback paths in series on the same pathway, then the maximum clock cycles needed is bounded above by $L+1$, i.e., the number of clock cycles needed for an error to show its full effect is less than or equal to $L+1$. Therefore, to determine the maximum number of required clock cycles, it is sufficient to examine the connections between the network feedback paths and determine how many of them are serially connected in a single pathway. To do this, we define a graph theory topological metric called *closeness* (*CL*). The $0 \le CL\left(X,Y\right) \le 1$ parameter for quantifying the closeness of two molecules *X* and *Y* in a network is the inverse of the distance $d\left(X,Y\right)$ between the two molecules, defined as the length of the shortest path between the two molecules in the network graph

$$CL\left(X,Y\right)=1/d\left(X,Y\right). \qquad \text{(SE1)}$$

Some examples of how to calculate the closeness parameter are presented in Figure S1.

To determine the maximum number of clock cycles needed for computing the vulnerability levels in a network, we propose the following algorithm with four steps

I. Identify the feedback paths and the nodes that initiate the feedbacks, in the network. If they are not known a priori, identify them by finding the loops in the network graph, using a graph algorithm such as the depth-first search algorithm [4].

II. Assume there exist *F* feedback paths in the network. Arbitrarily label the feedback initiating nodes as f_i, $i=1,...,F$. Then start with $i=1$, by calculating the closeness of f_1 with respect to the other feedback initiating nodes f_j s, $j \ne i$ and $j=1,...,F$. If $CL(f_1, f_j)=0$ for all *j* values, it means f_1 is on no other pathway with other feedback initiating nodes, and now f_2 has to be examined similarly, i.e., $i=2$, $j \ne i$ and $j=1,...,F$, and so forth. However, if $CL(f_1, f_j) \ne 0$ for $j=j_0$, then this indicates that there is a path between f_1 and f_{j_0} , i.e., the feedback initiating nodes f_1 and f_{j_0} are on the same pathway. This finding needs to be pursued in the next step.

III. For fixed i and j values, e.g., $i=1$ and $j=j_0$, calculate $CL(f_j, f_k)$ for all other feedback initiating nodes f_k s, $k \ne i, j,$ and $k=1,...,F$. Depending on the *CL* being non-zero or zero, it can be identified if f_k is on the same pathway that includes both f_i and f_j feedback initiating nodes or not.

IV. Repeat Step III until all the feedback initiating nodes are examined.

Using the information obtained from executing the above steps, the algorithm finds the pathway that contains the largest number of feedback initiating nodes on it, in series. With this specific number being called *L*, then at most *L* + 1 clock cycles are needed for computing the vulnerability levels.

Toy examples: Here we compute vulnerability levels in two toy networks (Figure S2A and Figure S2C) that have different number of feedback paths, to describe the relation between *F* and vulnerabilities. Note that since each network has a single pathway, we have $L = F$ in both networks.

The first toy network (Figure S2A) has four nodes: $x_1(t)$ is the input node (molecule), the intermediate nodes are $x_2(t) = x_1(t) \times (\sim x_3(t-1))$ and $x_3(t) = x_2(t)$, and $x_4(t) = x_3(t)$ is the output node, where "$\times$" is used for the AND operation and "$\sim$" is used for the NOT operation, and $x_3(0) = 0.$ Herein, the node x_3 initiates a negative feedback to the node x_2. Since there is only one feedback path in the network, $F = 1,$ at most two clock cycles are enough, $F + 1 = 2,$ to observe the full error effect at the output. To demonstrate this, we compute the vulnerability level of the node x_2 (Figure S2B), for different number of clock cycles $\mathrm{CC} = 1, 2, 3, 4,$ using Equation (9) (Methods, Section A). We observe that the vulnerability of x_2 with 1 clock cycle is 0.5 and it becomes 0.75 with 2 clock cycles, and then remains at 0.75 with 3 and 4 clock cycles. These indicate that the full vulnerability of x_2 is 0.75 that is determined by analyzing the network having feedback for two clock cycles $(F + 1 = 2)$. In other words, two clock cycles are needed for the erroneous x_2 signal values to show their full effects at the network output x_4. Additionally, more clock cycles are not needed.

The second toy network (Figure S2C) has five nodes: $x_1(t)$ is the input node, $x_2(t) = (x_1(t) + x_4(t-1)) \times (\sim x_3(t-1))$, $x_3(t) = x_2(t)$ and $x_4(t) = x_3(t)$ are the intermediate nodes, and $x_5(t)$ is the output node, where "$+$" is used for the OR operation, and $x_3(0) = x_4(0) = 0.$ Here the nodes x_3 and x_4 initiate a negative feedback and a positive feedback to the node x_2, respectively. Since there are two feedback paths in the network, $F = 2,$ at most three clock cycles are needed, $F + 1 = 3,$ to observe the full error effect of x_2 at the output. The computed vulnerability level of x_2 (Figure S2D) for different number of clock cycles corroborates what we stated earlier in this section, that is, $F + 1$ is indeed an upper bound and less number of clock cycles may be needed for computing vulnerabilities in a network with feedbacks. In fact, we observe that the full vulnerability of x_2 is 0.75, obtained using 2 clock cycles only, and analyzing the network for the upper bound of $F + 1 = 3$ clock cycles is not needed (Figure S2D). In other words, 2 clock cycles are enough for errors originated from x_2 to show their complete effects at the output x_5.

S2 ERBB Signaling Network – Vulnerability and the Number of Clock Cycles

In this section, we apply the proposed algorithm in Section S1, to the ERBB signaling network (Figure 1). We start by identifying feedbacks in the network. Given the small size of the network, visual inspection of the network reveals five loops, which topologically may contain the indicators of feedback interactions. The five loops are (i) IGF1R → AKT1 → IGF1R, (ii) IGF1R → MEK1 → ER-α → IGF1R, (iii) CDK4 ⇢ p27 ⇢ CDK4, (iv) CDK4 ⇢ p21 ⇢ CDK4, and (v) CDK2 ⇢ p27 ⇢ CDK2. Despite the existence of five loops, there are only four feedback initiating nodes, AKT1, ER-α, p21 and p27, because p27 is common between two loops, i.e., the loops (iii) and (iv). Note that in the absence of prior information about the feedbacks, the feedback initiators may not be uniquely determined within the identified loops. For example, based on these five loops, one can identify IGF1R, MEK1, CDK4 and CDK2 as feedback initiators as well. Nevertheless, different choices for the feedback initiating molecules within these loops do not affect the algorithm that aims at finding the pathway that contains the largest number of feedback initiating nodes on it, in series. This is because if the feedback initiating nodes f_i and f_j chosen from two loops are connected through a pathway, i.e., $CL(f_i, f_j) \neq 0,$ then other choices of the feedback initiating nodes $f_i^{'}$ and $f_j^{'}$ from the said two loops will be connected through a pathway as well, i.e., $CL(f_i^{'}, f_j^{'}) \neq 0.$ Thus, the algorithm to determine L is independent of the choice of the feedback initiating nodes.

Using the identified feedback initiators, the algorithm outputs the upper bound of $L+1=5$ clock cycles that may be needed for computing the vulnerability level of a molecule. This is because the algorithm identifies a pathway that contains all the feedback initiating molecules on it, in series. For instance, AKT1 → ER-α → p27 ⇢ CDK4 ⇢ p21 ⇢ CDK2 → pRB is a pathway that contains all of the feedback initiating molecules on it. Through this specific pathway, erroneous signals originated from an upstream molecule of AKT1 may require five clock cycles to show their full effects on the output molecule pRB, due to the signal propagation delays introduced by the feedback paths connected in series on the same pathway. Computed using Equation (9) (Methods, Section A), Figure S3 presents the single-fault vulnerability levels versus the number of clock cycles CC for some molecules in the ERBB signaling network. We observe that the vulnerability levels of the molecules can be computed in less than five clock cycles, which confirms that it is sufficient to simulate and analyze the network for at most five clock cycles, as specified by the algorithm.

S3 T Cell Signaling Network – Vulnerability and the Number of Clock Cycles

In this section, we apply the proposed algorithm in Section S1 to the T cell signaling network (Figure 3). In the network, first we identify four feedback initiating nodes that are shp1, ccblp1, pag, and gab2, using the time indices in Table S2 and also by visual inspection of Figure 3. After using the proposed algorithm, we obtain the upper bound of $L+1=5$ clock cycles, because there is one pathway that contains all the four feedback initiating nodes in series. Next, we compute the single-fault vulnerability levels of some molecules versus the number of clock cycles CC (Figure S4A), with cre considered as the output molecule, and using Equation (9) (Methods, Section A). We observe that the vulnerability levels of the molecules can be computed in less than five clock cycles, which confirms that it is sufficient to simulate and analyze the network for at most five clock cycles, as determined by the algorithm.

Furthermore, we compute the double-fault vulnerability values of some pairs of molecules versus the number of clock cycles CC (Figure S4B), with cre considered as the output molecule, and using Equation (9) (Methods, Section A). As anticipated by the algorithm, the vulnerability levels of the molecular pairs can be computed in less than five clock cycles, i.e., it is enough to simulate and analyze the network for at most five clock cycles, irrespective of considering single faults or double faults.

A noteworthy point is that when two molecules are faulty at the same time, more clock cycles may be needed to observe the aggregated full effects of the two erroneous signals at the network output, compared to single faults. However, the upper bound found by the algorithm still works for both scenarios. This is because the upper bound depends only on the topological positions of the feedback initiating nodes and the connections among them, and not on how many nodes are grouped together, to represent a group of faulty molecules.

As some numerical examples, consider the scenario of zap70 and slp76 being individually faulty (Figure S4A), where three and one clock cycles are needed, respectively, to compute their full single vulnerabilities of 0.56 and 0.25, respectively. On the other hand, when they are simultaneously faulty (Figure S4B), four, i.e., more clock cycles are needed to compute their full double vulnerability of 0.56. This indicates that if multiple molecules are faulty concurrently, then there may be further delays in observing the entire effects of multiple erroneous signals, propagating via various pathways towards the network output. Additionally, we observe that the upper bound of $L+1=5$ clock cycles holds true for computing both single and double vulnerabilities.

S4 An Example of Computing Vulnerabilities on the ERBB Signaling Network

In the ERBB signaling network (Figure 1), we have $K=18,\ I=2,\ \text{and}\, \text{CC}=5,$ with the CC being determined using the algorithm introduced in Section S1. When $N=1,$ a single faulty molecule, there are two events associated with the *l*-th faulty molecule M_l being SA0 or SA1

$$S_{l,\text{SA0}}=\{v_1,\ v_2\},\ S_{l,\text{SA1}}=\{v_1^{'},\ v_2^{'}\}, \tag{SE2}$$

where $v_1, v_2, v_1^{'}, v_2^{'} \in S = \{c, e\}^5$ for all $l = 1,...,18.$ Using Equation (9), single-fault vulnerability levels of molecules are calculated for all $l = 1,...,18.$ To illustrate and for $\mathrm{CC} = 2,$ assume hypothetically that for the M_l faulty molecule we have $S_{l,\mathrm{SA0}} = \{(c, c),\ (e, e)\}$ with the probabilities of $P(v_1) = 0.5$ and $P(v_2) = 0.5,$ and $S_{l,\mathrm{SA1}} = \{(c, e)\}$ with the probability of $P(v_1') = 1$. Based on the definition of E_t, it can be shown that $E_1 = E_2 = \{(e,\ e)\}$ when M_l is SA0, whereas $E_1 = \varnothing$ and $E_2 = \{(c,\ e)\}$ when M_l is SA1. Using Equation (9), vulnerability level of *l*-th faulty molecule can be computed for equi-probable SA0 and SA1 faults as follows

$$\begin{aligned}\mathrm{Vul}\left(M_l\right) &= P\left(E_1 \cup E_2 \middle| M_l \text{ is SA0}\right) P\left(M_l \text{ is SA0}\right) + P\left(E_1 \cup E_2 \middle| M_l \text{ is SA1}\right) P\left(M_l \text{ is SA1}\right),\\ &= P\left((e,\ e)\right)0.5 + P\left((c,\ e)\right)0.5,\\ &= P\left(v_2\right)0.5 + P\left(v_1'\right)0.5,\\ &= 0.5 \times 0.5\ +\ 1 \times 0.5 = 0.75.\end{aligned} \tag{SE3}$$

Similarly, when $N = 2,$ a pair of faulty molecules, there are four events associated with each pair of faulty molecules

$$S_{l,\text{SA0-SA0}} = \{v_1,\ v_2\},\ S_{l,\text{SA0-SA1}} = \{v_1^{'},\ v_2^{'}\},\ S_{l,\text{SA1-SA0}} = \{v_1^{''},\ v_2^{''}\},\ S_{l,\text{SA1-SA1}} = \{v_1^{'''},\ v_2^{'''}\}, \tag{SE4}$$

where $v_1, v_2, v_1^{'}, v_2^{'}, v_1^{''}, v_2^{''}, v_1^{'''}, v_2^{'''} \in S = \{c, e\}^5,$ for all $l = 1,...,153$ (there are 153 possible pairs, i.e., $C\left(18,\ 2\right) = 153$). Then, Equation (10) is used to compute the double-fault vulnerability level of each pair as follows

$$\begin{aligned}\mathrm{Vul}\left(M_{l_1}, M_{l_2}\right) = P&\left(\bigcup_{t=1}^{\mathrm{CC}} E_t \middle| M_{l_1} \text{ is SA0 \& } M_{l_2} \text{ is SA0}\right) P\left(M_{l_1} \text{ is SA0 \& } M_{l_2} \text{ is SA0}\right)\\ +P&\left(\bigcup_{t=1}^{\mathrm{CC}} E_t \middle| M_{l_1} \text{ is SA0 \& } M_{l_2} \text{ is SA1}\right) P\left(M_{l_1} \text{ is SA0 \& } M_{l_2} \text{ is SA1}\right)\\ +P&\left(\bigcup_{t=1}^{\mathrm{CC}} E_t \middle| M_{l_1} \text{ is SA1 \& } M_{l_2} \text{ is SA0}\right) P\left(M_{l_1} \text{ is SA1 \& } M_{l_2} \text{ is SA0}\right)\\ +P&\left(\bigcup_{t=1}^{\mathrm{CC}} E_t \middle| M_{l_1} \text{ is SA1 \& } M_{l_2} \text{ is SA1}\right) P\left(M_{l_1} \text{ is SA1 \& } M_{l_2} \text{ is SA1}\right),\end{aligned} \tag{SE5}$$

where $l_1, l_2 = 1,...,18,\ l_1 \neq l_2$, with equi-probable and independent SA0 and SA1 faults. These steps are repeated for all $N > 2,$ to complete the ERBB network extreme signaling failure analysis. Note that what we observe in Figure 2 are maximum vulnerabilities, e.g., $\max_l \mathrm{Vul}(M_l)$ and $\max_{l_1,l_2} \mathrm{Vul}\left(M_{l_1}, M_{l_2}\right)$ for $N = 1 \text{ and } 2$, respectively. To compute the vulnerability levels for the T cell signaling network, the considered parameter values are $K = 64,$ $I = 8,$ and $\mathrm{CC} = 5.$

SUPPLEMENTARY REFERENCES

1. Azpeitia E, Muñoz S, González-Tokman D, et al. The combination of the functionalities of feedback circuits is determinant for the attractors' number and size in pathway-like Boolean networks. Scientific Reports, 2017;7.
2. Somogyi R, Greller LD. The dynamics of molecular networks: Applications to therapeutic discovery. Drug Discovery Today, 2001;6(24):1267-1277.
3. Abdi A, Tahoori MB, Emamian ES. Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways. Science Signaling, 2008;1(42).
4. Cormen TH, Leiserson CE, Rivest RL et al. Introduction to Algorithms. Cambridge: The MIT Press, 2009.

SUPPLEMENTARY FIGURES

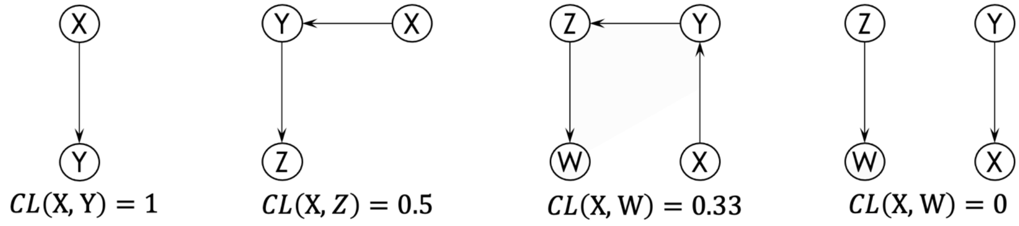
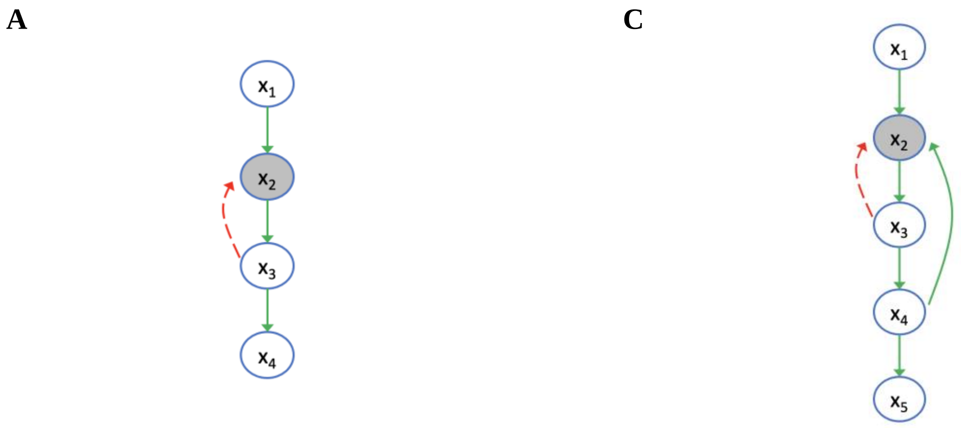


Figure S1. Numerical examples of the closeness parameter *CL* between two molecules.



B

$x_1(t)$ $t=1,2,3,4$	$x_2(t)$ $t=1,2,3,4$	$x_4(t)$			
		$t=1$	$t=2$	$t=3$	$t=4$
0	Fault free	0	0	0	0
1		1	0	1	0
0	Faulty (SA0)	0	0	0	0
1		0	0	0	0
0	Faulty (SA1)	1	1	1	1
1		1	1	1	1
Vulnerability of x_2 (CC = 1)		0.5			
Vulnerability of x_2 (CC = 2)		0.75			
Vulnerability of x_2 (CC = 3)		0.75			
Vulnerability of x_2 (CC = 4)		0.75			

D

$x_1(t)$ $t=1,2,3,4$	$x_2(t)$ $t=1,2,3,4$	$x_5(t)$			
		$t=1$	$t=2$	$t=3$	$t=4$
0	Fault free	0	0	0	0
1		1	0	1	0
0	Faulty (SA0)	0	0	0	0
1		0	0	0	0
0	Faulty (SA1)	1	1	1	1
1		1	1	1	1
Vulnerability of x_2 (CC = 1)		0.5			
Vulnerability of x_2 (CC = 2)		0.75			
Vulnerability of x_2 (CC = 3)		0.75			
Vulnerability of x_2 (CC = 4)		0.75			

Figure S2. Toy networks illustrating the number of clock cycles needed for the erroneous signal of a faulty (dysfunctional) molecule to show its full effect at the network output. (A) Toy network with one feedback path. (B) Output truth table for fault-free and faulty x_2. (C) Toy network with two feedback paths. (D) Output truth table for fault-free and faulty x_2. Note: The green solid edges represent activatory interactions and the red dashed edges represent inhibitory interactions.

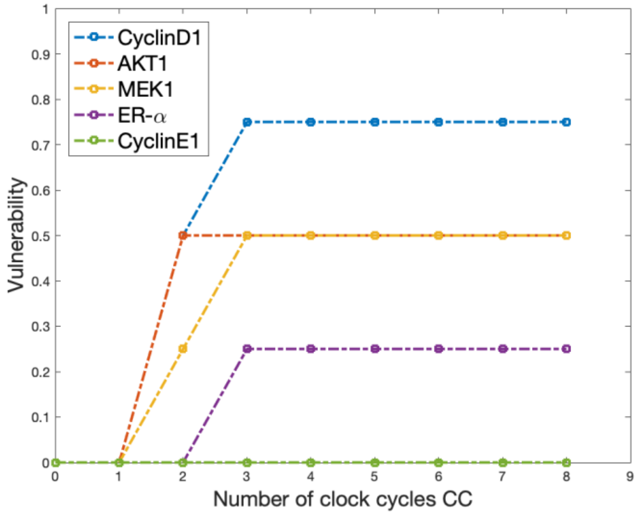


Figure S3. Vulnerability versus the number of clock cycles CC for some molecules in the ERBB signaling network.

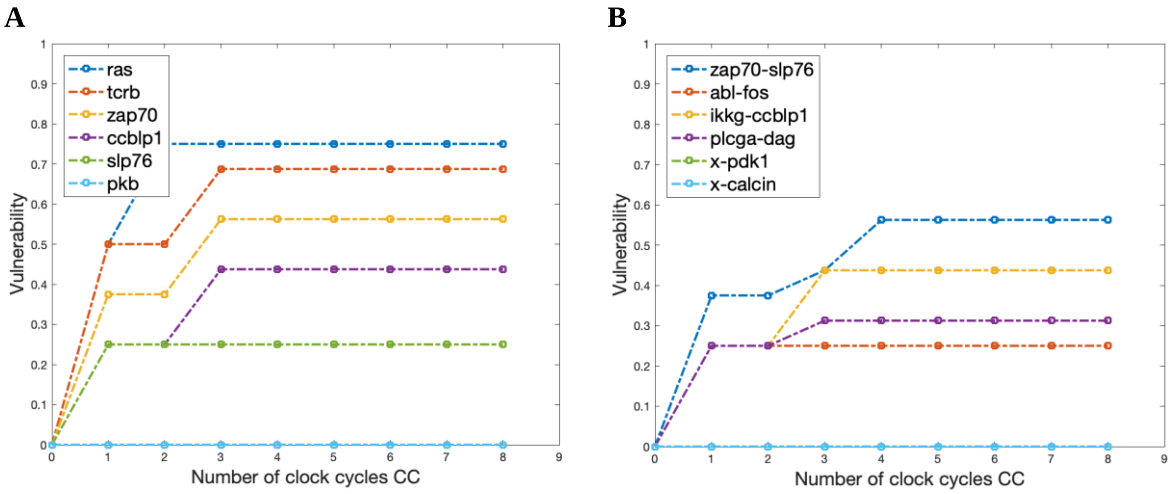


Figure S4. Vulnerability versus the number of clock cycles CC for some single and pairs of molecules in the T cell signaling network, with “cre” considered as the output molecule. (A) Single-fault vulnerability levels. (B) Double-fault vulnerability levels.

SUPPLEMENTARY TABLES

Table S1. The Boolean equations for the ERBB signaling network (Figure 1) provided in [17]. In the equations, “x” is used for the AND operation, “+” is used for the OR operation, and “~” is used for the NOT operation.

Molecules	Boolean Equations
AKT1	AKT1 = ERBB1 + ERBB1_2 + ERBB1_3 + ERBB2_3 + IGF1R
c-MYC	c-MYC = AKT1 + MEK1 + ER-α
CDK2	CDK2 = CyclinE1 x (~ p21) x (~ p27)
CDK4	CDK4 = CyclinD1 x (~ p21) x (~ p27)
CDK6	CDK6 = CyclinD1
CyclinD1	CyclinD1 = ER-α x c-MYC x (AKT1 + MEK1)
CyclinE1	CyclinE1 = c-MYC
EGF	EGF: Input
ER-α	ER-α = AKT1 + MEK1
ERBB1	ERBB1 = EGF
ERBB1_2	ERBB1_2 = ERBB1 x ERBB2
ERBB1_3	ERBB1_3 = ERBB1 x ERBB3
ERBB2	ERBB2 = EGF
ERBB2_3	ERBB2_3 = ERBB2 x ERBB3
ERBB3	ERBB3 = EGF
IGF1R	IGF1R = (ER-α + AKT1) x (~ ERBB2_3)
MEK1	MEK1 = ERBB1 + ERBB1_2 + ERBB1_3 + ERBB2_3 + IGF1R
p21	p21 = ER-α x (~ CDK4) x (~ AKT1) x (~ c-MYC)
p27	p27 = ER-α x (~ CDK4) x (~ CDK2) x (~ AKT1) x (~ c-MYC)
pRB	pRB = (CDK4 x CDK6) + (CDK4 x CDK6 x CDK2)

Table S2. The Boolean equations for the T cell signaling network (Figure 3), provided in [2]. In the equations, “x” is used for the AND operation. “+” is used for the OR operation, and “~” is used for the NOT operation. The symbol “t” represents the current time whereas “t+1” stands for the next time instant.

Molecules	Boolean Equations
abl(t)	abl(t) = lckp1(t) + fyn(t)
akap79	akap79 = 0
ap1(t)	ap1(t) = fos(t) x jun(t)
bad(t)	bad(t) = ~ pkb(t)
bcat(t)	bcat(t) = ~ gsk3(t)
bcl10	bcl10 = 1
bclx1(t)	bclx1 = ~ bad(t)
ca(t)	ca(t) = ip3(t)
cabin1(t)	cabin1(t) = ~ camk4(t)
calcin(t)	calcin(t) = (~ cabin1(t)) x (~ akap79) x (~ calpr1) x cam(t)
calpr1	calpr1 = 0
cam(t)	cam(t) = ca(t)
camk2(t)	camk2(t) = cam(t)
camk4(t)	camk4(t) = cam(t)
card11	card11 = 1
card11a(t)	card11a(t) = card11 x bcl10 x malt1
cblc(t+1)	cblb(t+1) = ~ cd28
ccblp1(t+1)	ccblp1(t+1) = zap70(t)
ccblp2(t+1)	ccblp2(t+1) = fyn(t)
cd28	Input
cd4	Input
cd45	cd45 = 1
cdc42	cdc42 = 0
cre(t)	cre(t) = creb(t)
creb(t)	creb(t) = rsk(t)
csk(t)	csk(t) = pag(t)

Molecules	Boolean Equations
cyc1(t)	cyc1(t) = ~ gsk3(t)
dag(t)	dag(t) = (~ dgk(t)) x plcga(t)
dgk(t+1)	dgk(t+1) = tcrb(t)
erk(t)	erk(t) = mek(t)
fkhr(t)	fkhr(t) = ~ pkb(t)
fos(t)	fos(t) = erk(t)
fyn(t)	fyn(t) = tcrb(t) + (lckp1(t) x cd45)
gab2(t+1)	gab2(t+1) = lat(t) x zap70(t) x (gads(t) + grb2(t))
gadd45	gadd45 = 1
gads(t)	gads(t) = lat(t)
gap	gap = 0
grb2(t)	grb2(t) = lat(t)
gsk3(t)	gsk3(t) = ~ pkb(t)
hpk1(t)	hpk1(t) = lat(t)
ikb(t)	ikb(t) = ~ ikkab(t)
ikkab(t)	ikkab(t) = ikkg(t) x camk2(t)
ikkg(t)	ikkg(t) = pkcth(t) x card11a(t)
ip3(t)	ip3(t) = plcga(t)
itk(t)	itk(t) = slp76(t) x zap70(t) x pip3(t)
jnk(t)	jnk(t) = mekk1(t) + mkk4(t)
jun(t)	jun(t) = jnk(t)
lat(t)	lat(t) = zap70(t)
lckp1(t)	lckp1(t) = (~ shp1(t)) x (~ csk(t)) x cd45 x cd4
lckp2(t)	lckp2(t) = tcrb(t)
malt1	malt1 = 1
mek(t)	mek(t) = raf(t)
mekk1(t)	mekk1(t) = hpk1(t) + cdc42 + rac1p2(t)
mkk4(t)	mkk4(t) = mlk3(t) + mekk1(t)
mlk3(t)	mlk3(t) = hpk1(t) + rac1p1(t)
nfat(t)	nfat(t) = calcin(t)

Molecules	Boolean Equations
nfkb(t)	nfkb(t) = ~ ikb(t)
p21c(t)	p21c(t) = ~ pkb(t)
p27k(t)	p27k(t) = ~ pkb(t)
p38(t)	p38(t) = ((~ gadd45) x zap70(t)) + mekk1(t)
p70s(t)	p70s(t) = pdk1(t)
pag(t)	pag(t) = ~ tcrb(t)
pag(t+1)	pag(t+1) = fyn(t)
pdk1(t)	pdk1(t) = pip3(t)
pi3k(t)	pi3k(t) = ((~ cblb(t)) x X(t)) + ((~ cblb(t)) x lckp2(t))
pip3(t)	pip3(t) = pi3k(t) x (~ ship1) x (~ pten)
pkb(t)	pkb(t) = pdk1(t)
pkcth(t)	pkcth(t) = pdk1(t) x dag(t) x vav1(t)
plcga(t)	plcga(t) = plcgb(t) x (~ ccblp2(t)) x slp76(t) x zap70(t) x vav1(t) x (itk(t) + rlk(t))
plcgb(t)	plcgb(t) = lat(t)
pten	pten = 0
rac1p1(t)	rac1p1(t) = vav1(t)
rac1p2(t)	rac1p2(t) = vav3(t)
raf(t)	raf(t) = ras(t)
ras(t)	ras(t) = (~ gap) x rasgrp(t) x sos(t)
rasgrp(t)	rasgrp(t) = dag(t)
rlk(t)	rlk(t) = lckp1(t)
rsk(t)	rsk(t) = erk(t)
sh3bp2(t)	sh3bp2(t) = zap70(t) x lat(t)
ship1	ship1 = 0
shp1(t+1)	shp1(t+1) = (~ erk(t)) x lckp1(t)
shp2(t)	shp2(t) = gab2(t)
slp76(t)	slp76(t) = (~ gab2(t)) x zap70(t) x gads(t)
sos(t)	sos(t) = grb2(t)
sre(t)	sre(t) = rac1p2(t) + cdc42

Molecules	Boolean Equations
tcrb(t)	tcrb(t) = (~ ccblp1(t)) x tcrlig
tcrlig	Input
tcrp(t)	tcrp(t) = (tcrb(t) x lckp1(t)) + (tcrb(t) x fyn(t))
vav1(t)	vav1(t) = (sh3bp2(t) x zap70(t)) + X(t)
vav3(t)	vav3(t) = sh3bp2(t)
X(t)	X(t) = cd28
zap70(t)	zap70(t) = (~ ccblp1(t)) x abl(t) x tcrp(t)