Analyzing gene regulatory networks by comparing the dynamics obtained via DSGRN (Dynamic Signatures Generated by Regulatory Networks) and RACIPE (Random Circuit Perturbation)

Aaron Scheiner[1], Konstantin Mischaikow[1,2], Marcio Gameiro[1,2]

[1]Rutgers University–New Brunswick
[2]DIMACS

Abstract

In this research project, we are studying the analyses generated by Huang et al. using random circuit perturbation (RACIPE). After gaining a comprehensive understanding of Huang's paper, we use RACIPE to reproduce these results. We move from there to produce analogous results in DSGRN. After both sets of results have been generated, we will compare our RACIPE results to our DSGRN results. The results of the comparisons between our RACIPE and DSGRN results could potentially lead to a paper describing the results and could also lead to additional problems that could be studied during the research project or afterwards.

Introduction

Gene regulatory networks (GRNs) are networks formed from pairwise interactions between molecular regulators, including proteins, genes, and signaling molecules [1]. Understanding GRNs could lead to the targeting of many diseases without the dangerous side effects that characterize plenty of modern drugs [1]. When modeling GRNs, the most popular approach is the ordinary differential equation (ODE) method. These methods "provide a good quantitative match and are easily generalized" in the method landscape [1]. The major hurdle when using ODE methods is the necessity for a multitude of kinetic parameters [1]. Random circuit perturbation (RACIPE) is a method for modeling GRNs that circumvents this hurdle by random sampling and the generation of a great number of models. RACIPE is a popular method capable of handling larger networks. Another method that circumvents the hurdle is Dynamic Signatures Generated by Regulatory Networks (DSGRN). DSGRN combines the Boolean and ODE modeling methods for a valuable and mathematically rich GRN modeling method. DSGRN avoids the problem of needing many kinetic parameters by computing course information about the dynamics of a network. Coarse information means regions in phase space that likely contain invariant sets including fixed points, periodic orbits, etc. By computing coarse information, DSGRN can give information about the dynamics of a network for all parameter values via parameter space decomposition.

Background Information

Huang et al. assert in [4] that the core gene circuit in a gene regulatory network determines the dynamics of the network, not the kinetic parameters. To this effect, RACIPE generates many random kinetic models from a fixed circuit topology. Specifically, RACIPE:

- Randomizes all key parameters in set ranges, except the threshold values

- Solves the rate equations, which are comprised of Hill functions and additional (g, k) parameters
- Finds steady states
- Repeats the process

After the models have been generated, statistical analysis is employed to determine robust dynamical properties about the input gene circuit [4]. This process is visualized below:

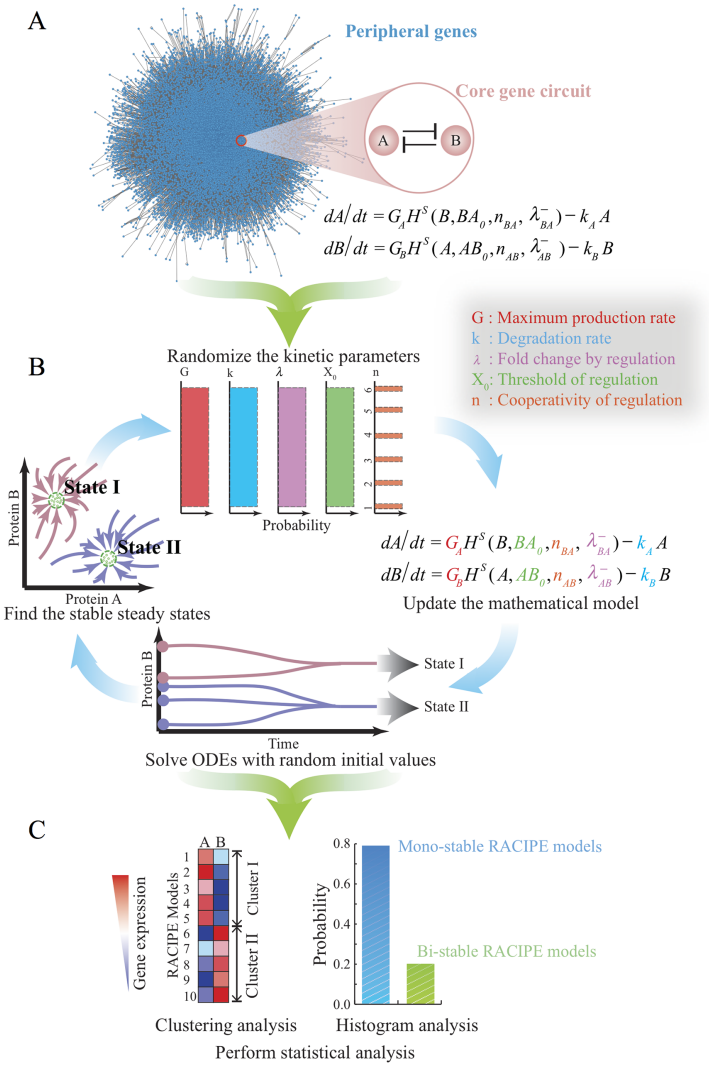

Fig. 1. The process of random circuit perturbation (RACIPE) [4].

Meanwhile, DSGRN makes use of combinatorics and algebraic topology when modeling GRNs. It focuses on "multi-level discrete maps—a direct generalization of Boolean maps—that are compatible with an ODE system" [1]. In particular, they "propose that only the asynchronous updates of these discrete maps have biological meaning" [1]. They then look at a "class of ODEs that can be viewed as a continuous parameterization of a family of multi-level discrete maps" [1].

DSGRN models the dynamics of gene regulatory networks as a system of equations expressed in terms of step functions with one step discontinuity, which represents approximations for Hill function models of the dynamics. The software computes the parameter graph for an input regulatory network. For each node of the parameter graph, DSGRN can find the associated morse graph, whose nodes are nontrivial strongly connected path components. These morse nodes represent stable recurrent dynamics for each parameter. The collection of morse graphs for each parameter in the parameter graph is referred to as a "DSGRN database" [1].

Methods

A. Notation and Circuitry

2.1. Regulatory networks.

Definition 2.1. *A* regulatory network $\mathbf{RN} = (V, E)$ *is an annotated finite directed graph with vertices* $V = \{1, \ldots, N\}$ *called* network nodes *and annotated directed edges* $E \subset V \times V \times \{\rightarrow, \dashv\}$ *called* interactions. *An* $\rightarrow$ *annotated edge is referred to as an* activation *and an* $\dashv$ *annotated edge is called a* repression. *We indicate that either* $i \rightarrow j$ *or* $i \dashv j$ *without specifying which by writing* $(i, j) \in E$. *We allow for self-edges but admit at most one edge between any two nodes, e.g., we cannot have both* $i \rightarrow j$ *and* $i \dashv j$ *simultaneously. The set of* sources *and* targets *of a node* n *are denoted by*

$$\mathbf{S}(n) := \{i \mid (i, n) \in E\} \quad \text{and} \quad \mathbf{T}(n) := \{j \mid (n, j) \in E\}.$$

Fig. 2. The DSGRN definition of regulatory networks [2].

The above passage explains the notation of constructing graphs to model GRNs.

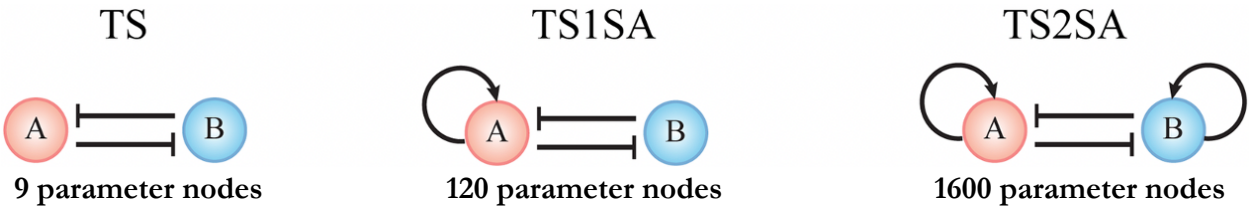

Fig. 3. Three toggle switch motifs varying in the amount of self-activating edges present and the number of nodes in their respective parameter graphs [4].

These are the three toggle switch motifs that we used when comparing RACIPE and DSGRN. Based on the passage above, these networks are comprised of two network nodes, or genes, that each inhibit the other. The difference between each of the models is the amount of self-activating edges. The equations that will be discussed are the equations corresponding to the TS network. The data generated and shown in tables corresponds to the TS2SA network. These toggle switch motifs are classical circuits of gene regulatory network architecture that form the basis for many complex networks [5]. The TS network works well in DSGRN, as it allows us to simulate the effects of compounds.

B. Equations

RACIPE employs deterministic rate equations to model input core gene circuits [4]. These rate equations are comprised of three parts: basal production rate, one or more Hill functions, and degradation rate. The Hill functions are non-linear shifted Hill functions, which in turn are made up of an inhibitory Hill function and the switching mechanic formed by the fold change and the fold change subtracted by one [4]. For the simplest toggle switch, the rate equation appears in the paper as follows:

$$\dot{A} = g_A H^S(B, BA_0, n_{BA}, \lambda^-_{BA}) - k_A A$$
$$\dot{B} = g_B H^S(A, AB_0, n_{AB}, \lambda^-_{AB}) - k_B B \quad ,$$

Eq. 1. The unexpanded rate equations for TS [4].

Here, H^S is the non-linear shifted Hill function. Here are the rate equations are fully expanded:

$$\dot{A} = g_A(\lambda^-_{BA} + (1 - \lambda^-_{BA})(1/(1+(B/BA_0)^{n_{BA}}))) - k_A A$$
$$\dot{B} = g_B(\lambda^-_{AB} + (1 - \lambda^-_{AB})(1/(1+(A/AB_0)^{n_{AB}}))) - k_B B$$

Eq. 2. The expanded rate equations for TS.

In these equations, g_B refers to the basal production rate (the production rate without any regulator bound to the promoter) of B and k_B means the innate degradation rate of B. λ^-_{AB} is the maximum fold change of the level of gene B caused by the inhibition of gene A. This value ranges in [0, 1). In the case of activation, λ^+_{AB} is greater than 1. The threshold is denoted by AB_0 [4]. The n_{AB} term is the Hill coefficient (a major component in the comparison between DSGRN and RACIPE). The importance of this Hill function comes from $\left(\frac{A}{AB_0}\right)^{n_{AB}}$. If the gene is above the threshold, then $\left(\frac{1}{1+\left(\frac{A}{AB_0}\right)^{n_{AB}}}\right)$ tends to 0 (as the Hill coefficient increases). If the gene is below the threshold, however, the expression tends to 1 (as the Hill coefficient increases). When the expression equals 0, the production half of the rate equation equals $g_B*(\lambda^-_{AB})$. When the expression equals 1, the production half of the rate equation equals g_B. This becomes extremely important when the degradation term of the rate equation is greater than one and less than the other. In such a case, gene A being above or below the threshold determines the sign of the rate equation. This will prove relevant when looking at the mathematics underlying DSGRN. The above information is true for the variables in the $\dot{A}$ rate equation as well [4].

Toggle-switch circuit (TS):

$$\dot{A} = G_A H^S(B, BA_0, n_{BA}, \lambda^-_{BA}) - k_A A$$
$$\dot{B} = G_B H^S(A, AB_0, n_{AB}, \lambda^-_{AB}) - k_B B$$

Toggle-switch circuit with one-sided self-activation (TS1SA):

$$\dot{A} = G_A H^S(B, BA_0, n_{BA}, \lambda^-_{BA}) H^S(A, AA_0, n_{AA}, \lambda^+_{AA}) / \lambda^+_{AA} - k_A A$$
$$\dot{B} = G_B H^S(A, AB_0, n_{AB}, \lambda^-_{AB}) - k_B B$$

Toggle-switch circuit with two-sided self-activation (TS2SA):

$$\dot{A} = G_A H^S(B, BA_0, n_{BA}, \lambda^-_{BA}) H^S(A, AA_0, n_{AA}, \lambda^+_{AA}) / \lambda^+_{AA} - k_A A$$
$$\dot{B} = G_B H^S(A, AB_0, n_{AB}, \lambda^-_{AB}) H^S(B, BB_0, n_{BB}, \lambda^+_{BB}) / \lambda^+_{BB} - k_B B$$

Eq. 3. The unexpanded rate equations for TS.
Eq. 4. The unexpanded rate equations for TS1SA.
Eq. 5. The unexpanded rate equations for TS2SA [4].

The above equations are the unexpanded rate equations for the three toggle switch motifs used in the comparison. These equations will prove relevant when we connect RACIPE's Hill functions to DSGRN's rate equations.

DSGRN includes all possible parameter values by mapping over parameter space.

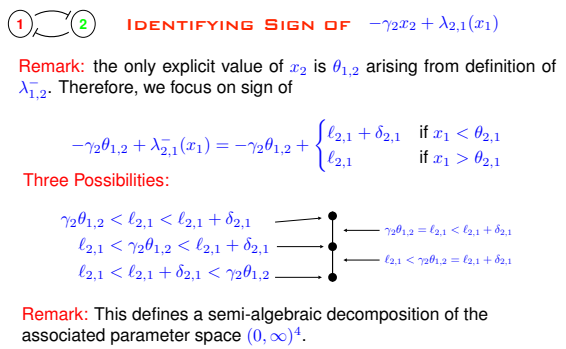

Fig. 4. (Left). TS inequalities and step function.

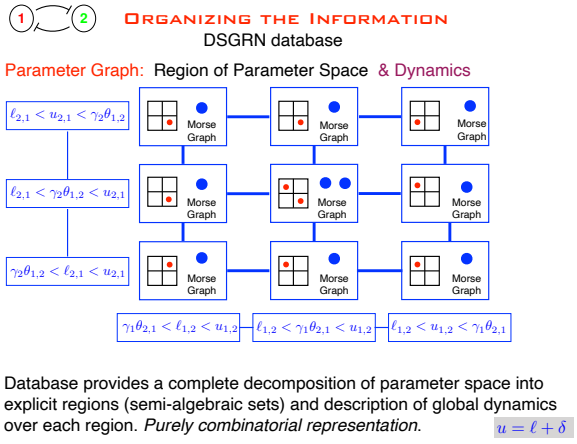

Fig. 4. (Right). TS parameter space for DSGRN.

Here, we can see how DSGRN models TS, the toggle switch with no self-activation:

$$\dot{x_1} = -\gamma_1 x_1 + \lambda_{1,2}(x_2) = -\gamma_1 x_1 + \begin{cases} l_{1,2} + \delta_{1,2} \; if \; x_2 < \theta_{1,2} \\ l_{1,2} \; if \; x > \theta_{1,2} \end{cases}$$

$$\dot{x_2} = -\gamma_2 x_2 + \lambda_{2,1}(x_1) = -\gamma_2 x_2 + \begin{cases} l_{2,1} + \delta_{2,1} \; if \; x_2 < \theta_{2,1} \\ l_{2,1} \; if \; x > \theta_{2,1} \end{cases}$$

Eq. 6. DSGRN TS step functions.

In the TS parameter space visualization shown above, $u = l + \delta$. These step functions are related to RACIPE's rate equations, particularly when examining the Hill function component. The two options of a gene being above or below the threshold are shown in RACIPE and DSGRN. Thus, the data used in RACIPE simulations and the simulation outputs can be used in DSGRN. In regard to the Hill coefficient, DSGRN treats the Hill coefficient as infinity. Thus, DSGRN uses a step function. RACIPE, however, favors lower (and more biologically relevant) Hill coefficient values, so the inequalities that define DSGRN cannot be formed from RACIPE's rate equations.

For each parameter node in the parameter space, DSGRN generates a morse graph based on the inequalities associated with that parameter node.

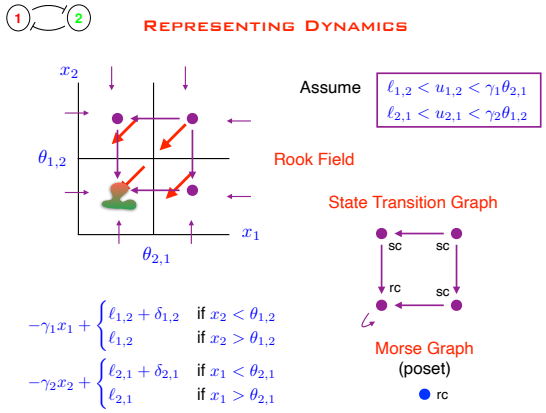

Fig. 5. DSGRN morse graph computation for one TS parameter node.

The above figure shows the steps taken to output the morse graph. Firstly, a rook field is constructed based on the inequalities, with the two genes as the x and y axes. The red arrows are first added to show the direction of each region of the rook field, which is synonymous with phase space. After adding the purple nodes and edges, we observe the direction of the graph and form a state transition graph, in which the node with only incoming edges is called a recurrent component or a stable fixed point. These points led to the generation of the morse graph. As you can see in the 3x3 visualization of the parameter space of the TS network, the middle parameter node has two fixed points. The number of fixed points gives us information about the kind of stability in that parameter region. In the parameter space for the TS network, the middle parameter node is bistable, and all the other nodes are monostable.

C. RACIPE Sampling

Parameters	Min to Max Values (Uniform)	Mean, Standard Deviation (Rectified Gaussian)[+]	Mean (Exponential)
Maximum production rate (G)	1-100	50.5, 49.5	50.5
Degradation rate (k)	0.1-1	0.55, 0.45	0.55
Fold change (λ)[*]	1-100	50.5, 49.5	50.5
Threshold (X_0)	The ranges, which depend on the inward regulations, are estimated by a Monte Carlo simulation.		
Hill coefficient (n)[#]	1-6	3.5, 2.5	3.5

Table 1. RACIPE sampling [4].

Aside from the threshold values, RACIPE samples its parameters in set ranges, though the fold change requires explanation as denoted by "*". When it is an activating edge, the fold change ranges in 1-100. For an inhibiting edge, the fold change ranges between 0.01-1 [4]. RACIPE allows the user to affect the simulation in a variety of ways, including changing parameter ranges. When running RACIPE simulations to compare to DSGRN, we changed the Hill coefficient constantly. For the threshold values, RACIPE employs the "half-functional rule," which samples the threshold values such that "each regulatory link should have roughly equal chance of being functional or not functional" [4]. Thus, the thresholds are sampled such that the level of a gene at the steady states is equally likely to be above or below its threshold level. The half-functional rule was included because, "if the threshold level is too large or too small, the regulatory link is either not functional most of the time or constitutively active, thereby changing the effective circuit topology, and limiting the comprehensive understanding of circuit function" [4].

D. Testing and Simulations

For our RACIPE simulations, we changed parameter ranges and important values in the program. These included the Hill coefficient range (labeled minN and maxN), the number of models generated by RACIPE (num_paras), the number of initial conditions to solve the ODE (num_ode), and the ODE solver used (1 for the Euler ODE solver and 2 for the Runge-Kutta-Fehlberg (RK45) method). In general, we decided to set both minN and maxN to the same value, and we chose the following values: 2, 4, 6, 10, 20, 30. The first three are more biologically relevant, and the last three are closer to the results we would receive from DSGRN. Regarding the number of models, we set num_paras to either 1000 or 10000 (100 by default). For the TS network, we used 1000 models, and for TS1SA and TS2SA, we used 10000 models. Our computations were mostly focused on TS2SA, as it exhibits the most complex behavior and the highest levels of stability. Thus, for most of our computations, num_paras = 10000. The num_ode value was usually 1000 in our simulations, though we also tried 500 and 2000 to see if changing it led to any significant differences. We initially used the faster Euler ODE solver before moving to the more precise RK45 solver. In terms of how the results actually display, RACIPE outputs the number of models with each type of stability. It also outputs the running time for each simulation. For Hill coefficient values within the default range, the ratio between monostability and bistability is roughly 4:1 for the TS network. Increasing the Hill

coefficient leads the ratio to approach 1:1. In general, increasing the Hill coefficient increases the percentages of higher levels of stability.

DSGRN is more variable than RACIPE, and we can use it to output a variety of result formats depending on what we want to see. For example, one can use DSGRN to output morse graphs, stability percentages, stability counts, parameter inequalities, and more. We began by trying to get the stability percentages to compare to our RACIPE data. Unlike RACIPE, which uses random sampling, DSGRN's stability percentages are fixed.

E. Essential Parameters

We set out to test out the results generated by Huang et al. in DSGRN. This entails adapting RACIPE's half-functional rule in DSGRN. We decided to adapt the half-function rule in DSGRN by using essential parameters and their neighbors. Essential parameters are parameter nodes where the edges have probability to be both active and inactive. The neighbors are adjacent nodes to essential nodes. Thus, for the 3x3 parameter space visualization of the TS network, including only the essential nodes and their neighbors (though there is only one essential node) yields the following parameter regions:

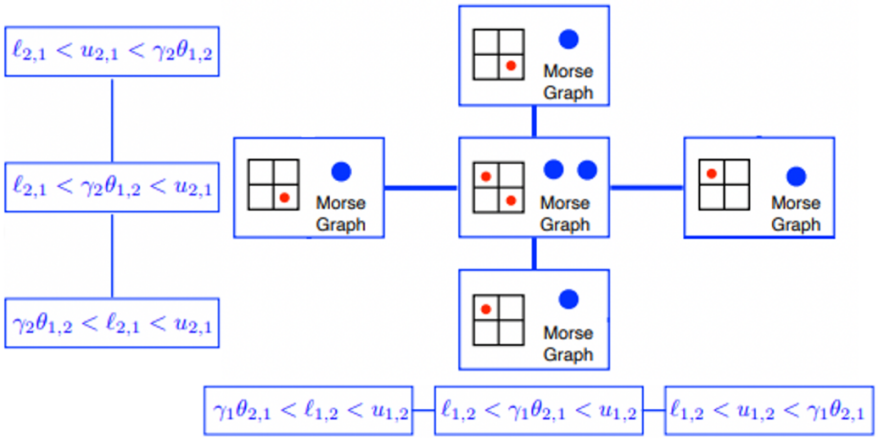

Fig. 6. TS essential parameter node and its neighbors.

Here, the bistable node in the middle is the essential node. There, the threshold value is in between the upper and lower values. Therefore, one gene's level being above or below the threshold determines whether or not that gene is inhibiting the other. This also determines the sign of the rate equation output for the other gene. Therefore, that middle bistable node is the essential node, and the four adjacent nodes are its neighbors.

F. Weighting DSGRN with RACIPE data

After generating our initial tables for RACIPE and DSGRN data, we sought to use DSGRN to interpret the RACIPE outputs. By using Lun Zhang's code, we were able to connect the outputs of RACIPE to the terms of parameter inequalities in DSGRN. This goes back to the relationship between DSGRN's step functions and RACIPE's rate equations containing Hill functions. To find

the parameter nodes that the RACIPE models belong to, we require the three parts of the inequality above, which we will refer to as L, U, and T. We get these from the parameters that go into the rate equations. For inhibition, the fold change ranges from 0.01 to 1, so L = g*λ. Then we get U = g. The T value of the inequality represents the threshold times the degradation. Thus, it would be the threshold (AB_0, BA_0, AA_0, or BB_0) times the degradation rate of the gene listed second in the threshold. For self-activation, which is the only type of activation we are concerned with, the differences arise with the L and U values. Since the fold change now ranges from 1 to 100, L = g and U = g*λ.

When a gene has two inward edges (including self-activating edges), we must split the basal production rate g between the two Hill functions in the rate equation. We decided to split g into $\sqrt{g}$ and $\sqrt{g}$, though g and 1 work as well with no change in the results. We tested to make sure both ways of splitting g resulted in the same results on the TS1SA network, and both ways yielded identical results.

In terms of what we did with the RACIPE data in DSGRN, we sorted the RACIPE models based on their respective parameter nodes using the Lun Zhang's code. From there, we weighted parameter nodes with a nonzero amount of RACIPE models and calculated the stability of each of these parameters. Finally, we added all of the monostable parameters' weights and did the same for the other four types of stability. In the second result table, there is a row that shows the percentages after this procedure was done on RACIPE TS2SA data with 10000 models, the RK45 solver, and the default Hill coefficient range of 1 to 6. This represents the result of weighting, and its significance will be explained in the results section.

Results

The following result concerns the effect of the number of initial conditions to solve the ODE. The labels for percentages of each type of stability list 1 followed by a percent sign to denote monostability and that it is a percentage and not a count. Hence, the next column represents the percentage of bistability among the models, and this trend continues to pentastability. These results show that changing num_ode does not really affect the stability percentages, though the running time does scale linearly with num_ode.

model	solver	num_paras	minN/maxN	num_ode		stable: 1 %	stable: 2 %	stable: 3 %	stable: 4 %	stable: 5 %		running time (seconds)	running time (hours)
TS2SA	2	10000	30	500		7.86	45.75	41.34	3.89	1.16		3272.614021	0.909059
TS2SA	2	10000	30	1000		7.78	45.01	41.79	4.14	1.28		6752.532845	1.875704
TS2SA	2	10000	30	2000		7.94	45	41.78	4.23	1.05		12383.88167	3.439967

Table 2. The effects of the number of initial conditions to solve the ODE.

The table below contains the main findings to date. For this table, we need only focus on the rows where num_paras = 10000 and the rows under those. Here, we can see the effects of changing the Hill coefficient. Additionally, the DSGRN TS2SA stability percentages are attached. This table leads to some important conclusions. Firstly, for TS2SA, increasing the Hill coefficient leads to a decrease in the monostability percentage and increases in the tristability, tetrastability, and pentastability percentages. We came to the conclusion that the Hill coefficient's impact on the percentages weakens for $N \geq 20$. As DSGRN treats the Hill coefficient as infinity, we see that the

RACIPE percentages at N = 20 and N = 30 are comparable to the DSGRN data, particularly the data for the 756 essential and neighbor nodes. Moreover, the DSGRN weighting percentages are very close to the high N RACIPE data. The importance of this closeness will be elaborated upon in the conclusion. Additionally, the third to last row reiterates the fact that essential parameters have 0% monostability.

model	solver	num_paras	minN/maxN	num_ode	stable: 1 %	stable: 2 %	stable: 3 %	stable: 4 %	stable: 5 %	running time (seconds)	running time (hours)
TS2SA	2	1000	2	1000	42.3	53.5	4.2	0	0	633.454939	0.17596
TS2SA	2	1000	4	1000	15.6	65.4	18.3	0.6	0.1	609.400952	0.169278
TS2SA	2	1000	6	1000	12.4	58.4	28.1	0.9	0.2	656.392266	0.182331
TS2SA	2	1000	10	1000	11.5	49.2	36	2.9	0.4	593.235701	0.164788
TS2SA	2	1000	20	1000	7.7	45.5	42.6	3.5	0.7	593.999812	0.165
TS2SA	2	1000	30	1000	6	46.8	42.2	4	1	682.994314	0.189721
TS2SA	2	10000	2	1000	41.47	54.51	3.98	0.04	0	6090.806487	1.691891
TS2SA	2	10000	4	1000	16.17	63.98	19.43	0.37	0.05	6273.187259	1.742552
TS2SA	2	10000	6	1000	11.48	59.36	27.93	1.1	0.13	6246.457913	1.735127
TS2SA	2	10000	10	1000	9.54	53.98	33.94	2.14	0.4	6529.467236	1.813741
TS2SA	2	10000	20	1000	8.42	47.72	39.7	3.45	0.71	7375.943426	2.048873
TS2SA	2	10000	30	1000	7.78	45.01	41.79	4.14	1.28	6752.532845	1.875704
TS2SA	2	10000	1 to 6	DSGRN Weighting	10.07	39.96	48.04	1.66	0.27		
TS2SA		DSGRN (196 Essential)			0	23.469388	52.040816	14.285714	10.204082		
TS2SA		DSGRN (756 Essential + Neighbors)			14.814815	39.94709	32.539683	10.05291	2.6455026		
TS2SA		DSGRN (1600 Parameters)			35	40.875	16.375	6.5	1.25		

Table 3. The primary results table with both RACIPE and DSGRN stability percentages.

Conclusions

A major point of comparison between DSGRN and RACIPE is the time cost to run networks. DSGRN runs instantaneously. Meanwhile, as demonstrated in the graph below, the average time to run a model in RACIPE scales linearly with the total number of parameters in the model, as demonstrated in the figure below [3].

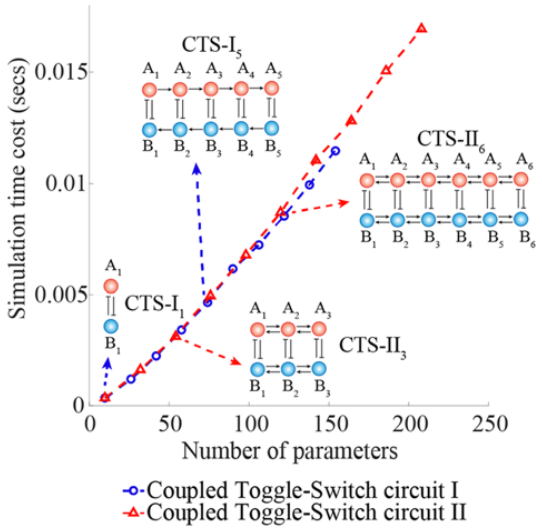

Fig. 7. Graph showing that average time cost to simulate a RACIPE model scales linearly with the number of parameters in the model [3].

The weighting percentages from the table, which were generated in DSGRN with RACIPE data with low Hill coefficients, are close to the RACIPE data for high Hill coefficients. Thus,

weighting in DSGRN with sampling from RACIPE yields comparable results to RACIPE with far quicker computations. Hence, with additional work we hope to show that DSGRN can achieve comparable results to RACIPE for a fraction of the computational cost of RACIPE.

Future Work

The current low-dimensional results bode well for the future. We will begin combining weighting with essential parameters and their neighbors. This entails observing how many RACIPE models were in essential parameters, their neighbors, and parameters that are not essential nor neighbors of essential parameters. The parameter nodes with the highest RACIPE model counts are of particular concern. Additionally, weighting will be done when N = 2, 4, 6, 10, 20, 30 for all three toggle switches. Following the generation of these results, we will observe the conclusions the comparisons yield. Depending on the conclusions, we can take the project in a new direction, such as applying these ideas to networks of biological interest, understanding how to sample for large networks, understanding how to apply these ideas to a broader range of dynamics, computing volumes of the DSGRN regions of parameter space, or understanding how the regions change as a function of more realistic parameters.

Acknowledgments

The author would like to acknowledge The David and Dorothy Bernstein Endowed Scholarship for Summer Research for supporting this research during the summer. The author would also like to thank Konstantin Mischaikow and Marcio Gameiro for being superb research mentors and Prince Rawal for being a great research partner throughout this experience. Finally, the author would like to recognize Lazaros Gallos and Parker Hund for organizing the 2020 DIMACS REU Program.

References

[1] Cummins B, Gedeon T, Harker S, & Mischaikow K. DSGRN: Examining the Dynamics of Families of Logical Models. *Front. Physiol.* 9(549). (2018). https://doi.org/10.3389/fphys.2018.00549

[2] Cummins B, Gedeon T, Harker S, Mischaikow K, & Mok K. Combinatorial Representation of Parameter Space for Switching Networks. *SIAM J. Appl. Dyn. Syst.*, 15(4), pp. 2176–2212. (2016). https://doi.org/10.1137/15M1052743

[3] Huang B, Jia D, Feng J, *et al.* RACIPE: a computational tool for modeling gene regulatory circuits using randomization. *BMC Syst. Biol.* 12(74). (2018). https://doi.org/10.1186/s12918-018-0594-6

[4] Huang B, Lu M, Jia D, Ben-Jacob E, Levine H, & Onuchic J. Interrogating the topological robustness of gene regulatory circuits by randomization. *PLOS Comput. Biol.* 13(3), e1005456–21. (2017). http://doi.org/10.1371/journal.pcbi.1005456

[5] Lu M, Jolly MK, Gomoto R, Huang B, Onuchic J, Ben-Jacob E. Tristability in Cancer-Associated MicroRNA-TF Chimera Toggle Switch. *J Phys Chem B.* 117(42), 13164–13174. (2013). https://doi.org/10.1021/jp403156m