# Markov Chain Computations using Molecular Reactions

*Sayed Ahmad Salehi, Marc D. Riedel, Keshab K. Parhi*

Department of Electrical and Computer Engineering
University of Minnesota
200 Union Street SE
Minneapolis, MN 55455
{saleh022,mriedel,parhi}@umn.edu

***Abstract*— Markov chains are commonly used in numerous signal processing and statistical modeling applications. This paper describes an approach to implement any first-order Markov chain using molecular reactions in general and DNA in particular. Markov chain consists of two parts: a set of states, and state transitions. Each state is modeled by a unique molecular type, referred as a *data* molecule. Each state transition is modeled by a unique molecular type, referred as a *control* molecule, and a unique molecular reaction. Each reaction consumes data molecules of one state and produces data molecules of another state. The concentrations of control molecules are initialized according to the probabilities of corresponding state transitions in the chain. The steady-state probability of Markov chain is computed by equilibrium concentration of *data* molecules. We demonstrate our method for the Gambler's Ruin problem as an instance of the Markov chain process. Both stochastic chemical kinetics and mass-action kinetics validate the computed probabilities using the proposed model. The molecular reactions are then mapped to DNA strand displacement reactions. The error in the probability of ruin computed by the proposed model is shown to be less than 1% for DNA strand displacement reactions.**

***Keywords—molecular computation; Markov chain; Gambler's ruin problem, molecular reaction; DNA strand-displacement***

## I. Introduction

With the advantage of a well-defined theory and extensive simulation software tools, molecular reactions or chemical reaction networks (CRNs) have been used for modeling in different applications. For example, there has been a groundswell of interest in molecular computations in recent years [1-6]. Since 1994, several approaches have been investigated for molecular computation; these include solving: NP-computational and combinatorial problems such as Hamiltonian path problem [1] and finding maximal clique problem [7], computing of deterministic functions and algorithms [8],[9], implementation of logical functions [10]-[14], and signal processing [15].

This paper, for the first time, presents a new methodology for modeling any first-order Markov chain by a set of chemical reactions in order to compute the steady-state probabilities of its states. The produced set of molecular reactions is implemented by DNA strand displacement reactions. Markov chain has been frequently used for modeling and analyzing systems of chemical reactions [4],[16],[17]; However, this paper addresses the reverse problem, i.e., modeling Markov chain and computing its steady-state probabilities by a system of chemical reactions. Since Markov processes are commonly used in numerous processing and statistical modeling applications, a systematic method for synthesizing Markov chains with DNA strand displacement reactions leads to a systematic method for implementing these applications using DNA.

This paper presents a systematic method for implementing first-order Markov chain processes using molecular reactions. Each state in the Markov chain is modeled by a unique *data* molecular type and each state transition is modeled by a molecular reaction and a unique *control* molecule. *Data* molecule for each state or *control* molecule for each state transition is distinguishable from molecules corresponding to other states or state transitions. All the reactions have the form of $C_{ij} + D_i \rightarrow C_{ij} + D_j$, where $C_{ij}$ is the *control* molecule that facilitates transition from state $i$ to $j$ and $D_i$ and $D_j$ are *data* molecules for states $i$ and $j$, respectively. The final concentration of *data* molecules related to each state determines the probability of that state. Since all of the reactions are bimolecular, the model can be mapped to a set of toehold-mediated DNA strand displacement reactions according to prior work [5].

In Section II we briefly review two models for a CRN: stochastic chemical kinetics and mass-action kinetics. A brief review of Markov chain process is presented in Section III. Section IV presents the proposed methodology for modeling Markov chain by molecular reactions. In Section V the proposed model is analyzed using stochastic as well as mass-action kinetics. Section VI explains mapping of the proposed model to DNA strand displacement reactions and finally Section VII concludes the paper.

## II. Modeling by Molecular reactions

This section describes the methodology of constructing a model for Markov chain process using molecular reactions. This model can be used to compute the steady-state probability of each state in the Markov chain diagram. The methodology has two parts: ***initialization*** and ***transition reactions***.

*Initialization:* This stage consists of initializing two groups of molecules: *data* molecules and *control* molecules.

*Data* molecule for each state of Markov chain is a unique type of molecule assigned to that state. The initial quantity for each

This work is supported by the National Science Foundation under grant number CCF-1423407.

data molecule, except the start state, is zero. For the start state the initial value can be any large nonzero number; however, larger the initial value, more accurate the probability estimates are.

*Control* molecules are used to control transformation of data molecules of one state to data molecules of other states according to the transition probabilities in the Markov chain diagram. A unique type of molecule is devoted for each state transition in the chain. The quantities of control molecules are time invariant and can be determined according to the probabilities related to their corresponding transition in the chain; the ratio of quantity of a control molecule over total quantities of all control molecules in a state equals the probability of corresponding transition.

In general, the number of unique molecular types in our model is the sum of the number of states and the number of transitions in the Markov chain.

*Transition Reactions:* The transition reactions determine how data molecules transfer in order to implement the desired Markov chain. There is a transition reaction for each transition in the chain. This reaction transfers data molecules in the source state of transition to the data molecules in the destination state. Each transition reaction uses a control molecule for transferring data molecules. However, transition reactions should not change the concentration of control molecules. Therefore, if a control molecule is used as a reactant in a reaction, it should be also be a product of the reaction.

To illustrate our methodology we explain the molecular model for gambler problem as an instance of Markov chain[21]; a gambler starts with $i$ dollars and plays game of chance in each step, either increasing his money by \$1 or decreasing by \$1. He stops when money is gone, *RUIN*, or when he has *N* dollars, *WIN*. Assuming the chances of winning, $w$, and loosing, $l$, for all states to be identical, what's the probability of ruin?

Fig. 1 shows a 4-state ($N$=3) gambler problem with $w = 0.3$ and $l = 0.7$. Theoretical ruin and win probabilities for this example are 0.886076 and 0.113924, respectively [21].

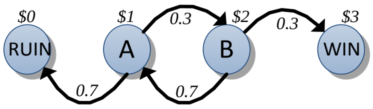


Fig. 1. State diagram for the gambler problem with N=3.

In order to design its molecular reactions first we devote a *data* molecular type to each state: Molecule $RUIN$ for ruin state, $A$ and $B$ for intermediate states, and $WIN$ for win state. Suppose we want to compute P1, i.e., the probability of ruin if gambler starts the game with \$1. In this case, state $A$ is the start state. Therefore, the initial value of its *data* molecule, $A$, is nonzero, while the other states have *data* molecules with zero initial values. We consider 100 as the initial value of $A$.

*Control* molecules $A1$ and $A2$ are assigned to the output transitions of state $A$. Similarly, $B1$ and $B2$ are assigned to the transitions from state $B$. Because $w$=0.3 and $l$=0.7 for this example, we choose initial values as $[A1] = [B1] = 30$ and $[A2] = [B2] = 70$. One should notice that despite the exact concentrations for the control molecules, they need to conform to (1).

$$w = \frac{[A1]}{[A1]+[A2]} = \frac{[B1]}{[B1]+[B2]}$$
$$l = \frac{[A2]}{[A1]+[A2]} = \frac{[B2]}{[B1]+[B2]} \tag{1}$$

The final step is to write the molecular reactions related to each state transition. Reactions (2) and (3) represent output transitions for states $A$ and $B$, respectively. These reactions with the initial concentrations for each molecular type are the proposed molecular model for the gambler problem in Fig. 1.

$$R1\text{:} \quad A + A1 \rightarrow B + A1$$
$$R2\text{:} \quad A + A2 \rightarrow RUIN + A2 \tag{2}$$

$$R3\text{:} \quad B + B1 \rightarrow WIN + B1$$
$$R4\text{:} \quad B + B2 \rightarrow A + B2 \tag{3}$$

Thus, the gambler problem with $N$=3 can be modeled by eight types of molecules and four molecular reactions. Here the transition probabilities for states $A$ and $B$ are similar and control molecules $A1$ and $A2$ can be used for both states and $B1$ and $B2$ can be omitted.

## III. Analysis of the Proposed Molecular Model

According to both stochastic chemical kinetics [18],[19] and mass-action kinetics [20], in this section the proposed molecular model is analyzed. We analyze the molecular model for the 4-state gambler problem shown in Fig. 1.

### A. Stochastic Model

If we only consider state $A$, there are two ways for data molecules $A$ to transfer from this state; they can participate either in reaction $R1$, or $R2$. Based on the stochastic kinetics the probability of participating in reactions $R1$ and $R2$ can be computed as (4) and (5), respectively. Since the quantity of $A1$ and $A2$ are time invariant, the probabilities remain constant.

$$P(R1) = \frac{\binom{a_1}{1}\binom{a}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}} = \frac{a_1}{a_1+a_2} \tag{4}$$

$$P(R2) = \frac{\binom{a_2}{1}\binom{a}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}} = \frac{a_2}{a_1+a_2} \tag{5}$$

If all the states are considered, all of the four reactions can be fired and their probabilities are computed as (6).

$$P(R1) = \frac{\binom{a_1}{1}\binom{a}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}+\binom{b_1}{1}\binom{b}{1}+\binom{b_2}{1}\binom{b}{1}} = \frac{a\,a_1}{a(a_1+a_2)+b(b_1+b_2)}$$
$$P(R2) = \frac{\binom{a_2}{1}\binom{a}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}+\binom{b_1}{1}\binom{b}{1}+\binom{b_2}{1}\binom{b}{1}} = \frac{a\,a_2}{a(a_1+a_2)+b(b_1+b_2)}$$
$$P(R3) = \frac{\binom{b_1}{1}\binom{b}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}+\binom{b_1}{1}\binom{b}{1}+\binom{b_2}{1}\binom{b}{1}} = \frac{b\,b_1}{a(a_1+a_2)+b(b_1+b_2)} \tag{6}$$
$$P(R4) = \frac{\binom{b_2}{1}\binom{b}{1}}{\binom{a_1}{1}\binom{a}{1}+\binom{a_2}{1}\binom{a}{1}+\binom{b_1}{1}\binom{b}{1}+\binom{b_2}{1}\binom{b}{1}} = \frac{b\,b_2}{a(a_1+a_2)+b(b_1+b_2)}$$

For the four probabilities in (6) we assume that at each step at least one reaction can be fired. In other words, $a(a_1 + a_2) + b(b_1 + b_2) \neq 0$. The number of molecules $RUIN$, $A$, $B$, and $WIN$ are considered as the states of the system, $S = (ruin, a, b, win)$. Depending on which reaction is fired, $S$ changes after each step. Fig. 2 shows the graph for the first two steps of the example in Fig. 1. One should keep in mind that the total number of *data* molecules in each state is constant.

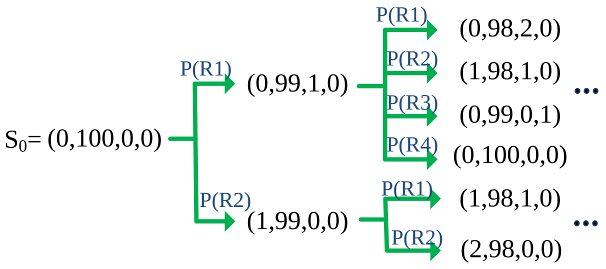


Fig. 2. First two steps of updating the state of molecular model for Fig. 1.

As another interpretation for the model we consider each molecule in the system. The molecule transforms to a molecule either in left state or right state with the probabilities of 0.3 or 0.7, respectively. Therefore, we can interpret each single molecule in the system as an instance of the gambler's play.

The *Monte Carlo* simulation is used for validating the model. The goal is to compute the ruin probability if gambler arrives to play with $1. Therefore, the simulation starts with the initial state $S = (0,100,0,0)$ and stops whenever no more reaction can be fired. The simulation is repeated $10^6$ times. Fig. 3 shows the simulation results. The horizontal axis represents the number of molecules and the blue (red) line represents the number of times the simulation ends up with those numbers of molecules in ruin (win) state. Ruin probability can be calculated as formulated in (7). The mean values of the ruin and win distributions in Fig. 3 are used as the number of molecules. If we simulate with a larger initial value of *data* molecule, the probabilities can be computed more accurately. Table I shows the probabilities obtained using different initial values for *data* molecule *A*. Note that the accuracy improves with increase in the initial value of *A*.

$$P_1 = \frac{number\ of\ data\ molecules\ in\ ruin\ state}{total\ number\ of\ data\ molecules\ in\ ruin\ and\ win\ states} \qquad (7)$$

### B. *Mass-action Kinetics*

Based on the mass-action law, time variation of data molecules can be represented by the ODEs (8)

$$\begin{aligned} \frac{d[A]}{dt} &= -k.[A_1][A] - k.[A_2][A] + k.[B_2][B] \\ \frac{d[B]}{dt} &= -k.[B_1][B] - k.[B_2][B] + k.[A_1][A] \\ \frac{d[S]}{dt} &= k.[A_2][A] \\ \frac{d[E]}{dt} &= k.[B_1][B] \end{aligned} \qquad (8)$$

TABLE I. SIMULATION VS THEORETICAL COMPUTATION OF RUIN PROBABILITY FOR EXAMPLE IN FIG. 1

| Initial value for *A* | Computed ruin probability | Error |
|---|---|---|
| 100 | 0.89 | 0.003 |
| 1000 | 0.887 | 0.0009 |
| 10000 | 0.8862 | 0.0001 |

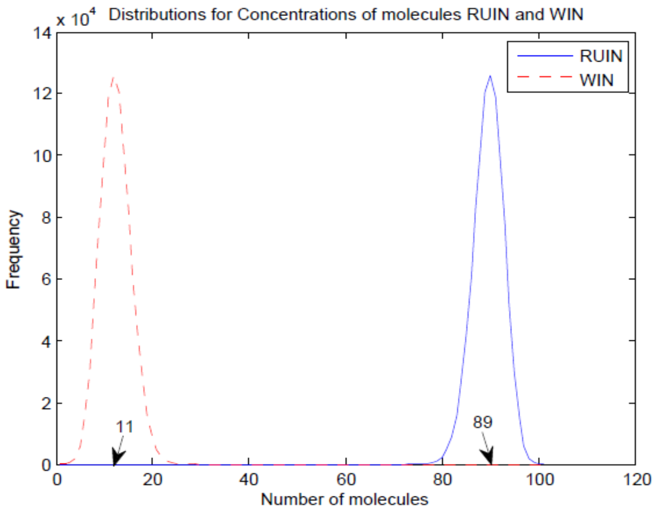


Fig. 3. Stochastic simulation results for molecular model of Fig. 1.

Solving these ODEs using the initial values of molecules, we can obtain the time variation for each molecule. The final concentration of data molecule related to each state can be used to determine the probability of that state.

We used MATLAB to solve the ODEs and plot them as shown in Fig. 4(a). The final concentration for ruin and win molecules are 88.61 (nM) and 11.39 (nM), respectively. Fig. 4 (b) illustrates the ratio $[RUIN]/([RUIN] + [WIN])$ which is the ruin probability and perfectly matches with theoretical value.

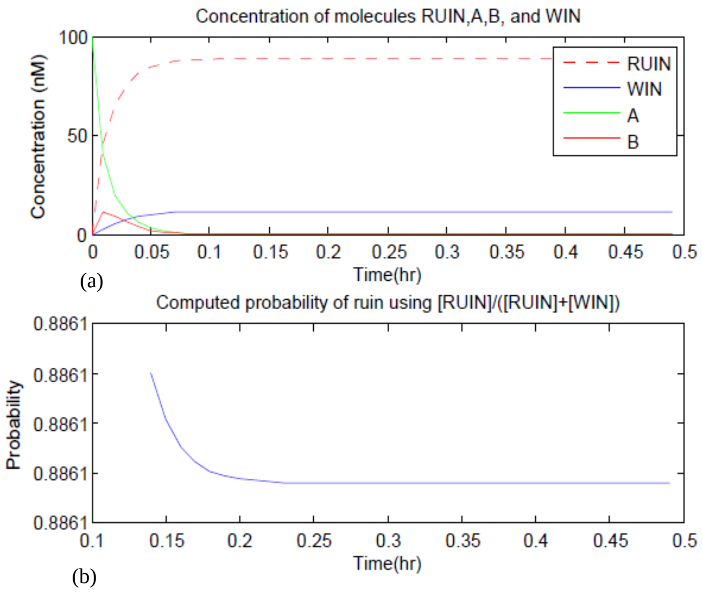


Fig. 4. a) ODE simulation for molecular model of Markov chain in Fig. 1, b) The computed [RUIN]/([RUIN]+[WIN]) ratio .

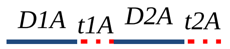


Fig. 5. DNA representation of molecule *A*.

## IV. DNA IMPLEMENTATION

To implement the proposed model with a real molecular system we used DNA strand displacement reactions. As Seelig *et al* [3] show a single strand of a double-strand DNA can be replaced by another single strand, provided a toehold binding is feasible. By properly designing of toeholds in DNA molecules, an arbitrary rate of binding can be achieved. Soloveichik et al [5] have illustrated that DNA strand displacement reactions can be used to implement an arbitrary bimolecular reaction. Our model consists of bimolecular reactions; therefore, it can be implemented by DNA strand displacements using the method presented in [5]. For this purpose each molecule needs to be identified by two toeholds and two domains as depicted in Fig. 5 for molecule $A$. In this representation continuous and dotted lines are used for domain and toehold parts, respectively.

To evaluate the DNA implementation of the proposed model, we implement the model for the example shown in Fig. 1. All the molecules are mapped to the DNA strands as described above. We use the *Mathematica* tool provided by Soloveichik et al [5] to simulate the designed DNA system. The similar initial parameters as [5] are used for simulation. Fig. 6 illustrates the dynamic concentrations of each *data* molecular type. The simulation results match with the simulation results of ODE model as shown in Fig. 4(a). The ruin probability is computed as the ratio of the final concentration of $RUIN$ molecule over the summation of the final concentrations of $RUIN$ and $WIN$ molecules.

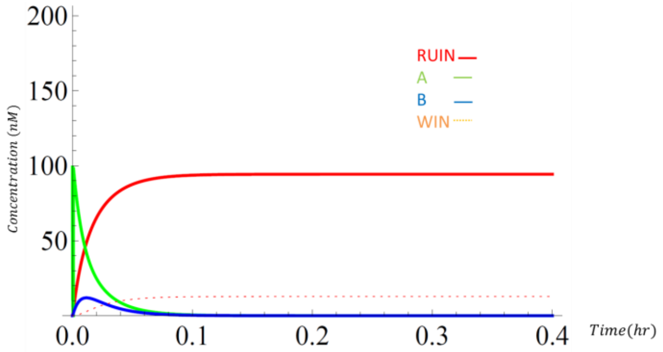
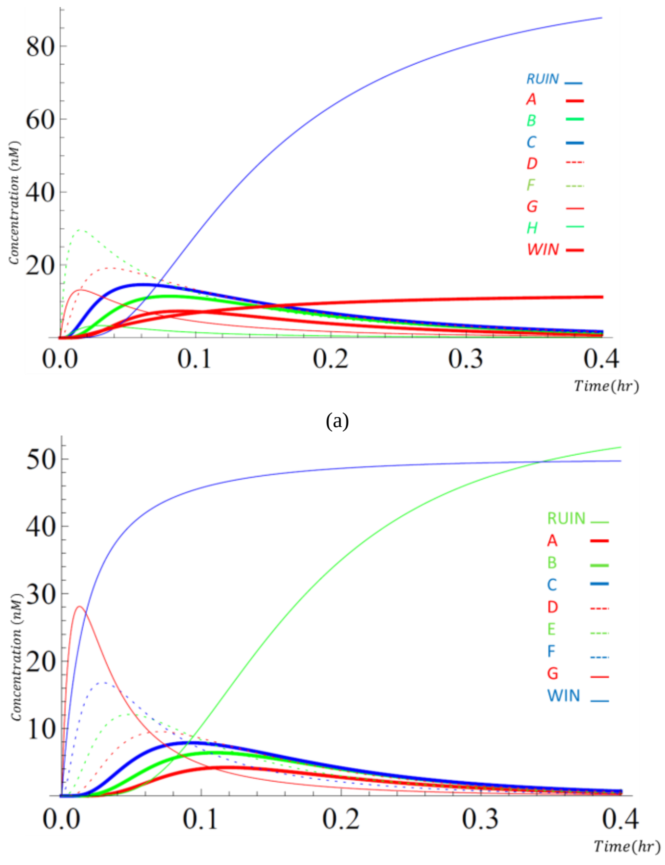


Fig. 6. Simulation results of DNA implementation for the proposed molecular model for Fig. 1.

We next use our DNA construction for a more complex instance of a gambler problem with $N$=9 and similar transition probabilities. We compute ruin probabilities when the gambler starts with \$5 and \$8. For the first case, we initialize the *data* molecule of the 5th state, $E$, to 100nM and the other *data* molecules to zero. While for the second case, we initialize the *data* molecule of the 8th state, $H$, to 100nM and the other *data* molecules to zero. Fig. 7 demonstrates the simulation results. Note that as tabulated in Table II, the ruin probabilities computed using the final concentrations shown in Fig. 7 match with the theoretical probabilities.



(a)

(b)

Fig. 7. Simulation results of the DNA implementation for the gambler problem with N=9 and starting with a) \$5, b) \$8.

TABLE II. SIMULATION VS THEORETICAL COMPUTATION OF RUIN PROBABILITIES FOR A 9-STATE GAMBLER RUIN PROBLEM

| Start state | [ruin]/([ruin]+[win]) | Theoretical probability of ruin |
|---|---|---|
| \$5 | 0.962 | 0.9667 |
| \$8 | 0.569 | 0.5717 |

## V. CONCLUSION

Molecular systems have been used for modeling different applications. This paper demonstrates a method for modeling the stochastic behavior of Markov chain processes using molecular reactions. Both stochastic and ODE simulation results validate our model. Although we describe the modeling of a gambler ruin problem, a first-order Markov chain with identical transition probabilities in each state, the method can be used for modeling any Markov chain process. A first-order Markov process with different transition probabilities for each state can be easily modeled by adjusting the initial quantities for control molecules of each state. Future work will be directed towards modeling of higher order Markov processes and generalizing the method for different types of random processes.

## REFERENCES

[1] L. Adleman, “Molecular computation of solutions to combinatorial problems,” *Science*, vol. 266, no. 11, pp. 1021–1024, 1994.

[2] L. Qian, D. Soloveichik, and E. Winfree, “Efficient turing-universal computation with DNA polymers,” in International Conference on DNA Computing and Molecular Programming, 2010.

[3] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree, “Enzyme-free nucleic acid logic circuits,” *Science*, vol. 314, 2006, pp. 1585–1588.

[4] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck, “Computation with finite stochastic chemical reaction networks,” Natural Computing, vol. 7, no. 4, 2008.

[5] D. Soloveichik, G. Seelig, and E. Winfree, “DNA as a universal substrate for chemical kinetics,” Proceedings of the National Academy of Sciences, vol. 107, no. 12, pp. 5393–5398, 2010.

[6] B. Yurke, A. J. Turberfield, A. P. Mills, Jr, F. C. Simmel, and J. Neumann, “A DNA-fuelled molecular machine made of DNA,” *Nature*, vol. 406, pp. 605–608, 2000.

[7] Q. Ouyang, P. D. Kaplan, S. Liu and A. Libchaber, “DNA solution of the maximal clique problem,” *Science* 278, 5337, 446-449, 1997.

[8] F. Guarnieri,M. Fliss, and C. Bancroft, “Making DNA add,” *Science*, vol. 273, 220–223, 1996.

[9] H.L. Chen, D. Doty, and D. Soloveichik, “Deterministic function computation with chemical reaction networks,” *DNA 2012, Lecture Notes in Computer Science*, vol. 7433, Springer-Verlag, 2012, pp. 25-42.

[10] J. C. Anderson, C. A. Voigt, and A. P. Arkin “A genetic AND gate based on translation control,” *Molecular Systems Biology*, 3(133), 2007.

[11] K. Ramalingam, J. R. Tomshine, J. A. Maynard, and Y. N. Kaznessis, “Forward engineering of synthetic biological AND gates,” *Biochem. Eng. J.* 47, pp.38−47, 2009.

[12] L. Qian, E. Winfree, “Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades,” *Science*, pp. 1196-1201, 2011.

[13] A. Tamsir, J. J. Tabor, and C. A. Voigt, “Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’,” *Nature* 469, pp. 212−215, 2011.

[14] J. Bonnet, P. Yin, M. E. Ortiz, P. Subsoontorn, and D. Endy, "Amplifying Genetic Logic Gates," *Science*, pp. 599-603, 2013.

[15] H. Jiang, S. Salehi, M. Riedel, and K. Parhi, “Discrete-time signal processing with DNA,” *ACS Synthetic Biology*, vol. 2, no. 5, pp. 245–254, 2013.

[16] M. Gibson and J. Bruck, “Efficient exact stochastic simulation of chemical systems with many species and many channels,” *Journal of Physical Chemistry A*, 104:1876–1889, 2000.

[17] D.T. Gillespie, “A General Method for Numerically Simulating the stochastic time evolution of coupled chemical reactions,” *The Journal of Computational Physics*, 22 :403–434, 1976.

[18] B. Lewin, Genes VII. Oxford University Press, 1999.

[19] D. T. Gillespie, “Stochastic simulation of chemical kinetics,” Annual Review of Physical Chemistry, 58:35–55, 2007.

[20] J. E. House: Principles of chemical kinetics, Academic Press, 2007

[21] S. Resnick, Adventures in stochastic processes, Birkhauser, 4th printing, 2005