An improved stochastic modelling framework for biological networks

I. Altarawni [a], S. Samarasinghe [a] and D. Kulasiri [b]

[a] Faculty of Environment, Society and design, Lincoln University, [b] Faculty of Agriculture and life sciences, Lincoln University
Email: Sandhya.Samarasinghe.lincolnuni.ac.nz

Abstract: It has become very clear that stochasticity in biology is a rule rather than exception. Gillespie stochastic simulation algorithm (GSSA) (direct method) is the first algorithm proposed to model stochasticity in biochemical systems. However, the computational intractability of direct method has been identified as the main challenge for using it to model large biochemical systems. In this paper, a novel variant of the GSSA is proposed to address computational intractability of the direct method. The direct method is combined with a Mapping Reduction Method (MRM) to target a single run of the direct method to be accelerated by advancing the system through several reactions at each time step to replace the single reaction in GSSA. MRM is a framework for mimicking parallel processes occurring in large systems using a large number of threads that work together and seen as a single system. It is used for parallel problems to be processed across large datasets using a large number of nodes working together as a single system. Link between GSk3 and p53 in Alzheimer's disease (AD) is modelled using the proposed method and tested and validated by comparing it with the direct method.

The framework of GSSA/MRM includes four steps. These steps are initialization, election (mapping), selection (reduction) and updating. As shown in Figure 1. Initialization step is used to create a thread pool that includes T threads (reactions) and initialize the system by calculating the propensity function (a_j) for each reaction. Election step is mainly used to elect the number of threads equal to the number of reactions that have $a_j > 0$ to run GSSA. Each thread that runs GSSA is able to determine the next reaction j to occur and its time step τ. All reactions that are returned from the election step are filtered and only reactions that are able to fire are selected. GSSA/MRM is equal to GSSA if only one reaction is selected. If two reactions are selected and to reduce the number of time steps as GSSA does, the time step τ is the sum of the time steps from both threads. If more than two reactions are selected, the time step is calculated as the sum of the largest 3τ. Then, t is updated and the number of molecules is updated. The simulation is repeated until all possible reactions have been fired or the time of simulation is exceeded. This paper shows that GSSA/MRM is faster than GSSA due to the possibility of firing more than one reaction at each time step.

Keywords: *GSSA, MRM, Alzheimer's disease, p53, GSk3*

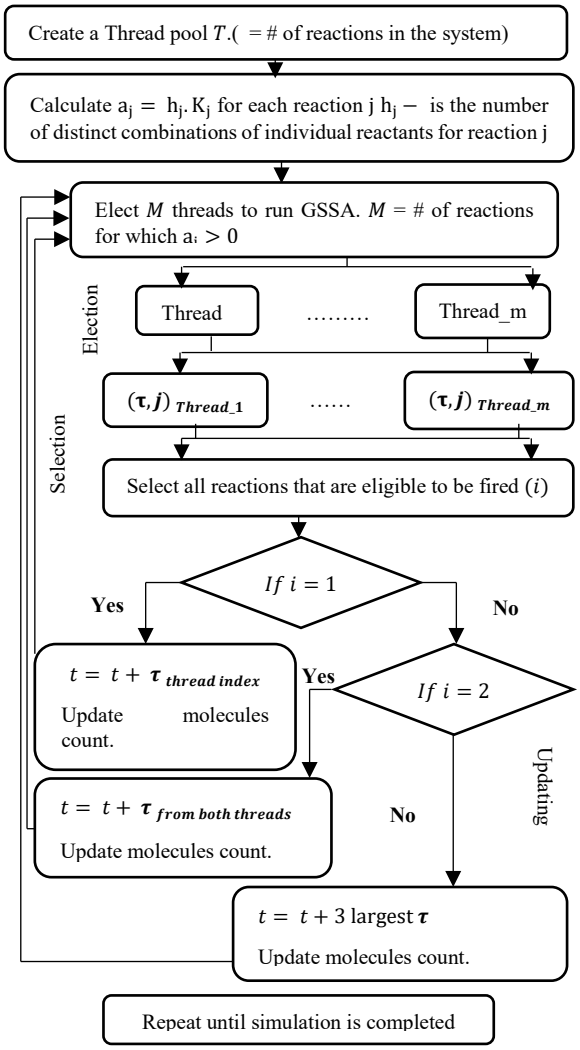

Figure 1. Schematic of GSSA/MRM

1. INTRODUCTION

It has become very clear that noise in biology is a rule rather than exception (Sauer 2012). Therefore, stochastic models have received a great deal of attention recently leading to many recent reviews (Raser and O'shea 2005, Gillespie 2007, Češka, Šafránek et al. 2014, Burrage, Burrage et al. 2017, Pischel, Sundmacher et al. 2017). Stochastic models are widely used to model biological systems classified as small systems (<100 molecules for each species in a given system). Direct method, also known as the Gillespie Stochastic Simulation Algorithm (GSSM), is the first algorithm used to stochastically model biochemical system. In each time step, the direct method uses the current state of the system and determines which reaction will occur next and when it will occur (Gillespie 1977).

However, the computational intractability of direct method has been identified as the main challenge for using it to model large biochemical systems. Different extensions of the direct method have been proposed to cope with its computational intractability. These extensions are: (1) the first reaction method (Gillespie 1977); (2) the next reaction method (Gibson and Bruck 2000); (3) the optimized direct method (Cao, Li et al. 2004); (4) the sorted direct method (McCollum, Peterson et al. 2006); (5) the logarithmic direct method (Madani, Poirier et al. 2006); and (6) the tau-leap modified Poisson method (Cao, Gillespie et al. 2006).

2. GILLESPIE STOCHASTIC SIMULATION ALGORITHIM GSSA (DIRECT METHOD)

The direct method is a well-known technique used to stochastically model biochemical reactions and it is roughly equivalent to the Chemical Master Equation (CME). The CME is an exact method that is used to enumerate all possible states for any stochastic system at any given time by tracking the behaviour of the system (Gillespie 1992).

Using GSSA, a PDF (probability density function) can be obtained from an infinite number of simulations and this PDF is identical to the true distribution of the system, as given by the CME (Haugh 2004). However, an identical PDF to the true distribution is never reached but an accurate PDF that depends on the system or type of application could be achieved using a high number of repeats of the GSSA (Gillespie 2007). The GSSA is used to generate a step-by-step trajectory of the system instead of following the time evolution of the probabilities of the CME. In each time step, the GSSA uses the current state of the system and determines which reaction will occur next and when it will occur. Assume a system involves N molecular species $(S_1, \dots\dots\dots\dots, S_N)$; that are represented by $X(t) = (X_1(t), \dots\dots\dots\dots, X_N(t))$ (the state vector), where $X_i(t)$ is the number of molecules of S_i at time t; and M reactions channels $(R_1, \dots\dots\dots\dots, R_m)$. The GSSA steps along in time reaction-by-reaction, governed by the reaction probability (a_j) (propensity function) and by the state change vector $v_j = (v_{1j}, \dots., v_N)$. $a_j(x)dt$ gives the probability that one reaction will occur in the next time step. The steps of the direct method are summarized in Figure 2.

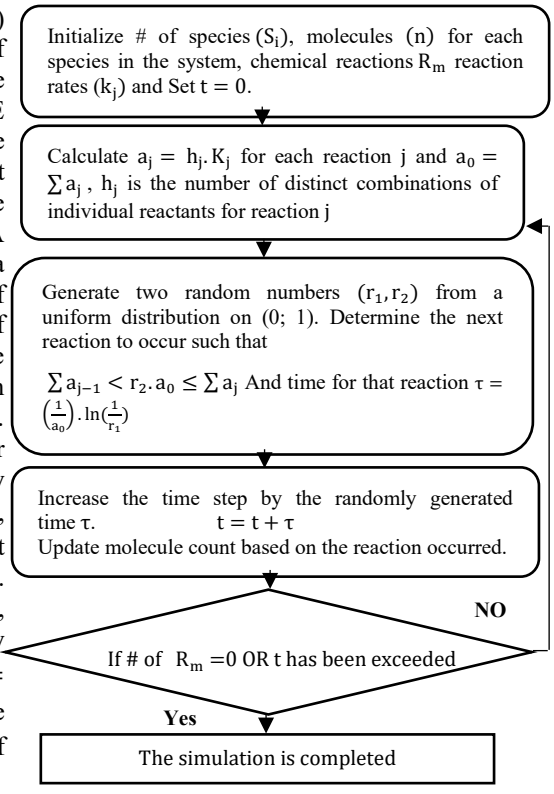

Figure 2. Schematic of the direct method

3. GILLESPIE STOCHASTIC SIMULATION ALGORITHIM/ MAPPING REDUCTION METHOD (GSSA/MRM)

3.1 Mapping Reduction Method (MRM)

The proposed approach, MRM, is a method for processing large data sets on a single multi-processor computer (using threads or processors) (Dean and Ghemawat 2008), a cluster (Barroso, Dean et al. 2003), or a grid (Bent, Thain et al. 2004). The MRM is also defined as a framework that is used for parallel problems to be processed across large datasets using a large number of nodes working together and seen as a single

system. Each node performs the same task and is controlled and scheduled by software (Dean and Ghemawat 2008, Lämmel 2008). A single multi-processor computer is able to employ multiple threads or processors to work in a parallel manner on the same machine (Dean and Ghemawat 2010). A computer cluster is composed of a set of loosely, or tightly, connected computers on the same local network and using the same hardware. A computer grid is also a set of connected computers but these computers are not only shared over geographically distributed systems, but also use heterogeneous hardware (Mann, Trasatti et al. 2003).

Implementing MRM using a single machine is less complex than using a cluster or a grid because the input data is split only among worker threads that all reside on the same machine and typically use the same data store (Lattanzi, Moseley et al. 2011). Additional complexity is added into the process when multiple computers are used to run MRM because the input data have to be split among all computers within the cluster using a master node (McKenna, Hanna et al. 2010). Another challenge for using a cluster is that different physical memories on different machines have to be used to save data from the reduction method (Lv, Hu et al. 2010). A cluster is needed to implement MRM especially when the input and output data are too large to fit into the memory of a single computer (Ferreira Cordeiro, Traina Junior et al. 2011).

3.2 GSSA/MRM

The GSSA and its variants advance the state of the system under study by executing one reaction at a time. In cases where the system involves a large number of reactions, its simulation with these methods becomes prohibitively expensive. Here, we propose a novel variant of the direct method of GSSA to address its computational intractability by using MRM on a single multiprocessor computer to advance the system by several reactions. Specifically, a single run is targeted to be accelerated by advancing the system through several reactions in each time step. MRM/GSSA is divided into four steps. These steps are initialization, election (mapping), selection (reduction), and updating the system as shown in Figure 1. The pseudo code of these steps is summarized in Table 1.

4. CASE STUDY: LINK BETWEEN GSK3 AND P53 IN ALZHEIMER' DISEASE

Alzheimer's disease (AD) is mainly characterized by the presence of two proteins and their aggregation relationship. These proteins are amyloid-beta (Aβ) and micro-tubular binding protein (tau) accompanied by glial cell activation (Nicoll, Wilkinson et al. 2003, Nicoll, Barton et al. 2006, Boche, Denham et al. 2010, Maarouf, Daugs et al. 2010, Zotova, Holmes et al. 2011).

Recently it has been suggested that glycogen synthase kinase-3β (GSK3β) is implicated in familial forms of AD. An increase in tau hyper-phosphorylation is indirectly caused by p53. Evidence has also suggested that GSK3β and p53 interact and this interaction has the responsibility to increase the activity of both proteins. Under normal cellular conditions as shown in Figure 3 (A), the level of p53 is kept low due to the binding with Mdm2 that targets p53 for proteasomal degradation. When cells are stressed, p53 is stabilized and may then interact with GSK3β. The interaction between p53 and GSK3β is suggested to be an important contributor to cellular outcomes (Proctor and Gray 2010). Proctor and Gray, (2010) proposed a stochastic simulation model to test this hypothesis. The stochastic model demonstrates that an increase in not only levels of Aβ plaques, but also levels of tau tangles is caused by increasing the activity of GSK3β. Therefore, Proctor and Gray (2010) in their model focused on the link between p53 and GSK3β and they suggested that modulating this interaction could be a useful therapeutic strategy.

5. RESULTS FROM TESTING AND VALIDATION OF GSSA/MRM

In order to not only test, but also verify the quality of MRM/GSSA, it is used to model the link between GSk3 and p53 in AD. We show that GSSA/MRM is a useful way to model biochemical systems when the number of reactions with propensity functions (a_j) greater than zero is quite large. This is because GSSA/MRM employs a large number of threads to run GSSA. Thus, the chance for multiple reactions to be eligible for the selection step is high. Therefore, the main difference between GSSA and GSSA/MRM is that GSSA advances the state of the system by executing one reaction at a time while MRM/GSSA advances the system state by several reactions within a calculated time step, τ. Therefore, it is mainly used to accelerate a single run of GSSA and explicitly include the concurrency feature. GSSA/MRM is compared with GSSA from three angles – results, performance in term of CPU time and representation of stochasticity.

Table 1. The pseudo code of GSSA/MRM

A- Initialization step

1- Create a thread pool that contains T number of threads,
$\boldsymbol{T = \#\ reactions\ in\ the\ system}$
The number of threads in the thread pool is set to be equal to the number of reactions for two reasons.
- All reactions in the system might have the ability to fire together.
- Creating and terminating threads as needed is an expensive process in terms of time.

2- Initialize the biochemical system
3- Create arrays Election_index_array and Election_time_step_array to store the indexes and time steps, respectively that are returned from threads.
4- Create Selection_index_array and Selection_time_step_array to store the indexes and time steps, respectively, after the selection step.

B- Election step

1. Elect M threads to run GSSA
$\boldsymbol{M = \#\ of\ reactions\ that\ have\ a_j > 0}$
2. Each thread returns an index and a time step.
3. Store the index of the next M reactions to occur in the Election_index_array and M time steps in the Election_time_step_array.

C- Selection step

❖ Test the eligibility of each reaction as follows:
If all reaction indexes in the Election_index_array are different
Function 1()
{
❖ Check reactions that needs same molecules (F reactions)
If molecules are enough for F reactions to be executed.
❖ All reactions in the Election_index_array are eligible for the selection step.
❖ Transfer time steps for all different indexes from the Election_time_step_array to the Selection_time_step_array
Else
❖ Depending on the number of molecules, # reactions X reactions that could be executed are determined and reactions with maximum are selected.
❖ X reactions are eligible for the selection step.
❖ Transfer time steps for X reactions Election_time_step_array to the Selection_time_step_array
}
- If some reaction indexes are the same (for example, L threads return the same index, j):
❖ Call Function 1() for all different reactions
For same reactions
▪ Check the number of molecules for each species in reaction j
▪ If the number of molecules for each participant species in reaction j is enough to run reaction j L times, reaction j is eligible for the selection step L times. So,
❖ Transfer the L reactions from the Election_index_array to the Selection_index_array
❖ Transfer the L time steps for reaction j from Election_time_step_array to the Selection_time_step_array
▪ Else
✓ Set x= the number of molecules for a species in reaction j that has the smallest number of molecules.
✓ Number of eligible j for the selection step = $L - x$
✓ Select indexes of j that have a large time step.
✓ Update the Selection_index_array.
✓ Update the Selection_time_step_array.

D- Updating the system

❖ Update the number of molecules
Update the molecules in the number of species in all the next reactions to occur in the Selection_index_array.
❖ Update the time of the system
If Selection_time_step_array length =1
$t = t + Selection\_time\_step\_array[0]$
If Selection_time_step_array length =2
$t = t + Selection\_time\_step\_array[0] + Selection\_time\_step\_array[1]$
If Selection_time_step_array length > 2
❖ $t = t +$ The largest three time steps in the Selection_time_step_array

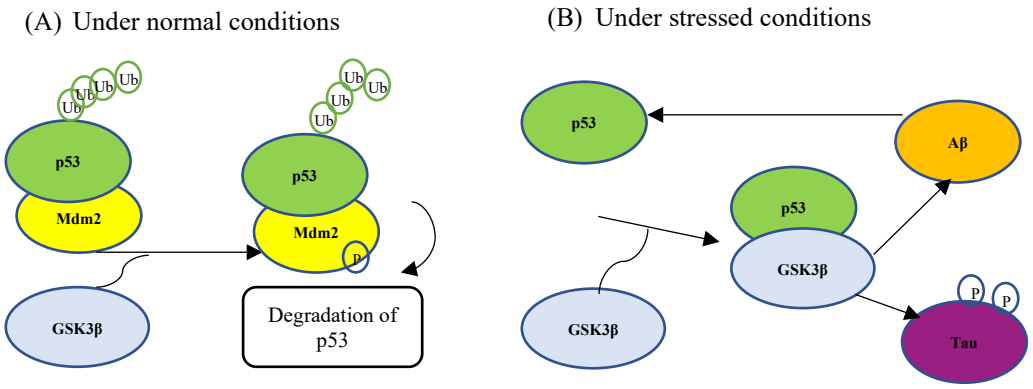

Figure 3. GSK3/p53 hypothesis for AD. (A) Binding relationship between p53 and Mdm2 under normal conditions targets p53 for proteasomal degradation. (B) Under stressed conditions, p53 is stabilized and it forms complex with GSK3β. This not only increases the production of Aβ, but also hyper phosphorylates tau.

5.1 Results

Figures 4 and 5, respectively, demonstrate the behaviour of p53 from four runs of GSSA and GSSA/MRM. GSSA/MRM shows good representation of the behavior of p53 comparable to GSSA even though it advances the system by several reactions.

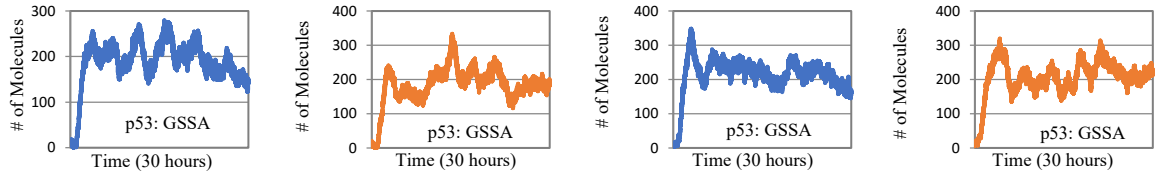

Figure 4. The behaviour of p53 from four runs of GSSA.

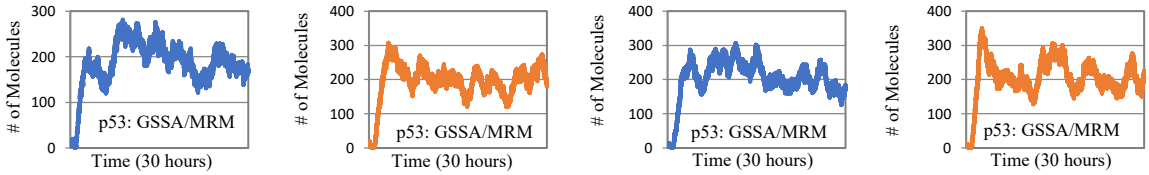

Figure 5. The behaviour of p53 from four runs of GSSA/MRM. GSSA/MRM shows good representation of the behavior of p53 comparable to GSSA even though it advances the system by several reactions.

5.2 Representation of stochasticity

MRM/GSSA showed ability to represent the stochasticity feature comparable to GSSA. Figure 6 shows the average behaviour of p53 from 200 runs of GSSA (red line), MRM/GSSA (blue line). At a random point (the vertical line in the Figure), the mean value of p53 from GSSA and GSSA/MRM are 213.5133, 178.0093. To check how each approach represents stochasticity, the standard deviation (σ) was calculated at that random point to assess how p53 values are spread around the mean μ for both approaches. GSSA/MRM is comparable to GSSA in terms of capturing a high level of stochasticity as indicated by their respective standard deviations of 59.9 and 62.3.

5.3 Performance (CPU time)

In MRM/GSSA, all reaction channels R_j with $a_j(x) > 0$ that are eligible for the selected step are saved in a list. The system state is advanced by executing all reactions in the list where each reaction is executed just once. GSSA advances the system by only one reaction at each time step while GSSA/MRM advances the system by several reactions. Therefore, it is expected that GSSA/MRM will be more time-efficient. GSSA and GSSA/MRM were used to run the AD model involving the relationship between GSk3 and p53 to produce just one realization of the system 10 times. As shown in Table 2, GSSA/MRM is much faster than

GSSA for all 10 runs. It is clearly seen that GSSA/MRM takes less than half the time required by GSSA. Thus, GSSA/MRM shows good performance in term of processing time compared to GSSA.

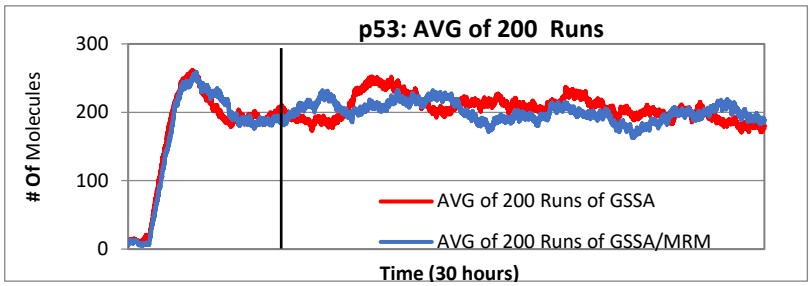

Figure 6. Average of 200 runs for p53 from GSSA and GSSA/MRM. GSSA advances the system by only one reaction at each time step while GSSA/MRM advances the system by several reactions. GSSA/MRM method shows good representation of the behaviour of p53 comparable to GSSA although it advances the system by several reactions. At a random point (the vertical line), we compare GSSA/MRM in terms of stochasticity with GSSA and results revealed a close agreement.

Table 2. CPU time for GSSA and GSSA/MRM. The average CPU time for MRM/GSSA is less than half that for the GSSA

#	GSSA	MRM/GSSA
1	9m.23s.233ms	3m.44s.212ms
2	8m.58s.723ms	4m.31s.777ms
3	9m.02s.641ms	2m.59s.854ms
4	9m.41s.234ms	3m.45s.12ms
5	7m.32s.621ms	4m.43s.13ms
6	7m.1s.223ms	3m.04s.04ms
7	8m.22s.431ms	2m.56s.821ms
8	7m.32s.143ms	3m.34s.523ms
9	9m.43s.721ms	3m.04s.241ms
10	8m.22s.245ms	5m.10s.221ms
AVG	8m.27s.403ms	3m.33s.367ms

6. SUMMARY

In summary, our proposed method produces the behavior of a biochemical system comparable to GSSA in terms of accuracy of representation and stochasticity. Importantly, as expected, GSSA/MRM takes less than half the time required by GSSA. Therefore, GSSA/MRM is able to replace GSSA; it is particularly beneficial when the biochemical system contains a very large number of reactions.

7. FUTURE DIRECTION

Modelling a large biochemical system (immunization in AD (Proctor, Boche et al. 2013)) using GSSA/MRM is the first future direction of this research. The second direction of this research is testing and validating GSSA/MRM results, performance, representation of stochasticity and reliability by comparing it with not only GSSA, but also the modified tau leap method classified to be one of the fastest versions of GSSA. Analysis more in detail to connect parallelism in the propose approach to parallelism in nature.

REFERENCES

Barroso, L. A., et al. (2003). Web search for a planet: The Google cluster architecture. IEEE micro 23(2): 22-28.

Bent, J., et al. (2004). Explicit Control in the Batch-Aware Distributed File System. *NSDI*.

Boche, D., et al. (2010). Neuropathology after active Aβ42 immunotherapy: implications for

Alzheimer's disease pathogenesis. *Acta neuropathologica* 120(3): 369-384.
Burrage, K., et al. (2017). A review of stochastic and delay simulation approaches in both time and space in computational cell biology. *Stochastic Processes, Multiscale Modeling, and Numerical Methods for Computational Cellular Biology*, Springer**:** 241-261.
Cao, Y., et al. (2006). Efficient step size selection for the tau-leaping simulation method. *The journal of chemical physics* 124(4): 044109.
Cao, Y., et al. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics* 121(9): 4059-4067.
Češka, M., et al. (2014). Robustness analysis of stochastic biochemical systems. *PloS one* 9(4): e94553.
Dean, J. and S. Ghemawat (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM* 51(1): 107-113.
Dean, J. and S. Ghemawat (2010). MapReduce: a flexible data processing tool. *Communications of the ACM* 53(1): 72-77.
Ferreira Cordeiro, R. L., et al. (2011). Clustering very large multi-dimensional datasets with mapreduce. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.*
Gibson, M. A. and J. Bruck (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A* **104**(9): 1876-1889.
Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry* 81(25): 2340-2361.
Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications* 188(1): 404-425.
Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.* 58: 35-55.
Haugh, M. (2004). Generating random variables and stochastic processes.*Monte Carlo Simulation: IEOR EA703*.
Lämmel, R. (2008). Google's MapReduce programming model—Revisited. *Science of computer programming* 70(1): 1-30.
Lattanzi, S., et al. (2011). Filtering: a method for solving graph problems in mapreduce. *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures, ACM*.
Lv, Z., et al. (2010). Parallel k-means clustering of remote sensing images based on mapreduce. *Web Information Systems and Mining*: 162-170.
Maarouf, C. L., et al. (2010). The biochemical aftermath of anti-amyloid immunotherapy. *Molecular neurodegeneration* 5(1): 39.
Madani, R., et al. (2006). Lack of neprilysin suffices to generate murine amyloid-like deposits in the brain and behavioral deficit in vivo. *Journal of neuroscience research* 84(8): 1871-1878.
Mann, B. E., et al. (2003). Loosely coupled mass storage computer cluster, Google Patents.
McCollum, J. M., et al. (2006). The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational biology and chemistry* 30(1): 39-49.
McKenna, A., et al. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research* 20(9): 1297-1303.
Nicoll, J. A., et al. (2006). Aβ species removal after Aβ42 immunization. *Journal of Neuropathology & Experimental Neurology* 65(11): 1040-1048.
Nicoll, J. A., et al. (2003). Neuropathology of human Alzheimer disease after immunization with amyloid-β peptide: a case report. Nature medicine 9(4): 448-452.
Pischel, D., et al. (2017). Efficient simulation of intrinsic, extrinsic and external noise in biochemical systems. *Bioinformatics* 33(14): i319-i324.
Proctor, C. J., et al. (2013). Investigating interventions in alzheimer's disease with computer simulation models. *PloS one* 8(9): e73631.
Proctor, C. J. and D. A. Gray (2010). GSK3 and p53-is there a link in Alzheimer's disease?. *Molecular neurodegeneration* 5(1): 7.
Raser, J. M. and E. K. O'shea (2005). Noise in gene expression: origins, consequences, and control. *Science* 309(5743): 2010-2013.
Sauer, T. (2012). Numerical solution of stochastic differential equations in finance. *Handbook of computational finance*, Springer**:** 529-550.
Zotova, E., et al. (2011). Microglial alterations in human Alzheimer's disease following Aβ42 immunization. *Neuropathology and applied neurobiology* 37(5): 513-524.