Summer 2019

Multiscale spatio-temporal modeling of cell population in tissue architecture and drug delivery nanoparticles

Mohammad Aminul Islam

MULTISCALE SPATIO-TEMPORAL MODELING OF CELL POPULATION IN TISSUE ARCHITECTURE AND DRUG DELIVERY NANOPARTICLES

by

MOHAMMAD AMINUL ISLAM

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

CHEMICAL ENGINEERING

2019

Approved by

Dipak Barua, Advisor
Sutapa Barua
Jee-Ching Wang
Joontaek Park
Sajal K. Das

ABSTRACT

Multiscale nature of a biological system span at many order of magnitudes in time and space. Molecular interaction at lower scale is connected with the higher scale behavior of tissue or organism. Integrating the dynamics and information at different time and space can give a fundamental physiological understanding of the higher level phenomena. But complex features, functions, interconnectivity between different scales and lack of information on the fundamental physiological property make the model difficult and computationally challenging. The multiscale modeling approach can bridge the gap between different scale by a systemic integration of the complex dynamic behavior.

Here, the focus is on developing multiscale modeling approaches to study the dynamic behavior of tissue. First, a multiscale spatiotemporal model is designed to investigate the tissue scale dispersion and penetration of nanoparticles from lower scale particle-cell interaction. The results obtained suggest that the size of nanoparticles may play less significant roles in tissue scale penetration and dispersion. The effect of nanoparticle size is less prominent due to the presence of particle-cell interaction and advection. This scalable spatiotemporal model can simulate the dynamics of drug delivery particles in the extracellular domain of a tissue.

Furthermore, a parallel framework is developed to study the collective behavior of the cell population in a tissue architecture from their intracellular and extracellular reaction kinetics. The framework can model population dynamics at the tissue scale from a single cell biochemical reaction network accurately and efficiently. Finally, the framework's capability is demonstrated by simulating a full-scale model of bacterial quorum sensing, where the dynamics of a population of bacterial cells is dictated by the intercellular communications in a time-evolving growth environment.

ACKNOWLEDGMENTS

First, I am indebted to Dr. Dipak Barua, my advisor, in a great measure, for teaching me the research in the field of computational and system biology. I sincerely thank him for his insightful thoughts, idea, encouragement, patience, advice, incisive feedback and valuable suggestions, throughout the journey. Without him, this dissertation would not be possible.

Second, I would like to express my gratitude to Dr. Sutapa Barua, Dr. Jee-Ching Wang, Dr. Joontaek Park, and Dr. Sajal K. Das, my committee members, for their continuing support, timely feedback, and advice during my research project. I sincerely thank Dr. Sutapa Barua for her advice and help in the design of my experiments. I thank Dr. Sajal K. Das for his insights and suggestions in computations of my projects. Moreover, I am expressing my gratitude to Satyaki Roy, a Ph.D. student in Missouri S&T and my collaborator from the CSE department for his support and contribution in the projects.

Third, I would like to thank the faculty, staff, and friends from Chemical and Biochemical Engineering Department at Missouri S&T. I appreciate all of my other friends from Bangladesh and other countries for their support. I also thank the funding agencies for their support to make this work possible.

I am eternally indebted to my mom and dad for their unconditional love and support. Their contribution to my life remained as the pillar of this work. I am also grateful to my brothers and sisters for their continuous support and motivation. Finally, I would love to thank my beautiful wife, Simu, for her unconditional love, support, and understanding.

All of you have made my academic pursuit more enjoyable and rewarding.

I dedicate this dissertation to my mother for her constant support, prayers, and unconditional love.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

NOMENCLATURE

Symbol	Description
A	total propensity
D	diffusion constant
K_B	Boltzmann constant
M	molecular concentration matrix
P	Pressure
R_c	reaction rate constant
R_r	reaction rules
T	simulation time
T_p	Temperature
Z_i	cell process
Γ	interstitial space
Λ	Area occupied by cells
Ω	System boundary of 2D tissue architecture
Ψ	interval between Master slave communication
β_i	division time
$\boldsymbol{f}$	force vector
$\boldsymbol{u}(\boldsymbol{x})$	local fluid velocity vector

$\boldsymbol{x}$	position vector
$\delta(\boldsymbol{x})$	Dirac delta function
γ	reaction set
μ	fluid viscosity
ω	reactant concentration
$\phi_\epsilon(\boldsymbol{x})$	cutoff function
ρ	capture probability
MAE	Mean Absolute Error

1. INTRODUCTION

1.1. BACKGROUND AND LITERATURE SURVEY

Multiscale nature is the inherent property for any biological system. The physiological functions and dynamics of a biological system occur at a different magnitude of time and space. Time required to occur any physiological phenomena (e.g.: molecular interaction or cellular death) span from nanoseconds ($10^{-}9s$) to years ($10^{8}s$) while the space is span from molecular scale ($10^{-}10m$) to organism scale ($1m$) (Figure 1.1) [1]. The dynamics of a biological system at a lower scale is stochastic due to randomness and fluctuation in a short time scale. On the other hand, the dynamics at higher scales are more deterministic due to a large number of species and a longer time scale. It is challenging to integrate the dynamics and information at different scales due to the interconnectivity and hierarchical structure of a biological system. Interactivity can occur at the same scale (e.g., intracellular protein-protein interaction) or across different scale (e.g., interactions between ligand and receptor expressing a gene to cause cellular death). So, it is difficult to experimentally observe the phenomena at different time and space at the same time as well as to develop a mathematical model to study the complex dynamic behavior.

The biological system often required hypothesis and experimentation to connect the link between the intracellular molecular interaction scale to the scale of cell population behavior and beyond [2]. Multiscale modeling approach can simulate the multiscale nature of a biological system quantitatively. The primary purpose of multiscale modeling is not only to model the system at multiple scales but also to conserve the information and dynamics between different scales. So, the model at higher scale will preserve the physiological data of lower magnitude and the model at a lower level will conserve the phenomenological information at a higher scale.

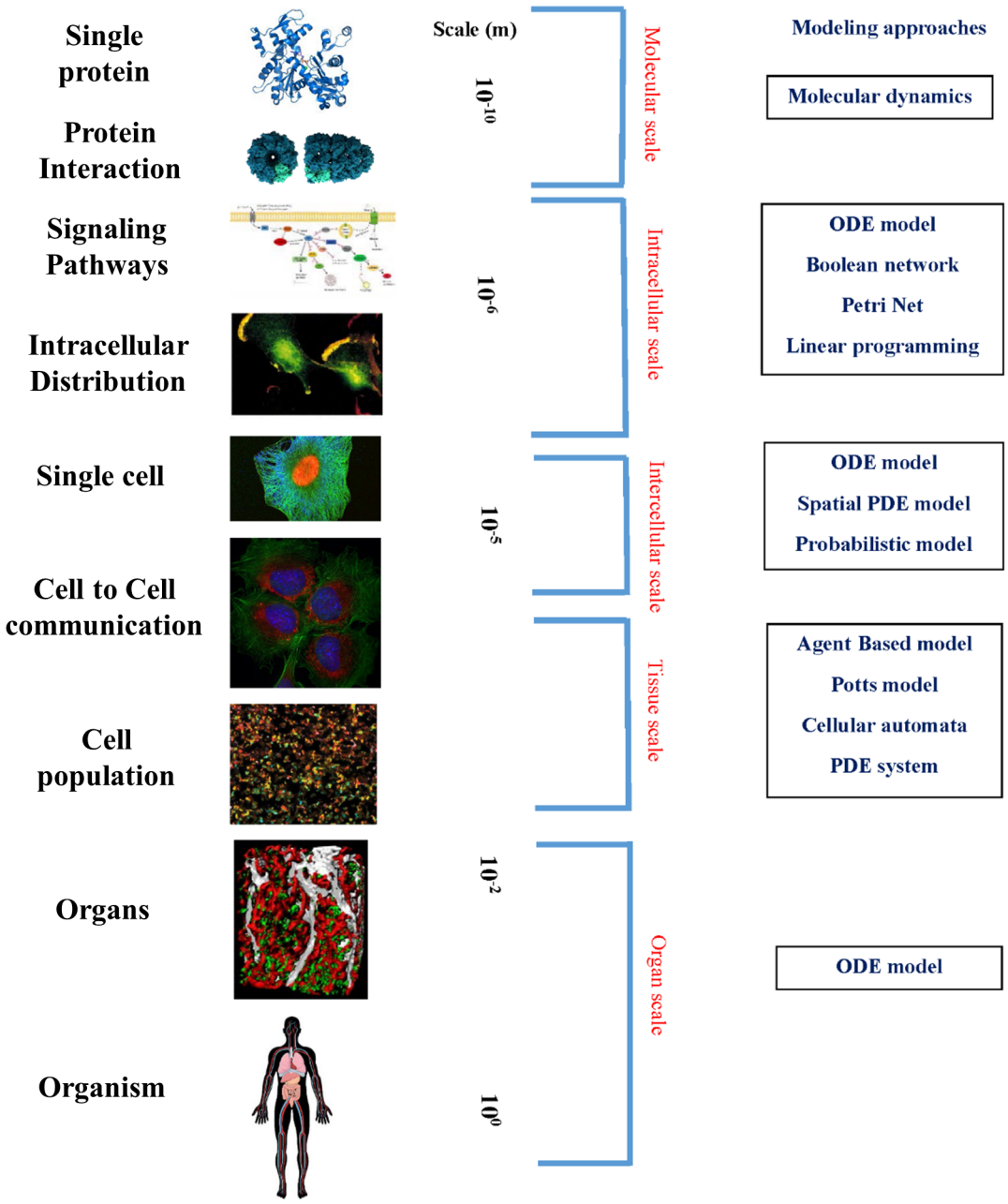


Figure 1.1. Different scales in the biological system and corresponding modeling approaches

The computational approach for multiscale modeling required several simulation approaches as different scale and time are involved. There exist several strategies to traverse and connect the link between different spatial and temporal scale using multiscale modeling. One plan is to observe the features at a high level and deduce the fundamental nature and mechanism from there to lower range. This approach is known as Top-down [3]. The main advantage of the Top-down approach is that the initial level is directly backed up by data. So, the hypothesis can include more fundamental property stepwise from a higher level to a lower level based on initial data. The models are easy to grasp, and mechanisms are relatively simple. The top-down approach usually tries to do the reverse engineering to get the underlying mechanism at a lower scale. But higher-level phenomena are interconnected with different pathways of molecular interaction towards the lower level. Sometimes, there can be multiple phenomena for a single mechanism or single mechanism for various behavior. It is difficult to deduce a specific fundamental explanation and mechanism for the system as there is no direct connection to the physiological parameters. The models are mainly phenomenological. Another strategy is to observe the behavior at different temporal and spatial scale starting from the fundamental level of molecular interactions and dynamics. It is known as Bottom-up [4]. The Bottom-up modeling approach tends to simulate the individual components and their interactions directly. Although the method gives a complete understanding of the system from fundamental features to higher level dynamics of the cell population, assembling such model from signaling network of molecular interactions to higher scale is challenging. Due to the lack of knowledge of the biochemistry of a specific cell type, it is necessary to speculate the molecular component and signaling mechanism in most of the cases. Again, different networks at the molecular scale or signaling pathways are interconnected.

To assemble a minimal network which will give observed phenomena at a higher level is quite complicated. Most of the time the model is system specific and does not reflect the real-time representation of the cellular biochemistry. Because of the interconnectivity

between different network or signaling pathways and lack of knowledge of biochemistry, Bottom-up model may face ambiguity at the fundamental level. So, it is required to invest a lot of care while constructing the first layer at a low scale. Sometimes, it is difficult to grasp, and the simulations are computationally intensive. Bottom-up modeling approach can remove the gap between lower scales to higher scales of a biological system. These models can be used to study the emerging properties of a considerable number of interacting components at a molecular scale. Bottom-up approaches are robust and adaptive. It unveils the features of a biological system at a fundamental level and opens up new directions for experiments.

All the phenomena at tissue or organ scale are based on the molecular interaction in inside, surface and extracellular environment of the cell. Normally, the model can make predictions based on the available parameters acquired from experiments. Due to the presence of different spatial and temporal scale for the biological system, it is difficult to represent the model with the experimentally observed parameter directly. System biology is driven by hypothesis [5, 6]. Small scale experiments are performed for different levels. Then a hypothesis is necessary to link the gap between different scales. The detailed mechanical insight of the system is observed by constructing a systematic model based on the observed data and hypothesis. Over time the molecular interaction networks are becoming more complex. It became difficult to extract the information on molecular and population dynamics at the different casual and temporal hierarchy. It is necessary to make a hypothesis and use phenomenological parameters to link different scales. For instance, cellular birth, death, and proliferation occur at longer time scale compare to the molecular interactions. Normally, the information associated with intracellular signaling pathways is processed within minutes. However, the death, division, and proliferation for eukaryotic cells can take hours. During that time, the mother cell split into two daughter cells while the major parts are modified and duplicated. If we want to explain the behavior of the cell population from the molecular interactions, then it is necessary to hypothesize how

cellular birth, death, and proliferation rate is connected with the molecular interactions. Modeling of cellular biochemistry involving cellular death, division or phenotype in more detail from the molecular scale will deepen our understanding of how cell process and make a decision. But the size of the model will increase as well as the computational cost for simulating the whole system. Multiscale modeling approach can address this issue by increasing computational efficiency and scalability. This has been used to study tissue engineering [7], tumor growth [8], neural systems [9, 10], cardiovascular fields [11] and other physiopathological processes [12–16].

The computational and mathematical description at different scale of a biological systems can be modeled by deterministic, discrete and rule-based, and stochastic approaches. These models can help us to explore and connect the gap between different scale of system biology. Some of the current modeling approaches are ordinary differential equations (ODEs) [17], partial differential equations (PDEs) [18], chemical master equation (CME) [19], Langevin equation, linear programming [20, 21], Boolean network [22], agent based model [23, 24], Petri Nets [25], Potts model [26, 27], finite state automata [28], models based on genetic variations [29] and so on.

Differential equations mainly model deterministic approaches (e.g., ODEs, PDEs or coupled ODEs). ODE models are suitable to simulate the kinetics for small scale intercellular and intracellular networks [30]. Due to their simplicity, these models are widely used to model the continuous dynamics of the time-varying effect of complex biological processes [31, 32]. Some ODE models can also incorporate cell to cell interactions [17, 33]. But parameter estimation is a bottleneck for ODE models. Parameters can be obtained using Particle Swarm Optimization (PSO) or Genetic Algorithm (GA). But for larger scale network ODE models become computationally expensive with decrease efficiency. The time scales for ODE models are also high and normally start from microsecond. The system becomes computationally intensive if the time goes beyond microseconds. Normally, ODE models are used to study and explore the macroscopic properties. When spatial aspect and other

features of cellular behavior (e.g., association, dissociation, charge) are incorporated in ODE models, it became PDE models. For instance, if a molecule or particle is diffusing in space, the model becomes a PDE model of reaction-diffusion equations. Reaction-diffusion equations have been used in modeling spatial distribution of signaling molecule [34], cell to cell communication [35], pattern formation [36], cardiovascular system [37] and so on. PDEs can evaluate the spatiotemporal properties of a biological process.

In the molecular scale, stochastic phenomena are apparent due to thermodynamic fluctuations. This stochastic nature may play a significant role in cellular and tissue scale dynamics. Mathematically, the stochastic noise strength of a system is inversely proportional to the square root of the total number of molecules. So, the deterministic model needed the molecule number to be high enough to diminish the effect of noise. But in a biological system, the particles are distributed in space and concentration is low. Thus, the deterministic models may yield a misleading result and do not represent a real biological system. Stochastic or probabilistic models can account the random fluctuations of these systems. Gillespie stochastic simulation algorithm (SSA) is a widely used stochastic approach. The algorithm treats each reaction as Markov transition and simulates the associated chemical reactions. Direct Gillespie stochastic algorithm gives exact solution but the method is computationally intensive. Tau-leaping method is an improved version of Gillespie SSA to speed up the simulation. But the process has an underlying assumption and give an approximation of the exact solution. Chemical master equation (CME) also provides the precise solution of stochastic reactions. CME can be reduced to a chemical Langevin equation for a large number of molecules. These equations are in the form of a stochastic differential equation. Despite the reduction of CME to the Langevin equation, the system remains computationally intensive.

Probabilistic models can investigate the stochastic dynamics of molecular interaction. Many of the tools are incorporated with the stochastic algorithm in addition to deterministic process to simulate the molecular dynamics in temporal scales. The fluc-

tuations and noise in the molecular dynamics in temporal scale can also be modeled by implementing Gillespies Monte Carlo algorithm [38, 39]. [40] describes an approach to model the stochastic nature of single molecules. For a large set where the fluctuation averages out, stochastic nature of the system can be neglected. Stochastic simulations are far too computationally expensive than deterministic simulation. So, depending on the system sensitivity, an appropriate approach can be selected. But at the same time, the stochastic fluctuation can also affect a parameter in another scale. For example, stochastic change at molecular level can affect the collective behavior of the cell population. The variation in the expression of the protein copy number is observed depending on the accessibility of the DNA of different cells [41]. This randomness can affect the response of cells due to an external signal.

At present, discrete and rule-based models are widely used. These models can represent a biological system by spatially and temporally discrete methods. These methods are computationally cheap and do not require distinct functions. In a discrete particle model, particles, molecules or atom is considered individually and can be represented by either deterministic or stochastic models. Some of the modeling approaches are Molecular dynamics (MD) [42], Monte Carlo (MC) [43], Direct simulation of Monte Carlo (DSMC) [44], Lattice Boltzmann (LB) [45], Brownian dynamics (BD) [46]. In MD, the interparticle potential is evaluated based on the fundamental solution of a particle of motion. MC is a molecular modeling tool focuses on the mapping of the system based on stochastic Markov-based framework. DSMC and LB are used to solve fluid-related systems. BD algorithm can simulate the dynamics of molecules and particles in a solution. So, BD algorithm has been widely used in modeling the distribution and penetration of extracellular signaling molecules and drug delivery nanoparticles. In a discrete agent-based model, the overall system is represented by a set of rules. So, it is known as rule-based or agent-based modeling. Petri nets (PN) is an agent-based model focusing on modeling dynamic discrete events. The mathematical model of PN is based on graphical representation to simulate the

gene regulatory networks [47], signaling pathways [48] and metabolic pathways [49, 50]. But the graphical representation for the complex biological system is still difficult to study as the system and approach are too primitive. Finite state automata models are focused on the consequence and transition of the state of an element. The state of a certain element is based on rules, state of that element and state of the neighboring elements. The Boolean network is a random network model with discrete state and time of an element. Boolean function determined the state of an element. It also requires random inputs. Boolean network is used to model gene networks [51] and cell cycle control [52]. Potts model is the general form of Ising model. It is capable of simulating the dynamics of cell population due to mechanical contact and shape of neighboring cells [26, 27, 53]. The elements of Potts model can have multiple state.

Agent-based modeling focuses on the interaction and actions of autonomous agents and simulate the whole system. The multiscale agent-based model can address some of the limitations of the above models. Usually, in the agent-based model, a cell is considered as an individual agent. But a protein, a molecule or an organism can also be regarded as an agent. In the multiscale agent-based model, the dynamics of each cell can be derived from fundamental features (e.g., signaling or regulatory network) and can be connected with the higher scale collective behavior of cell population. It can also incorporate different temporal scales such as the dynamics of intracellular signaling molecule at short time scale, cellular birth, differentiation or apoptosis in medium time scale and the response of drug delivery at longer time scale [54]. So, agent-based modeling has been widely used to model the multicellular communication, dispersion of particles, population of migrating cells, immune response or excitable tissues.

Figure 1.1 shows different scales in the biological system spanning from single protein to organism scale and their associated modeling approaches (as discussed before). In this dissertation, tissue scale dynamics are considered as higher scale phenomena. Experimental data at tissue scales are widely studied, and we have a better understanding of those.

But the underlying fundamental features at the molecular scale are still clouded. The penetration and dispersion of nanoparticles and signaling molecules in the extracellular domain of a tissue architecture are connected with the intracellular biochemical reactions of cells. Again, the dynamics of cell population depends on the microscopic scale gene expression rates to macroscopic scale collective behavior of cells. On top of that cellular death, division, growth, and heterogeneity impose intrinsic and extrinsic noise on the system. These are the key parameters to consider in the design of a drug delivery system. The complexity of modeling from the microscopic scale to tissue scale and connect them efficiently have been a major challenge. Modeling of such scheme is computationally challenging as well. To address that the Bottom-up approach is taken to study the extracellular and intracellular behavior of a Tissue architecture. Since different components of cells are involved with different space and time, several computational models are developed to study the multiscale property of the system quantitatively. The extracellular behavior can be investigated considering molecular scale (e.g., drug molecule) or nanoscale (e.g., drug delivery nanoparticles) as a fundamental scale. From there we can study the effect of size, shape and charge distribution of molecules or nanoparticles on tissue scale. The fundamental scale to study the collective behavior of the cell can be the intracellular signaling pathways at the molecular level. Here, multiscale spatiotemporal modeling approaches have been adopted to understand the intracellular and extra-cellular behavior of tissue architecture and address the current challenges and limitations. The following procedures in this dissertation are essential to fully understand and connect the multiscale behavior of any biological process and thus overcome the barrier to design a successful drug delivery system.

In tissue scale, the critical phenomena are the distribution of signaling molecules and particles in the extracellular domain, interaction between particle and cell, cellular uptake of signaling molecules, intracellular reactions and collective behavior of cells. A wide range of space and time is involved in tissue scale as discussed earlier. Each of the phenomena is incorporated with its own hierarchical structure, and several models are

necessary to understand the whole dynamics and their underlying interconnectivity. The multiscale agent-based models can address these challenges. First, a multiscale model is developed to realistically capture the transport behavior and cellular interactions of particles or molecules. In many aspects, a biological tissue can be compared with a heterogeneous porous media. Particle motion through the interstitial space of biological tissue is subject to advection, diffusion, and interaction with the cell boundaries. Tissue-scale particle distribution may occur over hours. However, the process is ultimately determined by the microscale adhesion and interaction of particles with the cell boundaries. Bridging these spatiotemporal phenomena at distinct spatial and temporal resolutions in a model could be computationally expensive. At a fundamental scale, BD algorithm is incorporated to include the randomness, and the tissue scale phenomena are modeled based on Stoke's equation. The simulation of BD algorithm over large scale and time is computationally intensive as well. So, BD algorithm is modified by time-adaptation. Here, a time-adaptive Brownian Dynamics (BD) simulation algorithm is developed. Then the algorithm is combined with the Method of Regularized Stokeslets (MRS) [55]. The integrated algorithm enabled multiscale simulation of particle transport under both advection and diffusion in a heterogeneous porous system. The time-adaptive feature captured particle-cell interactions at high resolution while allowing efficient computation. This multiscale model is used to study the mechanistic features of the experimental result of Wong et al. [56] and Tang et al. [57]. Their experimental data represents the distribution of different size drug delivery nanoparticles.

The hypothesis based on the experimental result is that the distribution of nanoparticles may differ in in vivo and in vitro condition due to particle size and particle-cell interaction. Drug-delivery nanoparticles are subject to a variety of transport barriers in biological tissues [58, 59]. To overcome these barriers, significant research efforts have been made over the years to study the principles of drug-delivery nanoparticle design [60]. The key nanoparticle design features that have been widely studied are particle size, geometry, and surface-attached targeting molecules [61]. Among these, the size of a particle

is believed to have significant effects on its immune clearance, transvascular delivery, and intra-tissue dispersion and penetration [61, 62]. The particle-cell interaction due to the presence of cells may also play an important role in the distribution of nanoparticles in the extracellular domain.

Then a second multiscale modeling framework is developed to connect the tissue scale collective behavior of cells to the intracellular and extracellular reaction network. The fundamental scale in this framework is based on the signaling pathways derived from interacting molecules and reactions. Normally, the molecules of the signaling pathways of the reaction network is modeled as featureless entities. Their molecular interaction is well defined by the laws of the mass of action, and ODE models can be used to study such cases. Insights from the structural analysis of molecules and nanoparticles can be incorporated in the signaling pathways as ligation of the receptor, ligand on specific binding sites. Some existing software tools can generate full reaction network and mathematical description of the multimolecular complex from specified bimolecular reactions [63–66]. But the system required the quantitative data of enzymatic reactions, association and dissociation rate for specific binding sites as input. Lack of availability of the quantitative data is the current bottleneck of the system. There exist other software tools to quantitatively simulate the molecular dynamics of signaling pathways at different temporal scale [64–74]. These tools are easy to use as technical aspects (e.g., Differential rate equation) are integrated into the system. For example, a parallel software framework, using discrete agent-based simulation, has been proposed in [75]. It models the behavior of a large cell population and updates molecular concentration using coupled Partial Differential Equations (PDE). A coarse-grained parallel approach is implemented in [76], to perform independent stochastic simulation. In [39], a C++ based stochastic and multi-scale simulation toolkit is proposed for the chemically reacting system. The performance of Gillespie Stochastic Simulation is accelerated in [77] using Graphics Processing Units (GPUs). A parallel algorithm for off-lattice individual-based models of multicellular populations is presented in [78]. Gillespie's

First Reaction is applied in [79] to present a stochastic simulation software framework for biochemical reaction networks. A graph-based model for parallel, distributed and portable applications is introduced in [80], and finally, a parallel algorithm is designed in [81], focusing on simulation of reaction-diffusion based system. Most of the tools are still not able to address and connect all the spatial and temporal scale as the computational and conceptual cost is high. Only a few of them can perform the spatial simulation to connect the gap between subcellular distribution and signaling network in a small volume. Most of the tools assumed a well-mixed system. So, the distribution of the molecules can be considered homogeneous and heterogeneous aspect of spatial distribution can be neglected. But in this case, system specific mathematical description is necessary. Also, the system will not represent a real biological system.

Here, a scalable parallel framework is developed to realistically simulate a biological system in tissue scale conserving the efficiency and accuracy. It provides a unique capability to systematically expand a single-cell biochemical network model into a cell population model. Using a message passing interface (MPI) [82] or multiprocessing [83] parallel algorithm, the models can bridge cellular processes to the temporal molecular events in signaling and gene transcription. Direct Gillespie SSA is used to model the single cell reaction network in fundamental scale. This model gives the exact solution mathematically but computationally intensive. So, this model is incorporated into a multiscale parallel framework to speed up the simulation.

In an unconventional approach, the framework launches parallel simulations on a single-cell biochemical network model and then treats each stand-alone parallel process as a cell object. Under the MPI/multiprocessing scheme, each parallel process behaves like an agent or software object of an agent-based model [84], and together, all parallel processes represent the cell population. Cellular heterogeneities can be introduced in the population by creating parallel processes with distinct parameter values or initial conditions. The cell objects (parallel processes) are communicated, synchronized, and controlled remotely

using MPI/multiprocessing communications. The cell objects evolve through death and division based on the state variables representing intracellular network species. The death of a cell object is simulated by terminating the process. The division of a cell is simulated by creating a new process (daughter cell) from an existing process (mother cell). Cellular attributes and memory can be passed from one process to another (mother to daughter) during the cell division process.

To implement the above scheme, a model is broken down into two separate computer programs. One program describes the cellular processes (cell death, division, or changes in the phenotypes) and their dependencies on the intracellular network species. In the other program, a complete biochemical network model describing the intracellular events is defined. Then communication between these two programs in a server-client fashion to connect the cellular processes and the intracellular network is enabled.

The separation of the cellular and intracellular molecular-scale processes into two programs provides modularity and versatility in model development. It permits a signaling pathway or gene transcription network model to be defined separately and then readily expanded into a population model. This distributed framework can enable scalable model development considering computationally expensive mechanistic details at the single-cell level.

Finally, the efficiency and scalability of our framework are analyzed with a real-time example of bacterial *Quorum Sensing (QS)*. Large population of bacteria communicates with one another by releasing signaling molecules, called *autoinducers*, into the environment. Bacteria are also capable of sensing the environmental autoinducer concentration and regulating the expression of certain specific genes in a coordinated manner. This mechanism of communication and mutual regulation is called *Quorum Sensing (QS)* [85]. Communication via QS has been observed in a wide range of bacteria species, such as marine bacteria (like *Vibrio fischeri*) and pathogenic bacteria [86, 87].

Since each cell responds uniquely to its environment, any cellular regulation and signaling are prone to stochastic fluctuation. There have been attempts to study how the population of bacteria achieve coordinated gene expression, despite such noise [88]. Sequential stochastic modeling of QS considers all reactions within the system, one reaction at a time. It makes modeling of a large population of cells significantly more expensive, concerning time and computational resources. [86].

Let us consider an example of bacterial growth in rich media, where inter-cellular communication may be assumed to be negligible. Such a system can be implemented in parallel due to the absence of significant dependency among the cells. However, in the case of QS, cellular interaction via autoinducers plays a pivotal role in the coordinated behavior of the system. *An ideal parallel QS framework must, therefore, incorporate, both, modeling accuracy of molecular (especially autoinducer) concentration, as well as efficiency in terms of time and resource utilization*. Among the aforementioned literature, only [81] takes cellular interaction via environment into consideration. However, even that work neither discusses the inevitable trade-off between the parallelism and accuracy, nor the capability of accurately modeling population dynamics due to cell birth and death.

In this dissertation, the first step is taken towards developing a scalable parallel framework for modeling biochemical network that meets both the requirements of accuracy and speed-up. This framework is applied to model QS in bacteria, where each cell is a process that exchanges messages with the master (or coordinator) process. It incorporates a simple approximation to maintain the uniformity of the environmental parameters. Simulation experiments show that this framework captures the dynamics of molecular concentration as accurately as the standard sequential QS model [86]. This system is analyzed in light of how sampling interval affects the overall accuracy and variation of computation overhead due to varying concentration of molecules. It is also discussed how this framework handles evolution due to cell birth and death. This framework exhibits higher speed-up and more balanced CPU usage when compared to the sequential model. Furthermore, cellular

heterogeneity and phenotypic variability are incorporated by sampling the QS system parameters from Gaussian distribution. It is noteworthy that existing literature on QS [86][89] has modeled up to a population of 240 cells, whereas the proposed framework has been used to simulate a population of 2000 cells. Scalability experiments are performed on 50 cores of Forge high-performance computing clusters built on Rocks 6.1.1, while other analyses are performed in Ubuntu 14.04 system with 8 CPUs.

1.2. MOTIVATION AND OBJECTIVE

First, a multiscale spatiotemporal model is developed to study the distribution and penetration of particles or molecules and their interaction with cells. Two earlier studies quantitatively investigated the effects of particle size on the efficacy of tissue delivery and penetration of drug-delivery nanoparticles [56, 57]. Nonetheless, the mechanistic aspects of these effects remain poorly understood. Earlier, an experiment by Wong et al. [56] indicated enhanced tissue penetration as a result of particle size reduction. Later, Tang et al. [57] reported similar effects from particle size variation but their experimental data revealed significantly narrower tissue distribution profiles and penetration of particles. Moreover, in Tang et al. [57], the effects of particle size variation appeared relatively modest. These apparent disparities motivated us to develop a multiscale model and mechanistically interrogate particle size effects on their efficacy of tissue distribution and penetration. The two studies above carried out investigations in different experimental settings. Wong et al. [56] employed in vitro experiments involving cell-free collagen tissue. On the other hand, the experiments of Tang et al. [57] were conducted in in vivo tumor tissues. So, an investigation was carried out about how these two experimental settings might affect the intra-tissue transport behavior and penetration efficacy of nanoparticles of different sizes.

Using the model, experimental data reported in Wong et al. [56] and Tang et al. [57] was analyzed. The analysis revealed how the different tissue conditions in these two experimental studies could lead to distinct particle distribution profiles and size effects.

Results and analysis indicate that particle size effects may appear pronounced in a cell-free tissue system, such as collagen matrix, often employed in in vitro microfluidic studies. In the absence of particle-cell interaction and under pure diffusion, particle size may have more dramatic effects on the tissue distribution and penetration efficacy of nanoparticles. However, in in vivo physiological conditions, the barriers imposed by the interstitial cells may moderate the effects arising from the particle size difference. In Section 2, it is shown that that particle-cell interaction imposes significant transport barriers and serves as a critical determinant of distribution and penetration efficacy of nanoparticles.

Again, cell fate decisions and phenotypes are determined by the subcellular molecular events in signaling and gene transcription [90–92]. The collective behavior and evolution of cells are linked to these subcellular events. Nevertheless, most of the cell population models are decoupled from signaling and gene transcription. A key challenge to modeling multicellular systems considering signaling and gene transcription is the multiscale nature of the problem. Cell death, division, and evolution take place at a longer time scale compared to intracellular biochemical transformations in signaling and gene transcription. Bridging these multiscale phenomena in a model is imperative to mechanistically study cellular development and evolution. So, a second multiscale parallel framework is developed to bridge the gap between intracellular biochemistry to population dynamics at tissue scale.

The framework is demonstrated with two example models. In one model, cells are treated as independent objects. In the other model, cellular dependency by incorporating cell-to-cell communications is considered. Using the former model, the computational performance of the MPI/multiprocessing formulation is analyzed. This model is also used to demonstrate how MPI/multiprocessing could be used to model a temporally-changing growth environment and link it to a multicellular system of evolving cell populations. The latter model is utilized to show how MPI/multiprocessing could be used to model cellular

interdependencies arising from cell-to-cell communication in an evolving cell population. Finally, the framework is validated demonstrating the accuracy and efficiency for bacterial quorum sensing.

1.3. DISSERTATION STRUCTURE

This dissertation is organized as follows. Section 2 presents the methodology and results of a multiscale modeling study of particle size effects on the tissue penetration efficacy of drug-delivery nanoparticles. Section 3 discusses the proof of concept for multicellular models bridging subcellular biochemistry to population dynamics. Section 4 shows the efficiency and scalability of a scalable parallel framework for multicellular communication based on a real-time example in bacterial quorum sensing. Finally, Section 5 closes the dissertation with concluding remarks and future recommendations.

2. A MULTISCALE MODELING STUDY OF PARTICLE SIZE EFFECTS ON THE TISSUE PENETRATION EFFICACY OF DRUG-DELIVERY NANOPARTICLES

In this Section, a multiscale spatiotemporal model is discussed to study the distribution of particles or molecules in biological tissue. The mechanism and dynamics of particle distribution in tissue architecture have a different scale of space and time. In the fundamental scale, molecular interaction, particle-particle interaction, particle-cell interaction and particle uptake by the cell are the significant physiological phenomena. They are connected to tissue scale dispersion and penetration. Experiments are performed to observe the distribution and penetration of particles in in vivo and in in vitro condition. But the fundamental explanation for the observed experimental results is yet to be adequately understood. So, a Bottom-up approach is adopted here to study the tissue scale phenomena from the fundamental scale. This multiscale spatiotemporal model is developed focusing on the experimental results of Wong et al. and Tang et al. [56, 57]. Their result shows the size effect of nanoparticle distribution in biological tissue. So, the nanoscale is selected as a fundamental scale. But the model is generic and applicable to other range of space and time as well.

This multiscale model links microscale particle-cell interactions and adhesion dynamics to tissue-scale particle dispersion and penetration. The model is based on a time-adaptive Brownian Dynamics algorithm which accelerate the Brownian dynamics algorithm. Method of regularized Stokeslets is used for the mathematical description of the system. The combination of both of these models can account for the advection, diffusion, and cellular uptakes of particles. The experimental result of Wong et al. and Tang et al. [56, 57] show that particle size is a key parameter in drug-delivery nanoparticle design. It is believed that the size of a nanoparticle may have significant effects on its ability to overcome the transport barriers in biological tissues. Nonetheless, such effects remain poorly understood. Using a

multiscale model, this work investigates particle size effects on the tissue distribution and penetration efficacy of drug-delivery nanoparticles. This multiscale model can consider a particle with a certain size and account for effect arising from particle size in particle-cell interaction and dispersion.

Based on the published experimental works that investigated particle size effects in in vitro and in vivo tissue conditions, the effects of particle size in the intra-tissue dispersion and penetration is analyzed. Results show that impact of particle size may appear pronounced in an in vitro cell-free tissue system, such as collagen matrix. In an in vivo tissue system, the effects of particle size could be relatively modest. A detailed analysis of how particle-cell interactions may determine the distribution and penetration of nanoparticles in biological tissue is provided here. This work suggests that the size of a nanoparticle may play a less significant role in its ability to overcome the intra-tissue transport barriers. Results show that experiments involving cell-free tissue systems may yield misleading observations of particle size effects due to the absence of advective transport and particle-cell interactions.

This Section is organized as follows. In Section 2.1, a detailed description of the methodology used in the adaptive Brownian dynamic algorithm is provided. In Section 2.2, results and comparison with experimental and theoretical data are presented. Section 2.3 describes a detailed discussion on the current limitation and future perspective of the model. Finally, this Section is concluded with remarks in Section 2.4.

2.1. METHODS

2.1.1. Materials and Methods. Below, the simulation approach together with the model of nanoparticle transport in biological tissues is described. The model is written in C++.

2.1.2. Domain Representation of Biological Tissue. The computational domain in this model represents a two-dimensional rectangular tissue section (Figure 2.1). The entire domain is reffered by Ω, and its left, lower, and upper edges by Ω_1, Ω_2, and Ω_3,

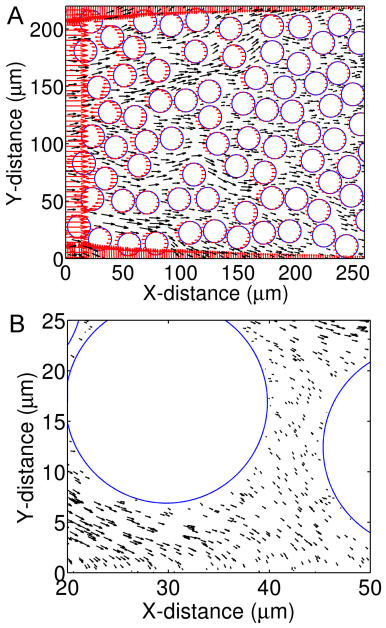


Figure 2.1. The MRS calculated force and velocity fields in a rectangular tissue section. (A) The red arrows represent force vectors at discrete locations along the domain edges and cell boundaries. The black arrows represent velocity vectors in the interstitial space. (B) A zoomed-in view of the velocity vectors in the interstitial space and near the cell boundaries

respectively. The rectangle is considered sufficiently wide such that the right edge can be ignored. The bottom-left corner of the domain ($\Omega_1 \cap \Omega_2$) represents the origin, and any point $\boldsymbol{x} \in \Omega$ represents a position with respect to this origin. The left edge, Ω_1, represents a porous capillary wall from where nanoparticles enter into the tissue space. The entry points of particles, $\boldsymbol{x} \in \Omega_1$, are selected randomly along this edge. The horizontal distance to the right with respect to Ω_1 represents tissue depth (labeled as X-distance in (Figure 2.1). The mobile nanoparticles is treated as circular objects with a defined size (radius) and each cell as a stationary circle of 10 μm radius. Cells are populated at non-overlapping random positions in the domain. The cells occupy 40% area of the domain area. This aggregate area occupied by the cells is referred as Λ. The remaining 60% area represents the interstitial space, which is referred to as Γ. The boundary of any cell $i \in \{1, 2, \cdots, n\}$ is referred as P_i, and the region it occupies as A_i. Therefore $\Lambda = (\cup_{i=1}^{n} P_i) \cup (\cup_{i=1}^{n} A_i)$. Thus, the entire computational domain, Ω is equal to $(\cup_{i=1}^{3} \Omega_i) \cup \Gamma \cup \Lambda$.

2.1.3. Nanoparticle Velocity. To evaluate nanoparticle velocities in the domain, the approach of Rejniak et al. [93] is adopted. At any position $\boldsymbol{x} \in \Omega$, the velocity of a nanoparticle is represented by the local fluid velocity $\boldsymbol{v(x)}$ (rejniak2013role). As in [93], the simulation is computed $\boldsymbol{v(x)}$ using the Method of Regularized Stokeslets (MRS) [55]. The MRS [55] has been used to model complex solid-fluid interactions in a variety of Stokes flow systems [94–98]. Here, for completeness, a brief description of the MRS and its implementation in this model is provided.

2.1.3.1. The Method of Regularized Stokeslets (MRS). The MRS is a Lagrangian approximation of the Stokes equations. It provides a convenient framework to avoid singularities associated with the fundamental solutions of the Stokes equations. Because of this property, the method is particularly useful for modeling Stokes flow associated with irregular geometries or non-smooth boundaries. The Stokes equations in two or three dimension are as follows:

$$\mu \frac{\partial^2 \boldsymbol{u}}{\partial^2 \boldsymbol{x}} = \nabla P - \boldsymbol{f}$$
$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = 0$$

In the above equations, μ is the fluid viscosity; $\boldsymbol{x}$ is a position vector; $\boldsymbol{f}$ is force; and P is pressure. $\boldsymbol{u}(\boldsymbol{x})$ is the local fluid velocity vector at $\boldsymbol{x}$. The Stokes equations can be solved for a single point force at $\boldsymbol{x}_0$, $\boldsymbol{f} = \boldsymbol{f}_0 \delta(\boldsymbol{x} - \boldsymbol{x}_0)$, where $\delta(\boldsymbol{x})$ represents the Dirac delta function.

$$\mu \frac{\partial^2 \boldsymbol{u}}{\partial^2 \boldsymbol{x}} = \nabla p(\boldsymbol{x}) - \boldsymbol{f}_0 \delta(\boldsymbol{x} - \boldsymbol{x}_0)$$
$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = 0$$

The solution of the above equations represents the velocity $\boldsymbol{u}(\boldsymbol{x})$ at $\boldsymbol{x}$ due to the single point force at $\boldsymbol{x}_0$. This solution, however, is singular at the point of application of the force (i.e., $|\boldsymbol{u}(\boldsymbol{x})| \to \infty$, as $\boldsymbol{x} \to \boldsymbol{x}_0$). To avoid this singularity, the MRS avoids direct use of the

point force $\boldsymbol{f}_0\delta(\boldsymbol{x}-\boldsymbol{x}_0)$ in the Stokes equations. Instead, it approximates (regularizes) the point force into a smooth, radially-symmetric force centered at $\boldsymbol{x}_0$: $\boldsymbol{f}_0\phi(\boldsymbol{x}-\boldsymbol{x}_0)$. With this regularized force term, the Stokes equations take the following form:

$$\mu\frac{\partial^2\boldsymbol{u}}{\partial^2\boldsymbol{x}} = \nabla P(\boldsymbol{x}) - \boldsymbol{f}_0\phi_\epsilon(\boldsymbol{x}-\boldsymbol{x}_0) \tag{2.1}$$

$$\nabla\cdot\boldsymbol{u}(\boldsymbol{x}) = 0 \tag{2.2}$$

The function $\phi_\epsilon(\boldsymbol{x})$ is known as cutoff function, which represents a spatially-symmetric sphere or blob of radius ϵ in the domain space. The regularized force $\boldsymbol{f}_0\phi_\epsilon(\boldsymbol{x}-\boldsymbol{x}_0)$ takes the maximum value at the center ($\boldsymbol{x}_0$), and decays smoothly towards the surface of the blob. The cutoff function satisfies the constraint $\int_{-\infty}^{+\infty}\phi_\epsilon(\boldsymbol{x})d(\boldsymbol{x}) = 1$. As $\epsilon\to 0$, $\phi_\epsilon(\boldsymbol{x})\to\delta(\boldsymbol{x})$, and the regularized force approaches the point force.

For an appropriate choice of the cutoff function $\phi_\epsilon(\boldsymbol{x})$, Equation 2.1 and 2.2 can be solved to evaluate the fluid velocity $\boldsymbol{u}(\boldsymbol{x})$ due to the regularized point force centered at any arbitrary position $\boldsymbol{x}_0$ in the fluid. Unlike the Stokes solution, the resulting velocity is non-singular at $\boldsymbol{x}_0$.

Now, the force field over the entire domain can be represented by a collection of N discrete point forces located at different points in the domain. If $\boldsymbol{f}_k$ located at $\boldsymbol{x}_k$ for $k\in\{1,2,\cdots,N\}$ represents such a point force, its contribution at $\boldsymbol{x}$ can be represented as $\boldsymbol{u}_k(\boldsymbol{x})$. By solving Equation 2.1 and 2.2, $\boldsymbol{u}_k(\boldsymbol{x})$ for $k\in\{1,2,\cdots,N\}$ can be evaluated. Then, the net velocity at $\boldsymbol{x}$, $\boldsymbol{v}(\boldsymbol{x})$, can be evaluated simply by linear superposition of the solutions corresponding to the N discrete forces: $\boldsymbol{v}(\boldsymbol{x}) = \sum_{k=1}^{N}\boldsymbol{u}_k(\boldsymbol{x})$

2.1.3.2. Force and velocity calculation. Following Rejniak et al. [93] and Tlupova et al. [99], the value of $\phi_\epsilon(\boldsymbol{x})$ is chosen as $\phi_\epsilon(\boldsymbol{x}) = \frac{2\epsilon^4}{\pi(r^2+\epsilon^2)^3}$, where $r = |\boldsymbol{x}|$. The solid boundaries of the tissue domain is discretized into $N = 6,700$ discrete points. The solid boundaries include the three domain edges (Ω_1, Ω_2, and Ω_3), and the boundaries of the circular cells, P_i for $i\in\{1,2,\cdots,n\}$. For the above cutoff function, the solution of Equation

2.1 and 2.2 is:

$$\begin{aligned} \boldsymbol{u}_k(\boldsymbol{x}) &= -\frac{\boldsymbol{f}_k}{8\pi\mu}\left(\ln\left(r^2+\epsilon^2\right)-\frac{2\epsilon^2}{r^2+\epsilon^2}\right) \\ &\quad +\frac{1}{4\pi\mu}\frac{1}{r^2+\epsilon^2}\left[\boldsymbol{f}_k.\left(\boldsymbol{x}-\boldsymbol{x}_k\right)\right]\left(\boldsymbol{x}-\boldsymbol{x}_k\right) . \end{aligned} \tag{2.3}$$

For the entire collection of the N discrete forces, the net velocity $\boldsymbol{v}(x)$ is obtained by linear addition of the solutions:

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{x}) &= \sum_{k=1}^{N}\boldsymbol{u}_k(\boldsymbol{x}) \\ &= \sum_{k=1}^{N}\left\{-\frac{\boldsymbol{f}_k}{8\pi\mu}\left(\ln\left(r^2+\epsilon^2\right)-\frac{2\epsilon^2}{r^2+\epsilon^2}\right)\right. \\ &\quad \left.+\frac{1}{4\pi\mu}\frac{1}{r^2+\epsilon^2}\left[\boldsymbol{f}_k.\left(\boldsymbol{x}-\boldsymbol{x}_k\right)\right]\left(\boldsymbol{x}-\boldsymbol{x}_k\right)\right\} . \end{aligned} \tag{2.4}$$

However, to obtain $\boldsymbol{v}(\boldsymbol{x})$ using Equation 2.4 (or $\boldsymbol{u}_k(\boldsymbol{x})$ using Equation 2.3), the unknown point forces, $\boldsymbol{f}_k$s, at the N discrete points needs to be evaluated first. To evaluate the $\boldsymbol{f}_k$s, a no-slip boundary conditions ($\boldsymbol{u}_k = 0$) is selected at the lower and upper domain edges (Ω_2 and Ω_3), and the cell boundaries P_i for $i \in \{1, 2, \cdots, n\}$. As mentioned previously, the left domain edge Ω_1 represents the particle or fluid entry points (the porous wall of a vascular capillary). At Ω_1, the boundary condition was set as $\boldsymbol{u}_k = 1\hat{\boldsymbol{j}}$ μm/second, where $\hat{\boldsymbol{j}}$ represents a unit vector towards the tissue depth (parallel to Ω_2 or Ω_3). Thus, for the N discrete points, a system of N independent linear equations from (Equation 2.4) is obtained. The left hand-side ($\boldsymbol{u}(\boldsymbol{x})$) of these equations were defined (either 0 or $\hat{\boldsymbol{j}}$), whereas the right-hand side contained the N unknown force terms $\boldsymbol{f}_k$s. Using the GSL package (https://www.gnu.org/software/gsl/), this system of linear equations is solved to evaluate the unknown $\boldsymbol{f}_k$s at the N discrete points. Then these force terms are plugged into Equation 2.4 to evaluate the velocity vector $\boldsymbol{v}(\boldsymbol{x})$ at any arbitrary position $\boldsymbol{x}$ in the interstitial space of the domain.

In Figure 2.1, the force vectors, $\boldsymbol{f}_k$s are represented by red arrows. The length and direction of each red arrow represent the relative magnitude and direction of the corresponding force vector at the indicated point. The velocity vectors at different points of the interstitial space are represented by black arrows. The length and direction of each black arrow represent the relative magnitude and direction of the fluid (nanoparticle) velocity at the indicated point.

2.1.4. Nanoparticle Diffusion. The diffusion constants of the nanoparticles are calculated based on the Einstein-Stokes equation:

$$D = \frac{K_B T_p}{6\pi\mu a} \tag{2.5}$$

where D is diffusion constant of a particle, K_B is the Boltzmann constant, T_p is temperature, μ is viscosity of the interstitial fluid, and a is radius of the particle.

2.1.5. Time-Adaptive Simulation Algorithm. In the BD algorithm, the nanoparticles are considered as independent and mutually non-interacting in a biological tissue. This consideration is based on the fact that drug-delivery nanoparticles can reach a target tissue at small quantities. Typical particle concentration in a biological tissue is expected to be small. Therefore, it is less likely that their mutual interaction can have a significant impact on their transport behavior over other factors, such as fluid flow, collision with the cell boundaries, and cellular uptake. Because particles are considered independent, the model allows independent simulation of one particle at a time.

Figure 2.2 illustrates the time-adaptive scheme of the algorithm. The algorithm is summarized in a pseudocode in (Figure 2.3). In the algorithm, particles are advanced adaptively with time steps $\Delta t_m \geq \Delta t \geq \delta t$, where Δt_m and δt represent the largest and smallest permissible time step, respectively. During the simulation, in each BD step, the algorithm first computes R, which is the distance between the center of a particle and its nearest interaction point on a solid boundary (Figure 2.2). The solid boundary can be any

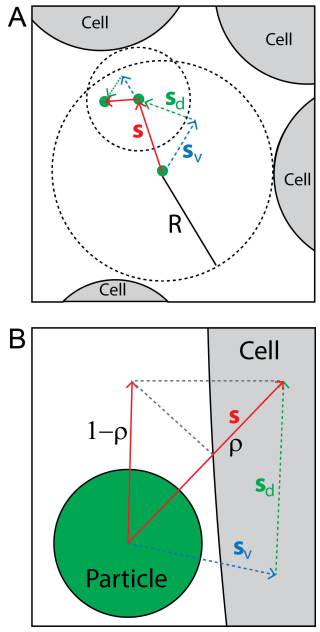


Figure 2.2. Illustration of the time-adaptive BD algorithm. (A) Particle motion in the bulk fluid. The small green circle represents a nanoparticle, and the large gray circles represent cells. The radius of the dashed circle, R, represents the distance between a particle's current position and its nearest cell boundary. In the bulk fluid, particle jump $\boldsymbol{S}$ is taken adaptively so that $|\boldsymbol{S}| < R$. $|\boldsymbol{S}|$ is determined by the time step Δt: $\boldsymbol{S} = \boldsymbol{S}_v + \boldsymbol{S_d}$, where $\boldsymbol{S}_v = \boldsymbol{v}\Delta t$ (displacement due to advection), and $\boldsymbol{S}_d = \sqrt{4D\Delta t}\boldsymbol{e}$ (displacement due to diffusion). (B) Particle motion near a cell boundary. $|\boldsymbol{S}|$ is determined by a constant but fine resolution time step $\delta t = 10^{-3}$ seconds. The cell boundary represents a sticky wall that captures or reflects a colliding particle with probability ρ and $1 - \rho$, respectively

of the three domain edges or cell boundaries. It then attempts to move the particle based on the largest permissible step Δt_m. It computes a possible jump: $\boldsymbol{S} = \boldsymbol{S}_v + \sqrt{4D\Delta t_m}\boldsymbol{e}$, where $\boldsymbol{S}_v = \boldsymbol{v}\Delta t_m$ represents displacement due to advection, $\boldsymbol{S}_d = \sqrt{4D\Delta t_m}$ represents displacement due to diffusion (Figure 2.2), and $\boldsymbol{e}$ represents a unit vector with random orientation. Velocity $\boldsymbol{v}$ and diffusion constant D are computed using the MRS and Einstein-Stokes equation, as detailed in the previous sections. If the jump length $|\boldsymbol{S}|$ is smaller than R, the move is accepted, and the particle position is updated accordingly. If the move based on Δt_m is rejected, the algorithm attempts to move the particle based on a new time step $\Delta t_a < \Delta t_m$. This time step Δt_a is obtained by solving $|\boldsymbol{v}|\Delta t_a + \sqrt{4D\Delta t_a} = R$. It then

1 Determine particle velocity $\boldsymbol{v}(\boldsymbol{x})$ (MRS).
2 Determine particle diffusion constant D (Einstein-Stokes).
3 Determine step size:
  a. Calculate R, distance between the particle and its nearest cell surface.
  b. Calculate time step Δt:

$$\Delta t = \begin{cases} \Delta t_m (0.1\ sec) & \text{if } |\boldsymbol{v}(\boldsymbol{x})\Delta t_m + \sqrt{4D\Delta t_m}\boldsymbol{e}| < R - a \\ \Delta t_a|_{|\boldsymbol{v}(\boldsymbol{x})|\Delta t_a + \sqrt{4D\Delta t_a} = R-a} & \text{if } |\boldsymbol{v}(\boldsymbol{x})\Delta t_m + \sqrt{4D\Delta t_m}\boldsymbol{e}| > R - a \\ & \quad \text{and } |\boldsymbol{v}(\boldsymbol{x})\Delta t_a + \sqrt{4D\Delta t_a}\boldsymbol{e}| > 4a \\ \delta t\ (10^{-3}\ sec) & \text{if } |\boldsymbol{v}(\boldsymbol{x})\Delta t_a + \sqrt{4D\Delta t_a}\boldsymbol{e}| \leq 4a \end{cases}$$

4 Calculate tentative particle jump: $\boldsymbol{S} = \boldsymbol{v}(\boldsymbol{x})\Delta t + \sqrt{4D\Delta t}\boldsymbol{e}$
5 Identify tentative particle position: $\boldsymbol{x}_t = \boldsymbol{x} + \boldsymbol{S}$
6 Update particle position:

$$\boldsymbol{x} = \begin{cases} \boldsymbol{x}_t & \text{if } (\boldsymbol{x}_t \in \Gamma) \\ \boldsymbol{x}_t\ (uptake\ by\ cell) & \text{if } (\boldsymbol{x}_t \in \Lambda)\ and\ (0 < U(0,1) \leq \rho)^* \\ \boldsymbol{x}_r\ (reflect\ by\ cell) & \text{if } (\boldsymbol{x}_t \in \Lambda)\ and\ (\rho < U(0,1) \leq 1)^{*,\dagger} \\ \boldsymbol{x}_r\ (reflect\ by\ domain\ boundaries) & \text{if } (\boldsymbol{x}_t \notin \Omega)^\dagger \end{cases}$$

*$U(0,1)$: a random number between 0 and 1.
†$\boldsymbol{x}_r$: new position due to reflection by a flat surface (Fig. 2B).

Figure 2.3. Pseudocode for the simulation algorithm

computes: $\boldsymbol{S} = \boldsymbol{v}\Delta t_a + \sqrt{4D\Delta t_a}\boldsymbol{e}$. The algorithm then compares $|\boldsymbol{S}|$ with the particle radius a. If $|\boldsymbol{S}| > 4a$ (i.e., the distance between the particle and a cell is at least twice the diameter of the particle), the move is accepted and the particle position is updated accordingly.

If $|\boldsymbol{S}| \leq 4a$, the algorithm attempts to move the particle based on the smallest permissible step δt: $\boldsymbol{S} = \boldsymbol{v}\delta t + \sqrt{4D\delta t}\boldsymbol{e}$. The move is accepted if the new particle position falls in the interstitial space (Γ). However, if the new position falls outside the domain edges, or in any of the cell regions (Λ), the algorithm treats it as a collision with the corresponding domain edge or cell boundary. In the former case, the particle is reflected by the domain boundary. In the latter case, the particle is captured or reflected by a cell boundary, as discussed in the next section.

2.1.6. Particle Interaction with Cell Boundaries. The cell boundaries are considered as sticky walls that can capture or reflect a hitting nanoparticle with a defined probability (Figure 2.2 B). Because a cell is much larger in size than a particle, a cell boundary is treated as a flat surface when a particle collides with the boundary (Figure 2.2

B). As mentioned in the previous section, a particle can hit a cell only when it is in the vicinity of a cell and advanced by the finest time step $\delta t = 10^{-3}$ seconds. This time step size requires any colliding particle to be within a few nanometers of a cell boundary. When a particle hits a cell, it is either captured with probability ρ, or reflected into the fluid with probability $(1 - \rho)$ (Figure 2.2 B). The value of ρ determines the rate of particle capture (uptake) by cells.

It should be noted that particle capture or uptake by a cell may involve complex biophysical and biochemical processes. These processes can be influenced by many factors, such as van der Waals force [100], particle surface charge effects [101], particle surface modification by corona formation [102–105], and molecular recognition by the receptor proteins in the cell membrane [106–109]. Explicit consideration of these different factors may be possible if quantitative information about their relative importance and molecular mechanisms of the recognition processes are known. Here, a simple approach is taken where the probability parameter ρ implicitly accounts for the lumped effects from the various factors that may influence particle capture by cells. For example, a particle with a small ρ in the model may represent a particle with a bare surface with a poor affinity for the cell membrane. On the other hand, a particle with a large ρ may represent a particle with a modified surface (functionalized with a targeting ligand, for example) with a high affinity for the cell membrane because of the molecular recognition by membrane proteins[101, 106–109].

2.1.7. Model Parameters Values. Table 2.1 lists the model parameters and their values. In the model, cells have a typical radius of 10 μm. Nanoparticles have a radius of 100 nm if a different size is not specified explicitly. Tissue porosity (Γ/Ω) is 0.60. The probability of particle capture per collision with a cell (ρ) is varied between 0.01 and 1. Physiological temperature (37 ^{o}C or 310 K) was used in the Einstein-Stokes equation to calculate particle diffusion. The remaining parameters, fluid viscosity (μ), entry fluid velocity (v_{in}), and regularization constant (ϵ) are based on [93]

Table 2.1. Model parameter values

Parameter	Value	Reference
Cell radius, r (μm)	10	This work
Nanoparticle radius, a (nm)	10–**100**	This work
Tissue porosity, α	0.6	This work
Particle capture probability, ρ	**0.01**–1	This work
Fluid viscosity, μ (cP)	2.5	[93]
Temperature, T_p (K)	310	
Entry fluid velocity, v_{in} (μm/s)	0.05–1	[93]
Regularization constant, ϵ (μm)	0.5	[93]

2.2. RESULTS

2.2.1. Size Effects of Nanoparticles in an In Vitro Cell-Free Tissue. In drug-delivery experiments, it is a common practice to employ cell-free tissue systems as a substitute of an in vivo physiological tissue. First, investigated particle size effects on the distribution and penetration of nanoparticles in such in vitro tissue systems. As mentioned previously, the experimental work of Wong et al. [56] studied the effects of particle size in a cell-free collagen matrix (Figure 2.4 A). In contrast, Tang et al. [57] investigated particle size effects in in vivo tumor tissues (Figure 2.4 B). The collagen matrix used in Wong et al. [56] was devoid of cells and advective transport. An experiment in the study compared the tissue distribution and penetration efficacy of 10 and 100 nm particles. Both particle sizes displayed a broad dispersion across the tissue system. However, the smaller particles revealed a significantly deeper penetration (Figure 2.4 A).

The experimental observation of Wong et al. [56] can be explained with a simple theoretical model. Comparing the tissue domain with a semi-infinite plane in one dimension, the solution of the following equation describes the time-dependent concentration profile (probability density function) of a single particle in the domain:

$$\frac{\partial G}{\partial t} = D\frac{\partial^2 G}{\partial^2 x} + \delta(x)\delta(t) \tag{2.6}$$

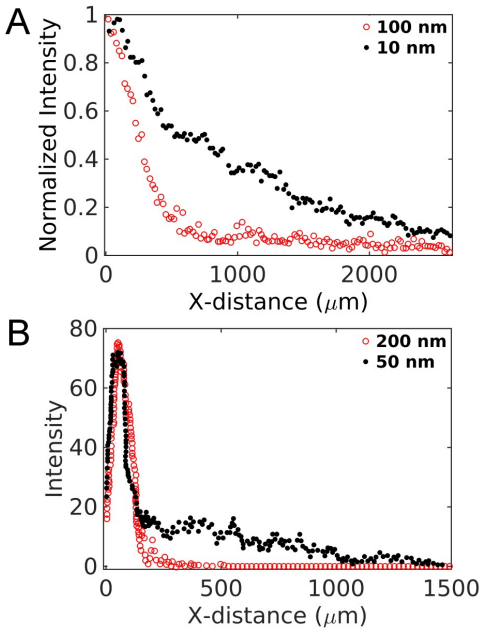


Figure 2.4. Experimental data adapted from two earlier works [56, 57]. (A) Data from Figure 3H of Wong et al. [56]. The figure compares distribution of 100 nm (red) and 10 nm (black) nanoparticles in collagen in an in vitro experiment. (B) Experimental data adapted from Figure 5d of Tang et al. [57]. The figure compares tumor tissue distribution of 200 nm (red) and 50 nm (black) particles in an in vivo experiment

where the source term (product of the Dirac delta functions) represents the initial particle location at the origin. D is the size-dependent diffusion coefficient (Equation 2.5). The solution of this equation is $G(x,t) = (1/\sqrt{(\pi Dt)})exp(-x^2/4Dt)$. The solution is similar to a Gaussian distribution in an infinite domain with the exception that the peak height is $1/\sqrt{(\pi Dt)}$ instead of the corresponding Gaussian peak $1/\sqrt{(4\pi Dt)}$, and the solution is valid only in the right half plane ($x \geq 0$). Figure 2.5 A represents this analytical solution for three different particle sizes. The diffusion coefficient of each particle size was calculated based on the Einstein-Stokes formula (Equation 2.5) and the physical properties of the interstitial fluid listed in Table 2.1. Figure 2.5 B shows corresponding results from the simulation for two different particle sizes (10 and 100 nm). The inset of Figure 2.5 B shows the normalized curves for a direct comparison with the fluorescence data in [56] (Figure 2.4 A).

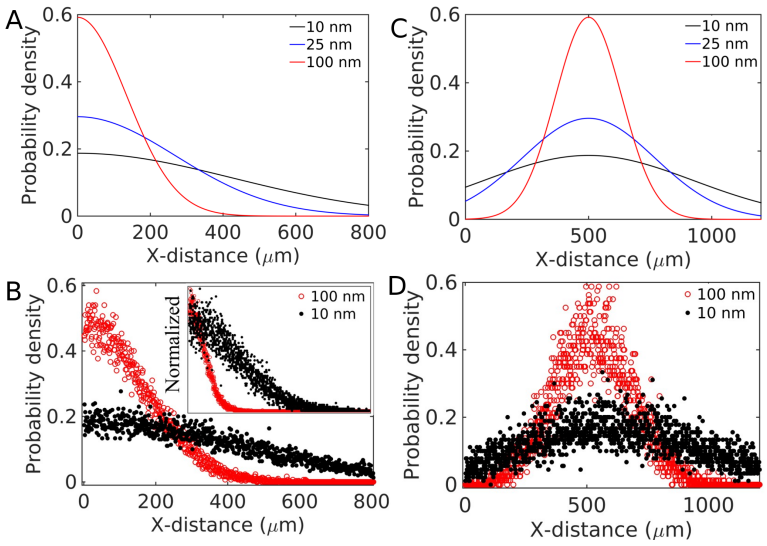


Figure 2.5. Particle size effects in a cell-free system. (A) Theoretical model (Equation 2.6) and (B) simulation considering pure diffusion. (C) Theoretical model (Equation 2.7) and (D) simulation considering a small advection (0.05 $\mu m/s$) and diffusion

In a biological tissue, however, it is unlikely to have a purely diffusive motion of particles. In the presence of a small flow (advection) to the right, particle distribution can be described by the following equation:

$$\frac{\partial G}{\partial t} = D\frac{\partial^2 G}{\partial^2 x} - v\frac{\partial G}{\partial x} + \delta(x)\delta(t) \tag{2.7}$$

where v is a constant velocity in the X-direction. The solution of this equation, $G(x,t) = (1/\sqrt{(\pi Dt)})exp(-(x-vt)^2/4Dt)$, is shown in Figure 2.5 C for $v = 0.05\ \mu m/s$. Corresponding simulation result is shown in Figure 2.5 D. The distribution peaks are shifted by a distance vt, as expected. Based on this result, in a cell-free system, it may take only few hundred seconds for a particle to travel tissue-scale distances (few hundred microns). Contrary to this, the in vivo distribution in Tang et al. [57] (Figure 2.5 B) clearly indicates that particles travel at a much slower pace in a physiological tissue condition perhaps because of the transport barriers imposed by the cells.

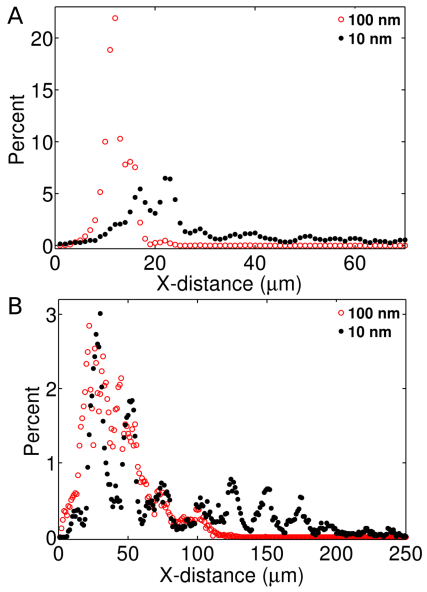


Figure 2.6. Predicted particle size effects in the presence of cells. The panels represent the following conditions: (A) pure diffusion and cells; (B) advection, diffusion, and cells. All simulations were carried out considering $\rho = 0.01$. The fluid velocity at the tissue entry (left edge) was assumed $v = 1\ \mu m/s$ [93]

2.2.2. Size Effects of Particles in In Vivo Tissue Conditions. Next, an investigated was performed on how particle size may impact the tissue distribution and penetration efficacy in a physiological tissue condition. It was interesitng to analysis the experimental result in the in vivo tumor tissue distributions reported in Tang et al. [57] (Figure 2.4 B). This in vivo data indicated a modestly deeper penetration by the smaller particle but the tissue distribution profiles of the particles were significantly different from those observed in the cell-free collagen sample in [56] (Figure 2.4 A). Both particles revealed narrow and overlapping peaks, suggesting a relatively poor tissue dispersion and penetration compared to the cell-free system. Two different scenarios were investigated in the presence of cells. In one case, only cells and diffusion were included but no advection (Figure 2.6 A). In the other case, cells, diffusion, and advection were included (Figure 2.6 B). This latter condition could be a more practical representation of a biological tissue.

Comparing Figure 2.6 with Figure 2.5, the presence of cells in the model had a dramatic effect on the penetration depth. The dispersion of both the 100 and 10 nm particles were significantly reduced under pure diffusion (Figure 2.6 A) as well as under advection and diffusion (Figure 2.6 B). The predicted distributions in Figure 2.6 B are qualitatively consistent with the experimental observations of Tang et al. [57]. Consistent with the experimental data, the model shows that the peaks of the 10 and 100 nm particle distributions align at the same location though the smaller particle distribution shows a tail stretched further to the right.

Comparing Figure 2.5 with Figure 2.6, a cell-free in vitro system may provide inaccurate information as to how the particle size affects the distribution and penetration of nanoparticles in biological tissues. Figure 2.5 indicates the 10 nm particles are significantly more efficient in tissue dispersion, consistent with the experiment of Wong et al. [56]. However, Figure 2.6 indicates the difference between the 10 and 100 nm particles may be less pronounced in a real tissue system, where particle motions could be hindered by their interaction with the cell boundaries.

The analysis above indicates that cell-surface adhesion and capture of particles may significantly compromise the particle size effects in in vivo physiological conditions. In a cell-free system, particle size effects could be more significant due to the unrestricted diffusion, which is directly determined by particle size. In contrast, in the presence of cells, diffusion plays a less significant role. Therefore, in vivo interstitial transport behavior of particles could be predominantly determined by the barriers imposed by the cell boundaries. Next the model was used to capture the experimental data of Tang et al. [57] (Figure 2.4 B). A direct fit between the model and the data was not possible due to the missing information on the exact experimental time frame and tissue properties, which include cell density and interstitial fluid properties, fluid velocity, and particle capture rate by cells. The system was simulated for 10,000 seconds and attempted to match the position of the distribution

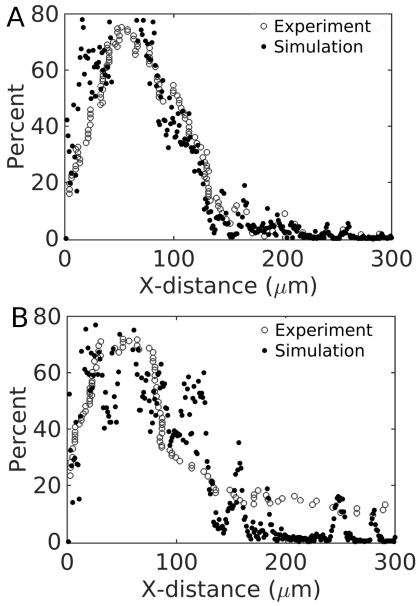


Figure 2.7. Comparison between simulation and experiment. The open circles represent the experimental data of Figure 2.4 B (plotted in a different scale). The filled circles represent simulation. (A) Particle size is 200 nm. (B) Particle size is 50 nm

peaks for the two particle sizes reported in [57]. The match between the simulation and data (Figure 2.7) required variations in the inlet fluid velocity (v in) and the probability of particle capture per collision (ρ), leading to v in = 4 $\mu m/s$ and ρ = 0.001.

The small value of ρ indicates that a particle gets captured after many contacts (collisions) with the cell boundary. At this range of ρ, it is found that the particle distribution profiles were less sensitive to the value of ρ in the simulations. The distributions were primarily determined by the fluid velocity and duration of the simulation. It should be noted that the parameter ρ does not capture the possibility of particle dissociation (reversible binding). Replacing this simple probabilistic construct based on ρ with more mechanistic details of particle uptake [110] and complementary quantitative experiments might shed light on particle uptake rate by cells in biological tissues.

2.2.3. Effects of Cellular Uptake Rate on Tissue Dispersion and Penetration. Previous analysis led to further investigate how cells influence the tissue distribution of particles. In Figure 2.8, the effects of ρ was investigated on the tissue penetration efficacy

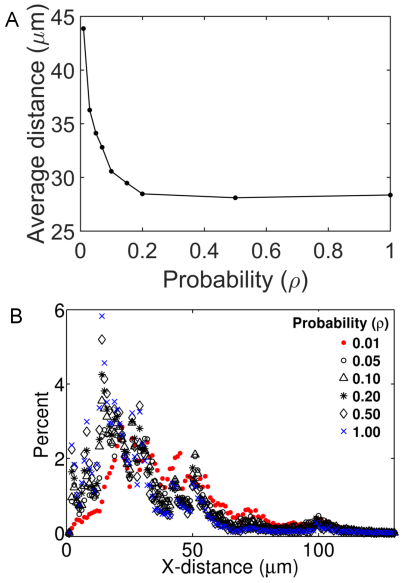


Figure 2.8. Predicted effects of cellular uptake rates on tissue distribution of nanoparticles. (A) Mean depth of tissue penetration by particles as a function of ρ. The mean depth of penetration represents the average of the horizontal positions (X-coordinate) of 16,000 simulated particles in 10^4 seconds after their tissue entry. (B) Histograms showing distribution of the nanoparticles. Each histogram corresponds to a different value of ρ, as indicated in the figure legend

of 100 nm nanoparticles. Figure 2.8 A shows the mean depth of penetration as a direct function ρ. Figure 2.8 B shows the tissue distribution profiles at different values of this parameter. As seen in the figures, the penetration depth and the distributions were insensitive in the range $0.1 < \rho < 1$. However, there was a noticeable change in the penetration depth and distributions in the range $\rho < 0.1$.

The results above indicate a non-linear relationship between the cellular uptake rate and tissue penetration depth. This nonlinearity could reflect the fact that the overall rate of cellular uptake is determined not only by ρ but also by the mean number of collisions a particle makes with the cell boundaries. If a particle on average makes C number of collisions with any cell boundary, the probability that it will get captured is ρC. As for example, with $\rho = 0.1$ and $C \geq 10$, particle can get captured with probability 1 upon its encounter with a cell. Therefore, a further increase in ρ beyond 0.1 could have little impact

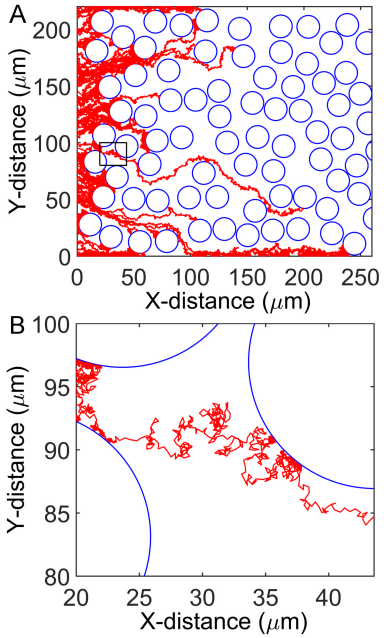


Figure 2.9. Representative travel paths of simulated nanoparticles. (A) Travel paths of 100 nanoparticles in the tissue domain. The particles are of identical size (100 nm radius). (B) A zoomed-in view showing a single particle travel path and its interaction with a cell boundary. Panel B corresponds to the small region in Panel A marked by a rectangle

on the overall capture rate. This adaptive algorithm takes fine-resolution time step (δt) near the solid (cell) boundaries, as discussed in Materials and Methods. As illustrated in Figure 2.9, the fine resolution $\delta t = 10^3$ second near the cell boundaries allows a particle to make many collisions with a cell before it gets captured. Therefore, the actual rate of cellular uptake could be high even though ρ is small. In the simulations, the default value of ρ is 0.01 (Table 2.1).

2.2.4. Model Prediction Sensitivity to Time Steps. Because the adaptive algorithm selects time steps over a wide range ($\Delta t_m = 0.1 < \Delta t < \delta t = 10^3 s$), it was investigated that if the predictions in Figure 2.10 could be sensitive to the selection of time steps. Therefore, the upper bound (Δt_m) was varied in the range 10^3 s to 0.1 s to enforce different resolution of time steps in the simulation algorithm. For each Δt_m , 16,000 nanoparticles were simulated for 10^4 s and then calculated the mean depth of tissue penetration by these particles. This analysis was carried out for different values of ρ. Corresponding plots are

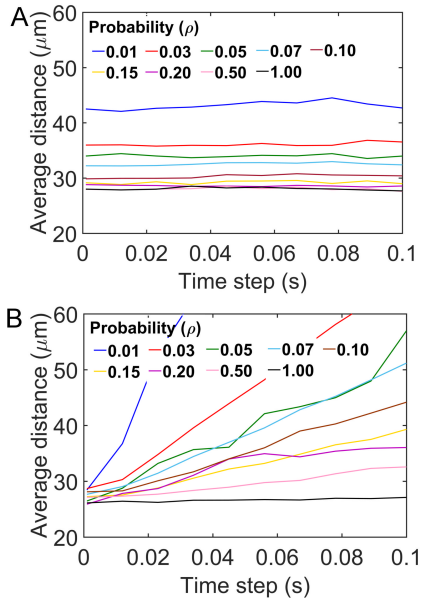


Figure 2.10. Effect of time step Δt_m variation on model predictions. (A) Average tissue penetration by particles as a function of Δt_m . Each curve corresponds to a different value of ρ (the probability of particle capture by a cell in a collision between the particle and the cell.) (B) The analysis of Panel A is repeated using a non-adaptive BD algorithm based on Rejniak et al. [93]

provided in Figure 2.10 A. As seen in the figure, the predictions remained insensitive to the Δt_m . This robustness reflects the fact that the algorithm adapts to smaller steps when particles are in close proximity to the cell boundaries regardless of the value of Δt_m.

However, it is important to note that Δt_m cannot be assigned an arbitrarily large value. A smaller Δt_m is needed to approximate particle velocities to the local fluid velocity. A large Δt_m enables the particles to advance with large steps. As a result, local velocity fields before and after the jump could be significantly different, thus introducing larger inaccuracies in the velocity approximation for the particles.

In Figure 2.10 B, the same analysis was performed using a non-adaptive algorithm, where the time step size was kept constant. This fixed time-step algorithm is similar to the algorithm of [93]. Contrary to this approach, the algorithm of Rejniak et al. [93], however, treated particles (drug molecules) as point objects. The algorithm moved the particles based on a fixed time step and rejected the moves in case of a conflict with the cell positions.

The algorithm also assumed an interaction layer of 0.25 μm around each cell periphery. A particle was considered captured by a cell immediately upon its arrival within the 0.25 μm interaction layer. These features were taken from the Rejniak model with the following exceptions: 1) Instead of treating the particles as points, the particles were treated as circular objects of 100nm radius, as in this model; and 2) Instead of assuming an immediate particle capture within the interaction layer, a capture probability was incorporated between $0 \leq \rho \leq 1$ in the layer. The predictions made by this algorithm at different selections of the time step size and ρ are shown in Figure 2.10 B. Clearly, the predictions were sensitive to the choice of the step size. This sensitivity is expected because the rate of particle capture by cells in this algorithm should depend on the thickness of the interaction layer and the relative choice of the time step size. For a thinner interaction layer, a particle would be less likely to hit the layer if advanced based on a fixed step. Similarly, an increase in the time step size would also reduce the possibility of hitting the interaction layer. Thus, the fixed time step algorithm should underestimate the rate of particle capture (cellular uptake) and overestimate the tissue penetration depth if a smaller interaction layer or larger time step is chosen. Moreover, due to the fixed (and large) time step size in the algorithm of Rejniak et al. [93], many particle moves might be rejected due to the conflicts with the cell positions.

As mentioned before, $\delta t = 10^3$ represents the smallest time step in this model. Because of the no-slip boundary condition, particle motion near a cell boundary is primarily driven by diffusion. Thus, the length of a particle jump near a cell boundary can be estimated based on pure diffusion: $\mid S \mid \approx \mid S_d \mid = \sqrt{4D\delta t}$. For the fluid properties and temperature listed in Table 2.1, this jump size becomes comparable to the size of the particle. Therefore, it is a sufficiently small step size to capture the fine resolution details of interactions occurring at the particle-cell interface.

2.3. DISCUSSION

In this work, a multiscale Brownian Dynamics algorithm to study particle transport behavior in biological tissues is developed. Using the approach, particle size effects on tissue distribution and penetration reported in two experimental studies were investigated. The analysis was focused on how these behaviors may vary in cell-free artificial tissue systems and in vivo tissue conditions.

This multiscale algorithm can be generally applicable to modeling advection-diffusion systems involving heterogeneous porous media. The approach implemented here is inspired by two previous modeling works [93, 111]. Earlier, Monine et al. [111] developed a time-adaptive Brownian Dynamics (BD) algorithm to study enzyme-substrate reaction in the plasma membrane of cells. Recently, Rejniak et al. [93] used the Method of Regularized Stokeslets (MRS) [55] to study drug molecule transport in biological tissues. Both these models treated the mobile particles (substrate and drug molecules, respectively) as point particles while considering their stationary reaction or binding partners (enzyme molecules and cells, respectively) as circular objects. In this model, the time-adaptive feature of the Monine model was combined with the MRS. This combination enabled multiscale modeling of particle transport under both advection and diffusion while capturing high-resolution details of particle interaction with the cell boundaries. Contrary to the point particle assumption in the Monine model and Rejniak model, the mobile nanoparticles were considered as spherical objects occupying space in the two-dimensional membrane.

Contrary to the general perception, this study revealed less significant effects of particle size on their intra-tissue distribution and penetration. This analysis shows that in vitro tissue systems, being devoid of cells and convective flow, may result in misleading conclusions regarding the transport behavior of particles in the biological tissues. Here, the focus was limited to particle size only. However, the multiscale algorithm can be extended to

incorporate other design attributes of particles, such as geometry and surface ligands. This extension will allow mechanistic interrogation of how these parameters affect the transport behavior of particles in biological tissues.

In the model, the nanoparticles were treated as mutually non-interacting objects and the particles do not collide or form aggregates. This consideration is based on the assumption that physiological tissue concentrations of drug-delivery nanoparticles are small. Apparently, there is no report on the mutual interactions of drug-delivery nanoparticles in the physiological tissue conditions. It has been reported that 1% of intravenously injected particles can reach the target tissue [112, 113]. Therefore, from the injection of 1 ml solution containing 100 $million particles/ml$ [114], only a 1 million particles are expected to reach the target tissues. Thus, for 100 nm radius particles, the estimated volume fraction of particles in the target tissues could be in the order of 10^9 assuming $1cm^3$ of tumor tissue volume (a single tumor or many smaller tumors). At this volume fraction, their non-specific collision is unlikely or less important considering many other cellular proteins and biomolecules that could present at comparable amounts.

This model does not consider the effects arising from the surface charges of particles or van der Waals forces acting between a particle and a cell. Moreover, in a body fluid, soluble biomolecules may interact with nanoparticles and form a coating or biocorona over the particle surface [102–105]. Formation of biocorona modifies the surface properties of particles. At present, the quantitative aspects of biocorona formation and how it modifies the particle surface properties and tissue interaction are not well-understood. Therefore, rather than explicitly incorporating these other properties (van der Waals and biocorona effects), a phenomenological parameter ρ was used in the model that accounts for a lumped measure of the affinity of interaction between a nanoparticle and a cell. Nevertheless, for a quantitative understanding of these other phenomena influencing tissue interactions of particles, it is crucial to explicitly address them in a mechanistic model. The Brownian Dynamics-based framework presented here could serve as an initial platform towards this

direction. The framework could be extended to capture these other types of particle- and tissue-specific physicochemical parameters. Integration of such predictive mechanistic models with complimentary experiments could be essential for a quantitative elucidation of these other effects on drug delivery nanoparticles in biological tissues [115].

Nanoparticle velocity was considered to be the same as the local fluid velocity while ignoring the influence of the particles on the velocity field. It is possible that large particles also modify the local velocity fields at the micro scale. However, nanoparticles are of the same dimension as many cellular proteins, biomolecules, and solute particle. This model is based on existing models where nanoparticles velocities were considered to be the same as fluid velocities in the porous media [116–119].

This modeling approach may be expanded for spatiotemporal modeling biochemical network systems. The rule-based modeling (RBM) approach [65, 120, 121] provides unique capability to model biochemical network systems by taking into account the coarse-grained structural details of protein molecules [122, 123]. However, most of the early RBM tools were developed aiming at non-spatial modeling. Recently, the RBM tools Kappa [124], Simmune [63], and BioNetGen [125] are being added with new capabilities for spatiotemporal modeling. The molecular dynamics (MD) simulation is used to model protein structures with atomistic details [126]. But MD can deal with very short time scales, and not scalable for biochemical network modeling considering a large number of species and their structural details.

2.4. REMARKS

In this Section, a multiscale spatiotemporal model is developed to address the challenges arising from multiscale nature in the extracellular domain of tissue architecture. This multiscale simulation method is robust for mechanistic modeling of particle transport in porous media. By combining a new time-adaptive BD simulation algorithm with the Method of Regularized Stokeslets (MRS), this method provides a unique capability to model

particle transport considering particle size and particle-cell interactions in heterogeneous biological tissue. Using the approach, particle size effects on their distribution and penetration in biological tissues have been investigated. Contrary to the general perception, this work shows that particle size may play a less significant role in particle transport in the physiological tissue conditions. In the presence of cells, the effects arising from the difference in particle size is small. Particle-cell interactions primarily determine particle penetration and distribution. This study underscores the roles of advective transport and cells that are often ignored in artificial tissue systems of in vitro experiments. This scalable model can be used to study the extracellular dynamics of nanoparticles as well as other signaling molecules in the cellular environment. Based on the distribution of these nanoparticles and signaling molecules cell will show their collective behavior. In the next Section, a multiscale model is developed to study the collective behavior of cells.

3. MULTICELLULAR MODELS BRIDGING INTRACELLULAR BIOCHEMISTRY TO POPULATION DYNAMICS

The multiscale nature of the cell population dynamics in a tissue architecture is another challenge in multiscale modeling of a biological system. Based on the molecular interactions in the signaling pathways or reaction networks, the dynamic behavior of cell population changes. The interconnectivity in signaling pathways at a fundamental scale make it difficult to understand the behavior in tissue scale. In this Section, a framework for the systematic development of multiscale cell population models is introduced. The dynamics of the cell population are incorporated with a wide range of space and time. Cell signaling and gene transcription occur at a molecular scale and faster time scales. But the collective behavior of cells (e.g., death, division, proliferation) occurs at a macroscopic level and longer time scales. The rate of gene transcription and production of the molecule in intracellular space depends on the signaling molecules in the extracellular space. The dynamics of the cells in a tissue are connected via the multicellular communication through environment [127–130]. Bridging these multiscale events are computationally challenging. The phenomena in tissue scale are mostly phenomenological rather than physiological. Deducing a fundamental mechanics is difficult due to the lack of knowledge of physiological property and mechanism.

Here, a parallel framework is developed to create a multiscale cell population model from a single cell biochemical network. In this Bottom-up approach, signaling pathways or reactions network is selected as a fundamental level. The method is based on Master slave communication where the Master process launches parallel processes. The parallel processes act as slave processes and simulate the cell-specific biochemical networks based on the Gillespie algorithm [38]. Then the Master process communicates with each cell processes after a fixed interval of time and modifies the cellular environment to capture

tissue scale dynamics. The Master process also simulates the collective behavior of the cell population. This multiscale framework is capable of modeling cell to cell and cell to environment communications.

In the framework, model-specific higher level rules link the intracellular molecular events to cellular functions, such as death, division, or phenotype change. Cell death is implemented by terminating a parallel process, while cell division is carried out by creating a new process (daughter cell) from an existing one (mother cell). First, these capabilities are demonstrated by creating two simple example models. In one model, a relatively simple scenario is considered where cells can evolve independently. In the other model, the interdependency among the cells is considered, where cellular communication determines their collective behavior and evolution under a temporally evolving growth condition. The dynamics of cell population due to the presence of environmental signaling molecule or apoptotic molecule is also demonstrated here by an example model. This mock model is similar to real biological system where cell death occur due to apoptotic molecules (Bcl-2, Fas, FasL, Caspase-8, P52 [131, 132]) or external factors (e.g. drug dosages [133–135]). This framework is also capable of modeling the cellular death due to a constant or periodic supply of different dosages of drug molecules or apoptotic molecules. It is a general framework to incorporate any biological processes for a large population of cells (i.e., tissue scale). The software version of the framework is ParCell and also discussed here.

This Section is organized as follows. Section 3.1 describes the construction of the multiscale parallel framework. Section 3.2 demonstrate the capability of the framework based on the two mock models. Section 3.3 discusses the current limitation and future perspective of the framework. Section 3.4 concluded the Section with remarks.

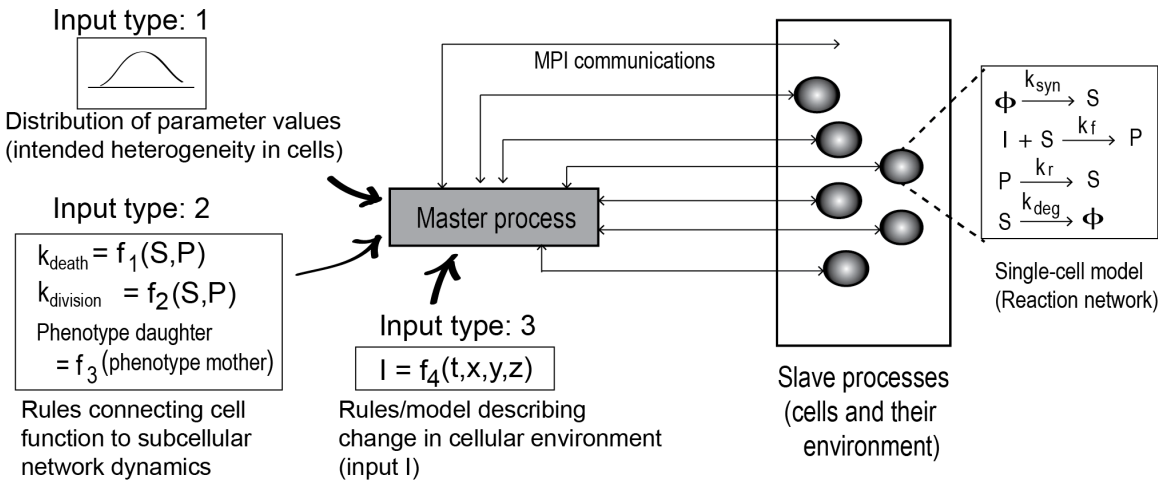


Figure 3.1. Schematic diagram of the population modeling framework. The framework uses MPI/multiprocessing parallelism to create multiscale population model from a single-cell biochemical reaction network model. Each single cell is represented by a stand-alone parallel process. Intercellular communication is mediated in a server-client fashion. On the far right of the figure, an example set of biochemical network reactions associated with each cell is shown. The network consists of two intracellular species S and P and their synthesis, degradation, and inter-conversion in response to an extracellular cue I. The master process creates a population of cells, where each cell has this same set of biochemical reactions. However, the master process incorporates heterogeneity among the cells by sampling the protein copy number or parameter values from defined distributions (Input 1). In addition, the master process implements rules for cellular decision making based on the intracellular species concentrations (Input 2). The master process also determines how the extracellular input I may change over time (t) or position (x, y, z) based on some defined rules (Input 3)

3.1. METHODS

3.1.1. The Population Modeling Framework. The framework is illustrated in Figure 3.1. To create a cell population model, the framework requires a single-cell biochemical network model implementing a stochastic method, such as Gillespie algorithm [38].

The biochemical network model could be a signaling pathway or gene transcription network model describing the intracellular protein-protein interaction and their biochemical transformations. The framework also requires three additional inputs, as shown. The Type 1 inputs are cellular distributions of the model parameter values and protein copy numbers. Distribution functions can be defined for the parameters and initial concentrations

to incorporate cell-to-cell variability in the population model. The Type 2 inputs are model-specific rules that link the temporal intracellular events (time evolution of a network species, for example) to cellular functions (death, division, or phenotypes). The Type 3 inputs define the cell environment (growth condition).

Given the inputs, the framework launches parallel simulations on the single cell biochemical network model. Each of the simulation processes is then treated as a cell object, as in an agent-based model. Based on the parameter distribution functions, each cell object may acquire properties that are distinct from the other cell objects in the population.

The framework works as a server (a master process), while each of the parallel simulations on the biochemical network model serves as a client (a slave process). MPI [82] or multiprocessing [83] is used to mediate the server-client communications. A detailed algorithm of this server-client scheme is provided in Appendix A. This sever-client scheme is illustrated in Figure 3.2 in case of MPI. Each cell object (slave process) is allowed to propagate simulation (Gillespie algorithm on the intracellular network) for a prescribed time interval Δt. At the end of each Δt, the master process and the slave processes communicate. The master process collects simulation data from each slave process (cell object) and analyzes the data against the rules (Type 2 and Type 3 inputs). Based on the analysis, it sends cell-specific instructions to each slave process. In addition, it takes actions to execute death, division, or other functions for a cell object if corresponding conditions specified in the rules are met.

If the internal state of any cell object meets the conditions for cell death (specified in the Type 2 rules), the master process terminates the parallel process. Similarly, if the internal state of a cell object meets the conditions for cell division, the master launches a new parallel process to create a daughter cell object. It then sends an message to the newly created process to specify its initial condition, as defined in the rules (Type 2 inputs). Following cell division, the framework partitions the contents (network species) of the mother cell between the mother and the daughter based on binomial distribution [136].

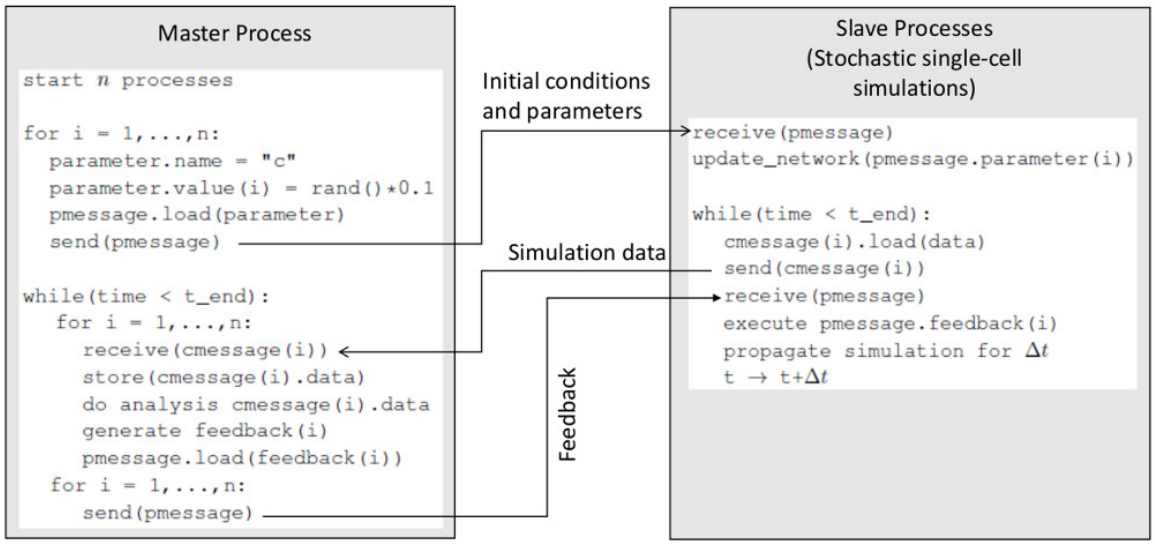


Figure 3.2. The skeleton algorithm of MPI communications. The master program and the slave program are shown in the left and right box, respectively. The master program defines parameters and rules associated with the cellular-scale processes and extracellular environment. The slave program defines the molecular-scale processes in the form of a complete biochemical network model. The master and slave programs contain complementary calls in a loop to mediate communications in a server-client fashion

Based on the Type 3 inputs, the master process evaluates the extracellular environment in each Δt. It sends messages to the cell objects about the updated environment variables. Such variables may include the concentration of an extracellular signal (the concentration of a drug, ligand, or other biomolecules, for example) that may affect the intracellular reactions. Based on the updated environment, the cell objects propagate simulation for another Δt, and the process can be repeated. A software named ParCell is developed based on the scheme discussed here.

It should be noted that the algorithm above is based on an approximation, rather than the accurate implementation of the Gillespie method. The approximation involves discretizing time in small intervals (Δt) within which the cells are assumed independent (simulated in parallel). An accurate implementation of the Gillespie algorithm over a multicellular system could be impractical if the detailed molecular interactions are considered at the single-cell level.

The simulation time is discretized with $\Delta t = 1$ second. The assumption here is that the change in the extracellular environment in this period is small. In addition, this period is also small compared to the time scale of the cellular processes (death and division), which typically take several minutes to hours. This approximation may potentially improve computation, as demonstrated in the result section. In this algorithm, the time step Δt can be chosen arbitrarily small. However, a smaller Δt could increase the frequency and overhead of the communication.

3.1.2. Gillespie Algorithm. The Gillespie Algorithm, also called Stochastic Simulation Algorithm (SSA), is a procedure for simulating changes in the molecular concentration of chemical species in a chemically reacting system. Hence the behavior of each cell and the advancement of simulation time within the system is determined by executing the Gillespie algorithm. The Gillespie algorithm (shown below) calculates *propensity* (likelihood) a_i of reaction γ_i, and the *total propensity* A as follows:

$$a_i = k_i \times \prod_{j=1}^{|\omega_i|} \omega_i^j \tag{3.1}$$

$$A = \sum_{i \in \gamma} a_i \tag{3.2}$$

Here, k is the rate constant and ω is the reactant concentration.

procedure GILLESPIE(R_r,R_c,M,t)
    Calculate a_i for all reactions and A.
    Select $r_1, r_2 = random(0, 1)$
    Update current time t: $t = t + ln(1/r_1)/A$
    Select reaction γ_j : $(\exists J \in \mathbb{N}) \sum_{j=1}^{J} a_j < A \times r_2 \leq \sum_{j=1}^{J+1} a_j$
    Update reactant concentration of γ_j in M.
    Return t, M
end procedure

In each step, the Gillespie algorithm takes four input parameters: the set of reaction rules (R_r), reaction rate constant (R_c), initial simulation time t and molecular concentration matrix M. The Gillespie algorithm probabilistically chooses a single reaction (γ_j), based on the individual reaction propensities. Finally, Gillespie updates the reactant concentration of γ_j according to its stoichiometric coefficients.

3.1.3. Sequential Algorithm. In the sequential model, each cell contains one copy of the regulatory network. All the reactions in the entire population, including diffusion reaction between cell and environment, is considered as a single global system. As shown in Line 4 of sequential algorithm, the Gillespie algorithm is invoked in a loop, until the current time t exceeds the total simulation time T. Note that the sequential model, when used in conjunction with the exact Gillespie algorithm, yields accurate result. Therefore, the sequential model is implementedas a benchmark of accuracy for our proposed parallel framework.

```
1: procedure SEQUENTIAL(R_r,R_c,M,T)
2:     t = 0
3:     While t ≤ T do
4:         t, M = Gillespie(R_r,R_c,M,t)
5:     Endwhile
6: end procedure
```

3.1.4. ParCell: Software Implementation of the Framework. ParCell is a Master-Slave based parallel software considering each cell as a parallel process or agent. Figure 3.3 shows a schematic diagram of ParCell model generator. It takes the reaction network of a single cell (e.g.: DNA transcription as shown in the figure) and expands the model into population scale. The software incorporates the parameter, species, initial molecular concentration and reaction rules in different blocks. These blocks are similar to the software BioNetGen [137] which focuses on the modeling of a single cell. But ParCell can incorporate cellular heterogeneity by distributing the rate constant parameter across all cells. Uniform, normal and lognormal distribution are available in the software. Dependent

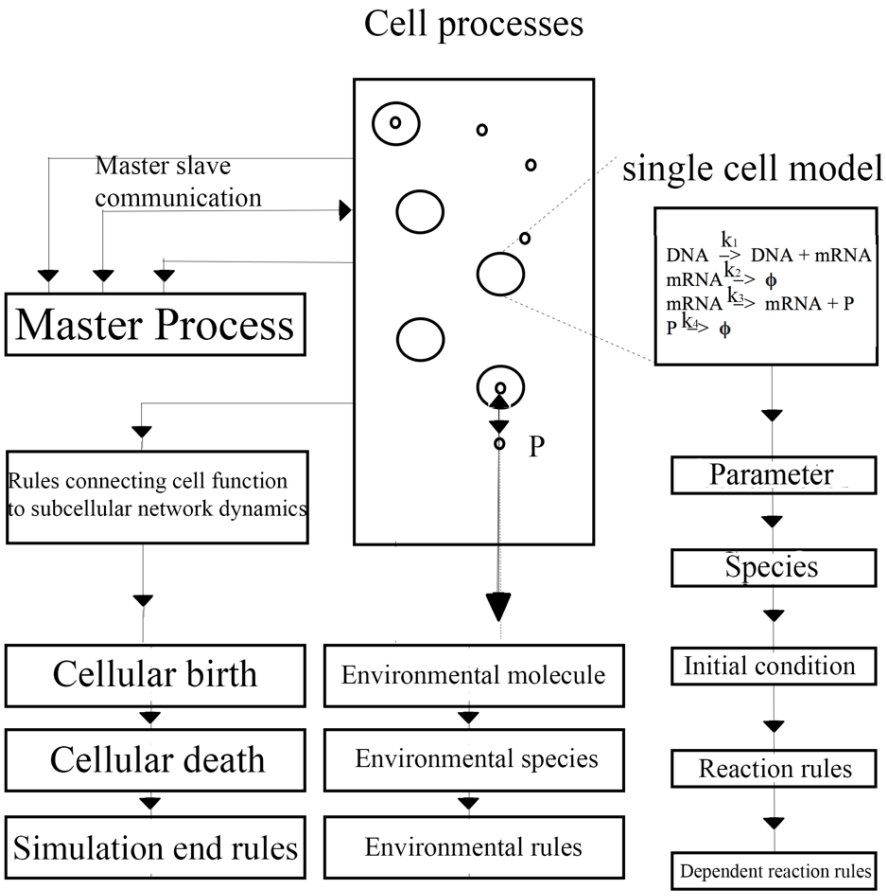


Figure 3.3. Schematic diagram of ParCell model generator

reaction rule block can include any dynamic dependency of the reaction rules. For example, if the rate constant parameter of a protein depends on the dynamic concentration of another species, the time-dependent concentration can be included in the rules. The environmental molecules are represented by environmental species block. It automatically synchronizes the environmental molecule across the system boundary homogeneously. Environmental rules can set an environmental species concentration to a fixed value which represents the external influx. Rules connecting cellular birth and death can be included in the cellular birth and death block. Simulation end rules block is used when a user wants to terminate the simulation after crossing a certain threshold of the total number of cells.

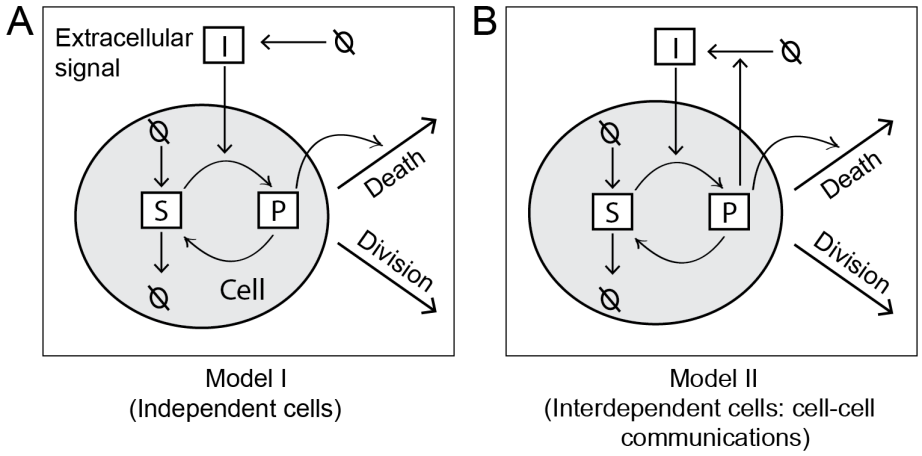


Figure 3.4. Models created using the framework. The gray circle represents the cell, and the white space around the circle represents the extracellular space. (A) Model I: cells are independent. (B) Model II: cells communicate by modifying the extracellular environment

The software is implemented in two example models: DNA transcription model and michaelis menten model. The formulation of the model in ParCell and associated results are shown in Appendix A. A more realistic biological system for bacterial quorum sensing and its implementation in ParCell is discussed in next Section.

3.2. RESULTS

3.2.1. Models. Two models (Model I and Model II) are created using the framework. The models are illustrated in Figure 3.4.

3.2.1.1. Model I. In this model, cells are considered as independent objects. Therefore, the death, division, or intracellular reactions of any cell is not affected by other cells in the population.

First, a biochemical network model describing the intracellular events at the molecule scale is defined. As illustrated in Figure 3.4 A, four elementary reactions are considered in the network. These reactions and corresponding rate constants are listed in Table 3.1. The reactions describe a signaling pathway, which is activated in response to an extracellular signal I. A receptor protein S is synthesized and degraded in a zeroth and first-order

reaction, respectively. In the presence of extracellular I, S is activated to produce P and deactivated in a reverse reaction. In the model, P serves as a pro-death signal for a cell. Intracellular accumulation of P elicits cell death by inducing the mean rate of cell death k_{death}.

Table 3.1. Signaling pathway model parameters (slave program)

Parameter	Value	Description
S	10,000 molecule/cell	Initial concentration of S.
P	0 molecule/cell	Initial concentration of P.
I	(Varied) molecule/cell	Extracellular signal.
k_{syn}	10 molecules s^{-1}	Rate constant for $\phi \rightarrow$ S.
k_{deg}	0.001 s^{-1}	Rate constant for $S \rightarrow \phi$
k_f	10^{-7} (molecule/cell)$^{-1}s^{-1}$	Rate constant for $S \rightarrow$ P.
k_r	0.1 s^{-1}	Rate constant for $P \rightarrow S$.

Table 3.2. Cellular model parameters (master program)

Parameter	Value	Description
$k_{death,0}$	10^{-3} s^{-1}	Mean rate of cell death in the basal state.
$k_{division}$	10^{-3} s^{-1}	Mean rate of cell division.
α	10^{-6} (molecule)$^{-1}s^{-1}$	Cell death sensitivity to intracellular P.
k_{death}	$(k_{death,0} + \alpha P)$ s^{-1}	Mean rate of cell death (function of intracellular P).
k_e	10^{-9} (molecule)$^{-1}s^{-1}$	Model II only: rate of cellular production and release I per of molecule of intracellular P.

In a separate program, the cellular processes and associated parameters are introduced (Table 3.2). Cellular heterogeneity is ignored (Type 1 input). A rule (Type 2 input) is included to define the dependency of cell death on intracellular P. This rule simply states a linear relationship: $k_{death} = k_{death,0} + \alpha P$ (Table 3.2). Here, $k_{death,0}$ is the mean rate of cell death in the absence of P or extracellular I. The proportionality constant α (Table 3.2) measures the sensitivity of a cell to intracellular P. A second rule is included to binomially distribute the contents (network species) of a cell when it is divided to create a new cell. The extracellular environment (I) is assumed to remain unchanged throughout the simulation.

As explained earlier, there are two programs: one program (acting as slave) defining the biochemical network model, and the other program (acting as master) defining the cellular processes. These two programs contain complementary calls to communicate in a server-client fashion. The master program launches a number of parallel simulations on the slave program to create the initial population of cell objects. For an interval, $\Delta t = 1$ seconds, all the simulation processes (cell objects) propagate simulations (Gillespie algorithm) on the biochemical network system. At the end of Δt, the master process collects simulation data from the cell objects. Based on the analysis of the simulation data against the rules, the master process performs few checks and takes actions accordingly. First, it evaluates each cell object for death or division. A cell is selected for death if $U(0,1) < 1 - e^{(-k_{death}\Delta t)}$, where $U(0,1)$ is a uniform random number between 0 and 1. Similarly, a cell is selected for division if $U(0,1) < 1 - e^{(-k_{division}\Delta t)}$. Cell death is executed by terminating the corresponding parallel process, while cell division is executed by creating a new process (daughter cell) from the existing process (mother cell). The master process then analyzes the level of P in each (living) cell and updates the cell's k_{death} according to the rule.

While the master process performs these checks, the slave processes (cell objects) remain in a blocking mode. This blocking mode is released upon receiving a message from the master program. The slave processes then propagate simulation for another Δt, and the process is repeated until the simulation end time is reached.

3.2.1.2. Model II. This model is shown in Figure 3.4 B. The model has the same set of intracellular reactions as Model I. However, in this model, cells are interdependent due to cell-to-cell communication. A reaction is considered whereby cells can secrete I in proportion to their intracellular P. Thus, each cell contributes to the global pool of I in the environment, which is shared by all cells. An increase in I in the environment further stimulates the intracellular production P in the cells, thus inducing cell death.

The slave program of Model II defining the biochemical network model remains essentially the same as in Model I, with the exception of the master program, where an additional rule (Type 3 input) is included. According to this rule, a cell i secretes $\Delta I^{(i)}$ amount of I in time Δt: $\Delta I^{(i)} = k_e P^{(i)} \Delta t$, where k_e is a constant (Table 3.2) and $P^{(i)}$ is the intracellular P at time t. The total change of the environment is computed by summing up the contributions from all cells: $\sum_{i=1}^{n(t)} \Delta I^{(i)}$, where $n(t)$ is the number of cells at time t.

As in Model I, the master process communicates with the slave processes after each interval, $\Delta t = 1$ second, and evaluates cell death, division, and k_{death} for each of the cell objects. In addition, it also evaluates the change in I in the environment. It then updates the slave processes (cell objects) about the change in the environment by sending messages. Upon receiving the message, the cell object replaces the old I with a new I. Each cell object then propagates simulation for Δt based on the new extracellular environment. The process is repeated until the simulation end time is reached.

3.2.2. Sequential Model. This model is equivalent to Model I but implements accurate Gillespie method without parallelism. The model is developed using the agent-based modeling approach, where the cells are represented by agents. The entire multicellular system is represented by a single reaction list. The list contains the intracellular reactions of all cells in the system. In addition, it includes cell death and division as reactions. Therefore, with $n(t)$ cells in the system, the list contains $4n(t)$ intracellular reactions, $n(t)$ cell death reactions, and $n(t)$ cell division reactions. Gillespie algorithm is used to probabilistically select and execute reactions from the list. The same criteria as in Model I is used for cell death and division. Cell death is executed by removing an agent, whereas cell division is performed by creating a new agent in the system. In a cell division, cellular contents are again distributed based on binomial distribution, as in Model I.

3.2.3. Scalability and Performance. First, Model I is tested for its computational performance. The test is performed against an equivalent model (referred to as Sequential Model, please see Method section) implementing serial computation.

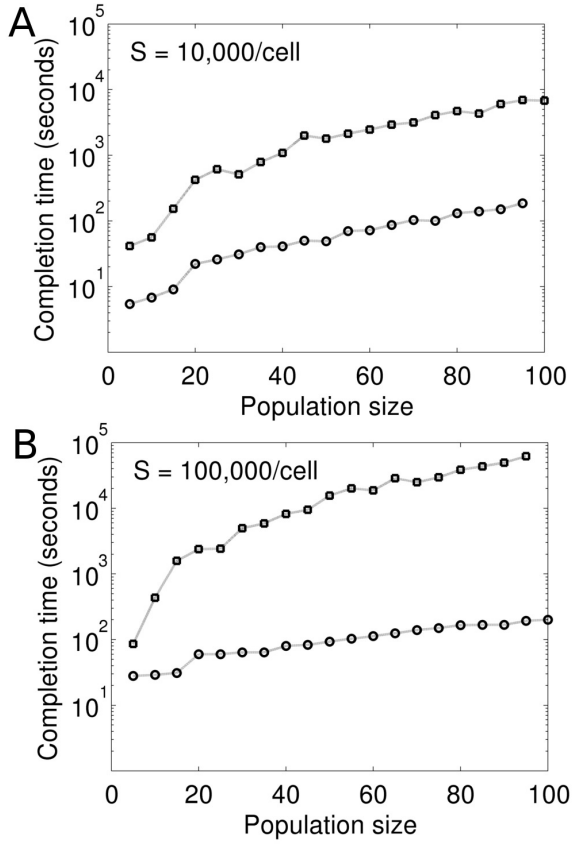


Figure 3.5. Computational performance of Model I. Model I (circles) and its equivalent sequential model (squares) are simulated for 1,000 seconds. The X-axis represents the average cell population size in a simulation. The Y-axis represents corresponding simulation completion time (clock seconds). Simulations are conducted in a dedicated 20-core computer

Figure 3.5 compares these two models in terms of their scalability and speed of computation. The scale of the model is varied by changing the cell population size. For consistency, all simulations are performed in the absence of stimulation ($I = 0$). This is to ensure that the mean cell population size remains constant during the simulations. For each population size, the two models are simulated for $1,000$ seconds and note the simulation completion times (wall clock seconds) (Figure 3.5). The simulations are carried out on a dedicated 20-core machine.

As seen in Figure 3.5, computation is orders of magnitude faster in the parallel formulation compared to its sequential counterpart. The difference becomes more significant when the models are made more computationally expensive by increasing the concentration of molecule S (Figure 3.5). Figure 3.5 B represents a 10-fold higher S in the models compared to Figure 3.5 A. Simulation using Gillespie method at the cellular level become more expensive when S is increased. This makes computation prohibitively slow in the sequential model.

The faster computation in the parallel formulation is due to the parallelism as well as the numerical approximation in the algorithm. As explained in the method section, the simulation time is discretized in $\Delta t = 1$ second. In each Δt, the cell objects are simulated independently and in parallel. At the end of each Δt, the cell objects are synchronized and updated. On the contrary, in the sequential model, Gillespie algorithm is accurately implemented over a single reaction list that represents the entire multicellular system. The sequential model is not scalable as the number of reactions in the list is determined by the number of cells in the system. In the parallel formulation, the time discretization ($\Delta t = 1$ second) is small because the mean waiting time for cell death and division is much longer ($1/k_{death} = 1/k_{division} = 1,000$ seconds). Some loss of accuracy is expected from this approximation, which, however, significantly improves computation.

3.2.4. Link between Intracellular Dynamics to Population Response. The rate of cell death in Model I is linked to the intracellular concentration of P (activated S) by a linear function, as discussed in the Method section. Figure 3.6 shows intracellular S and P of individual cells in an evolving cell population. An initial population of 100 cells is launched (parallel processes) at time zero. At $t = 1,000$ seconds, the population is stimulated with $k_f I = 0.01$ s^{-1} (Figure 3.6 A and B) and $k_f I = 0.1$ s^{-1} (Figure 3.6 C and D). In Figure 3.6, each curve for S or P represents an individual cell. The start and end of a curve correspond to the birth and death time of a cell, respectively. The sharp fall of any curve marks a cell division. The fall is due to the decrease in the cellular content of a

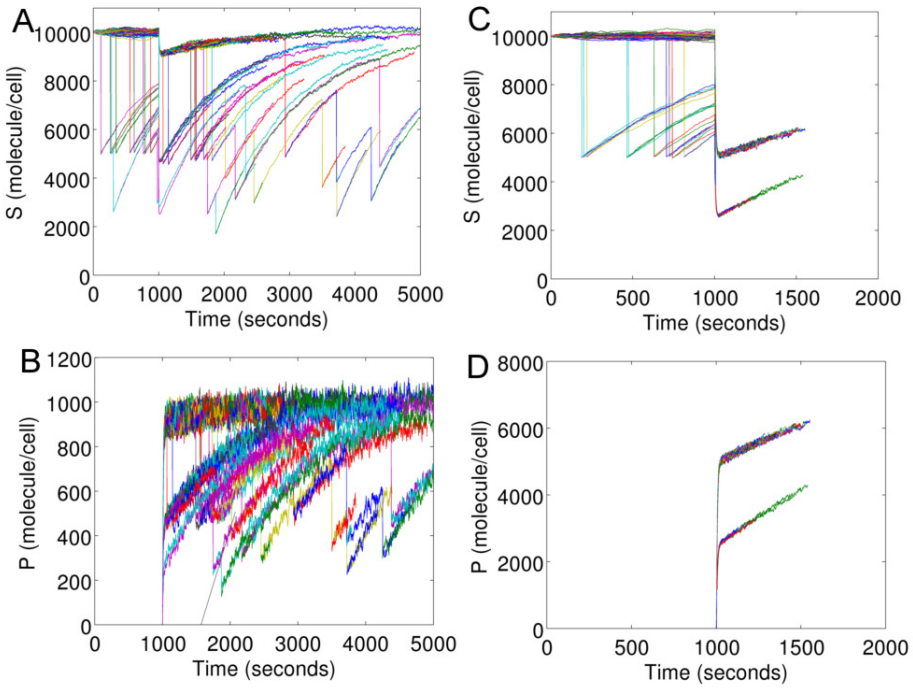


Figure 3.6. Temporal evolution of S and P in individual cells.(A), (B) The population is stimulated at $t = 1,000$ seconds with $k_f I = 0.01$. (C), (D) The population is stimulated at $t = 1,000$ seconds with $k_f I = 0.1$

mother cell because of its division. When a cell divides, the contents (network species) are divided between the two cells based on binomial distribution, as discussed in the Method section.

Figure 3.7 shows the cell population size at different doses of I. The curve showing a constant mean population size represents the basal condition ($k_f I = 0$). The intermediate curve indicating a slow decline in the population size represents stimulation at $k_f I = 0.01$ s^{-1}. The curve indicating a sharp decline represents stimulation at $k_f I = 0.1$ s^{-1}.

3.2.5. Modeling Extracellular Environment. Figure 3.8 demonstrates cell population response in Model 1 under a temporally-evolving extracellular environment. To model the cell environment, an additional rule is included (Type 3 input) in the master program of Model I. This rule introduces a periodic pulse in I, as shown in Figure 3.8 A.

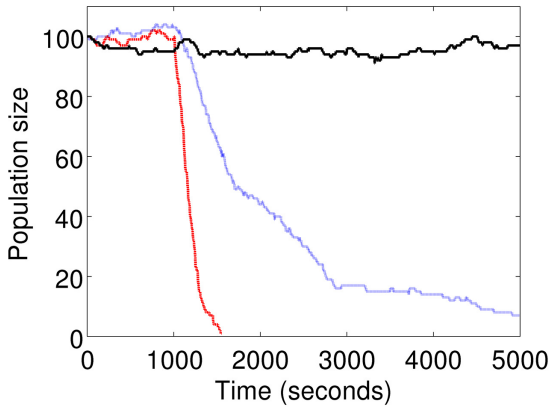


Figure 3.7. Cell population dynamics at different doses of stimulation, indicated by black ($k_f I = 0$ s^{-1}), blue ($k_f I = 0.01$ s^{-1}), and red ($k_f I = 0.1$ s^{-1}), respectively

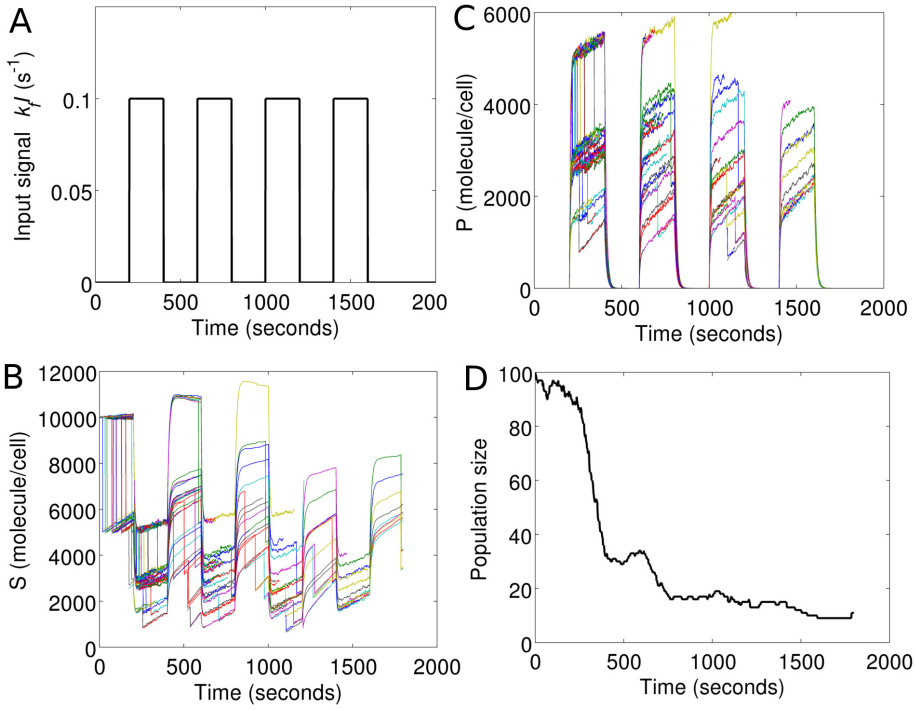


Figure 3.8. Modeling cell environment. (A) The cell environment with pulse variation in extracellular I.(B) Intracellular S of individual cells. (C) Intracellular P of individual cells. (D) Time evolution of the cell population size

At the end of each Δt, the master program sends a message to each cell object. The message contains a new value of I for the global environment, which is shared by all cells. Upon receiving the message, the new value of I replaces the old one, in each cell object. The cell objects then propagate the reactions (Gillespie algorithm) based on this new value of I. Figure 3.8 B and C show the intracellular S and P in each cell in the changing environment. Figure 3.8 D shows corresponding changes in the cell population size as a function of time.

3.2.6. Cellular Communication. Model II is inspired by multicellular systems where cell-to-cell communications determine the collective behavior of cells [138]. In many physiologically-relevant systems, cells respond in an interdependent manner. Such dependencies may arise from the direct or indirect communications among the cells. Cells may directly communicate via physical interactions [139]. On the other hand, indirect communications may arise when cells modify a shared growth environment. Cells can release proteins or metabolites into the environment, which may affect other cells. An example of such indirect communication is called quorum sensing [140].

Recent studies have identified a family of quorum-sensing peptides, called Extracellular Death Factors (EDFs), that induces programmed cell death in *E.coli* [141] and several other bacteria [142]. Reportedly, secretion of these peptides by bacteria may elicit collective cell death in a population. In Model II, a similar scenario is considered where cells can produce and secrete I in proportion to their intracellular P. This is implemented by simply adding a rule (a Type 3 input) in the master program of Model I. In each $\Delta t = 1$ second, the master program evaluates the intracellular P in each cell object and the secretion of I by the cell object. The global environment is changed based on the contributions from all cells in the population. The change in the environment is implemented by sending messages, as explained in the previous section.

Figure 3.9 shows the population response in Model II. In the top panels (Figure 3.9 A-D), an initial population of 10 cells are launched at time zero, and stimulated at $t = 500$ seconds with a small dose of the extracellular signal, $k_f I = 10^{-4}$ s^{-1}. The population

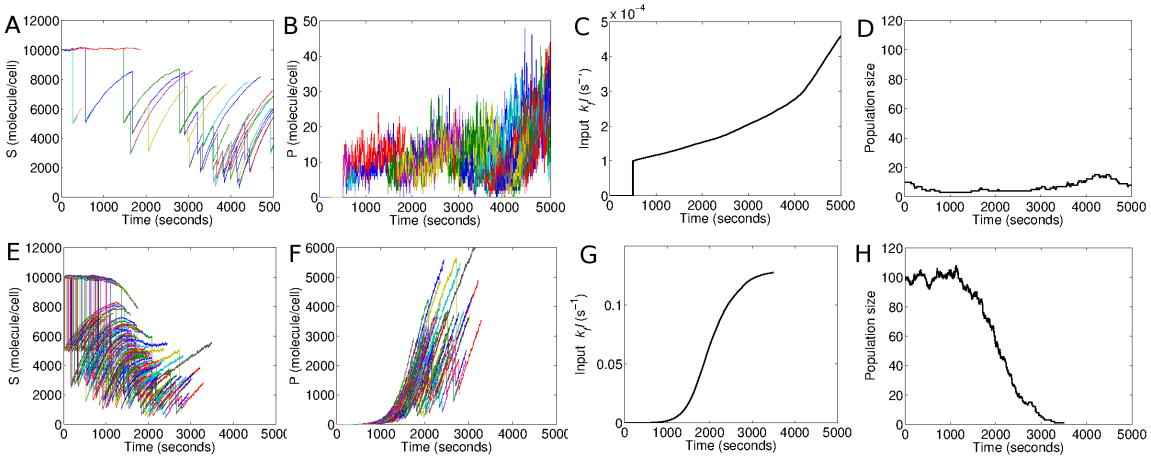


Figure 3.9. Population response under cellular communications in Model II. In the top panels (A-D), an initial population of 10 cells was subject to $k_f I = 10^{-4}$ s^{-1} at $t = 500$ seconds. In the bottom panels (E-F), an initial population of 100 cells is subject to $k_f I = 10^{-4}$ s^{-1} at $t = 500$ seconds. (A), (E) Intracellular S in individual cells; (B), (F) Intracellular P in individual cells; (C),(G) temporal evolution of extracellular I in the environment; (D),(H) Cell population size as function of time

remains non-responsive to this signal. The average population size is not affected, as seen in Figure 3.9 D. In the bottom panels (Figure 3.9 E-H), an initial population of 100 cells are launched and subjected to the same level of stimulation, $k_f I = 10^{-4}$ s^{-1}, at $t = 500$ seconds. Except for the population size, all other conditions are identical to those in the top panels. Nevertheless, contrary to the smaller population, this larger population displays a dramatic response (Figure 3.9 E-F). The quorum-sensing positive feedback is triggered because of the collective contribution of the population. This leads to a rapid buildup of intracellular P (Figure 3.9 F) and extracellular I (Figure 3.9 G). As a result, the population shows a rapid post-stimulation decay (Figure 3.9 H).

It should be noted that the selection of the population sizes in this example (100 or 10 cells) is completely arbitrary. In reality, millions of cells may participate in such a process. Nevertheless, the same qualitative differences could be reproduced at different

relative scales of the model. An increase in the model scale (population size) would simply require an adjustment in the parameter k_e (Table 3.2), which denotes the rate of secretion of I by each cell per molecule of its intracellular P.

Previous examples are involved with two mock models with very simple sets of intracellular reactions. Here, a full-scale model of bacterial quorum sensing is considered with a more complex intracellular biochemical network. The model is based on an earlier model by Boada et al. [87]. The intracellular biochemical network reactions associated of this model are illustrated in Figure 3.10. The expansion of this reaction network system into a population model is also illustrated in Figure 3.10. In the master-slave scheme of this framework (Figure 3.10), the biochemical reactions in each cell is independently simulated using the Gillespie algorithm for a short interval Δt. Each cell object simulates Gillespie algorithm based on the reactions shown in Figure 3.10. The updated intracellular species concentrations from each cell object (slave process) are sent back to the master process. The master process then updates the common (well-mixed) extracellular environment based on the rules specified as an input to the master process (Figure 3.10). Here the master takes two types of inputs. Type 1 input describes the distribution of parameters to incorporate cellular heterogeneity and Type 2 input is the relationship between cell and environment defined as the net change between external autoinducer concentration (AHL_{ext}) and internal autoinducer concentration (AHL) over each time interval Δt. The parameters values associated with this model are taken from [87].

In the model, cell division is simulated by splitting the molecular concentration of the mother cell based on binomial distribution into two daughter cells. The cell population size is kept constant by killing (removing) a randomly selected cell (parallel thread) every time a cell division occurs. Figure 3.11 shows comparison of the computational performance between the parallel model and the sequential model, carried out on dedicated 50 cores of a high performance computing cluster. Using the two models, different cell population sizes were simulated for 100 minutes and corresponding simulation completion times (clock

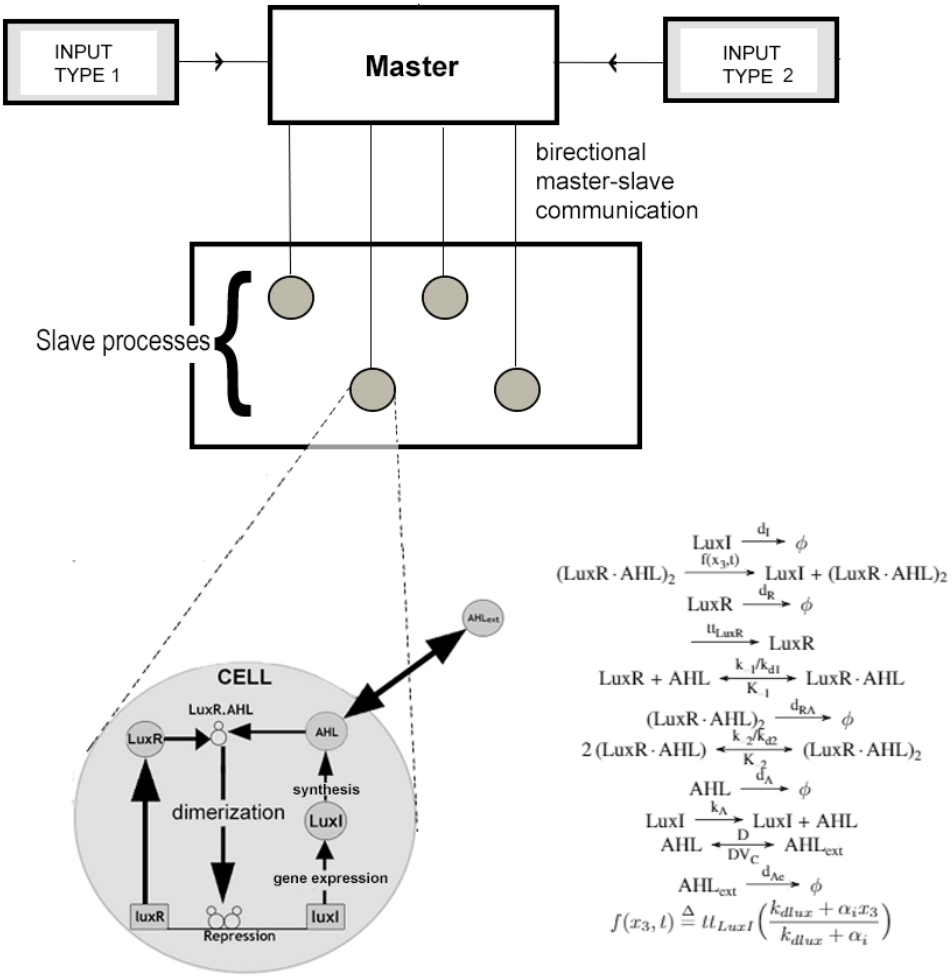


Figure 3.10. Schematic diagram of the multicellular quorum sensing model. The master process expands the single-cell biochemical network (set of reactions shown) into a multicellular model. Type 1 input defines the distributions of the model/network parameters values to incorporate cellular heterogeneity. Type 2 input defines relationship between a cell and the extracellular environment defined as the net change between external autoinducer concentration (AHL_{ext}) and internal autoinducer concentration (AHL) over each time interval Δt

time) were recorded for each model. The result indicates the parallel algorithm is at least two orders of magnitude faster than the corresponding sequential algorithm for a population of 25 cells. Moreover, this gap between the parallel and the sequential algorithms widens with the increase in population size. Note that existing literature on quorum sensing [86, 89] shows modeling of up to 240 cells using sequential approaches. Figure 3.11 shows that

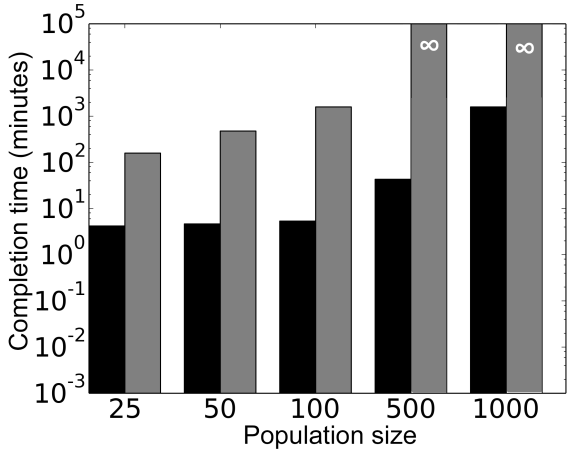


Figure 3.11. Semi-log plot for computational performance of the parallel quorum sensing model compared against a corresponding accurate (sequential) model. The black and grey bars represent the parallel and the sequential model respectively. The infinite signs represent the cases where simulation of the sequential model remained unfinished after 168 hours of simulation

simulation of 500 and 1000 cells using the sequential approach remain unfinished after 168 hours and are denoted by ∞. On the contrary, the proposed parallel framework exhibits scalability by simulating up to 1000 cells in less than 18 hours. A detailed description of the QS model in the parallel framework and algorithms are discussed in next Section.

3.3. DISCUSSION

The goal of this work is to introduce a simple method for developing mechanistic but computationally efficient cell population models. The question that is raised here was whether it is possible to systematically expand a biochemical network model into a cell population model. The commonly employed techniques to model multicellular systems include the continuum (equation-based) approach, cellular potts modeling (CPM) [26], and discrete agent-based modeling [143]. The equation-based approach and CPM are useful for modeling long-time behavior and evolution of cells. However, these methods have limited ability to capture the intracellular biochemistry in a population model.

Arguably, agent-based modeling (ABM) is the most versatile framework for mechanistic modeling of multicellular systems [144–147]. In an agent-based model, individual cell agents could be assigned with cellular attributes at various time and spatial scales. But mechanistic agent-based models could be computationally expensive and demand significant programming efforts.

Unlike the agent-based approach, cells are presented by stand-alone parallel simulation processes on a biochemical network model. It is shown that MPI/multiprocessing-based remote communications can be used to treat such parallel processes as software objects like an agent-based model. In an agent-based approach, parallelism could also address the scalability issues and speed up computation. However, such implementation could be model-specific and require expertise in parallel logic implementation. The aim of this work is to provide a modular framework that would require minimal programmatic efforts. A model is seperated into two different programs. It is shown that cellular-scale processes described in one program can be linked to the molecular-scale processes described in another program remotely to create a unified model. This allows a cellular and a biochemical network model to be defined separately and then combined to create a mechanistic population model.

This framework can be extended to make it compatible with other modeling languages and software platforms. It can be extended to create cell population models from the biochemical network models developed using other tools.

For demonstration purposes, the cellular biochemistry in this example model is defined by a network of four elementary reactions. However, the network could be made as detailed as intended. Many advanced tools can develop highly mechanistic biochemical network models. For example, the rule-based modeling tools [120, 121, 148] can model biochemical network systems where cellular protein molecules can be defined with their site-specific details, such as binding domains and phosphorylation motifs [123, 149, 150]. This framework can be integrated with such tools. This can be easily accomplished by

embedding necessary communication calls between a master program and the program running the simulation on a rule-based model. Such integration may provide the ability to study cellular evolution arising from the site-specific (point) mutations in the cellular protein domains and motifs.

This framework can only expand biochemical network models formulated in a stochastic approach, such as Gillespie method [38]. In a stochastic model, it is possible to introduce a run-time change in the model parameter values and simulation conditions. In a deterministic model's such changes would lead to discontinuities. Therefore, it is not extensible for the deterministic models.

An important limitation of this approach is that it must synchronize the parallel simulation processes (cell objects) at discrete intervals. The synchronization limits the computation speed by the slowest process in a population. In a large number of parallel processes implementing stochastic simulations, the cost of computation could be considerably different from the processes. This issue could be partially addressed by developing a dynamic load-balancing and resource allocation scheme.

A second limitation is an overhead associated with Master-slave communications. For a large number of cells in a model, such communication overhead could be significant. Overhead can be the limiting factor if the simulation of the biochemical network is not computation intensive. This issue could be partly addressed through serialization of the communication messages. In the current framework, the messages are passed as C data structures without serialization. The cost of overhead could also be addressed by developing a more distributed system having multiple servers and clients. Current framework is based on a centralized system, where communications occur between a single server and multiple clients.

This framework presents a general approach to connect the molecular scale to the cell population scale. It is noteworthy that cancer cells have similar multi scale features in low scale (production of molecules from genes), intermediate scale (interaction of gene

with its cellular environment) and high scale (collective behavior of cells) [127–130]. This framework can incorporate these hallmarks to study the nature and phenotype of cancer cells. Furthermore, it is intuited that this proposed framework can also be applied to any drug delivery system, where the concentration of optimum drug dosage can be quantitatively estimated [133–135, 151]. The different dosages (e.g. constant dose, periodic dose) of drug can be modeled as environmental variables as described in Figure 3.7 and Figure 3.8.

Finally, this approach could also find applications in modeling biological systems that are of scientific, pathological, and clinical interests. Examples include the evolutionary mechanisms of stem cells [152], clonal expansion of B and T lymphocytes [153, 154], autocrine, paracrine, and endocrine signaling [155, 156], and chemotactic cell migration [157].

3.4. REMARKS

In this work, a new framework is demonstrated for the systematic development of multiscale cell population models. The framework takes a biochemical network model as an input and expands it into a population model by linking intracellular network dynamics to cellular functions, fate decisions, and evolution. This capability is provided by a unique Master-slave scheme. This scheme also enables modeling cell-to-cell and cell-environment communications. The framework is further extensible to modeling multicellular systems evolving under both spatially- and temporally-resolved growth environments. In next Section, the framework's capability is demonstrated by simulating a full-scale model of bacterial quorum sensing, where the dynamics of a population of bacterial cells is dictated by the intercellular communications in a time-evolving growth environment.

4. A SCALABLE PARALLEL FRAMEWORK FOR MULTICELLULAR COMMUNICATION IN BACTERIAL QUORUM SENSING

Bacteria are single cell organism. But certain species of bacteria can communicate with each other via signaling molecules. This process is called Quorum Sensing (QS). Bacteria can release and sense these signaling molecules in the environment known as autoinducer. Based on the concentration of the signaling molecule in the environment, they can sense other bacteria around them and show coordinated behavior. Despite stochastic fluctuations, bacteria gradually achieve coordinated gene expression through QS, which in turn, help them better adapt to environmental adversities. This way bacteria cells behave like a multicellular organism. The mechanism in bacterial QS is simple and well studied. There are sufficient experimental results available on the mechanism of bacterial QS. So, understanding the fundamental property from the experimental outcome will give us a better understanding of complex biological processes. Existing sequential approaches for modeling information exchange via QS for large cell populations are time and computational resource intensive because the advancement in simulation time becomes significantly slower with the increase in molecular concentration.

In this Section, the bacterial quorum sensing model is incorporated into the ParCell framework described in Section 3 and presents a scalable parallel framework for modeling multicellular communication. Simulations show that the framework accurately models the molecular concentration dynamics of the QS system, yielding better speed-up and CPU utilization than the existing sequential model that uses the exact Gillespie algorithm. This framework can also accommodate evolving population due to cell birth, death and heterogeneity due to noise. Furthermore, the performance of the framework vis-á-vis the effects of its data sampling interval and Gillespie computation time are analyzed. Then, the model is simulated for a large cell population to observe the scalability of the framework. Note that existing literature on quorum sensing [86, 89] is capable of modeling up to

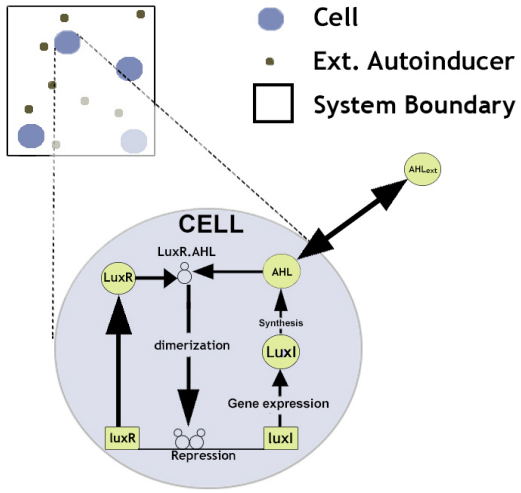


Figure 4.1. Population of cells and macro view of each cell with the LuxI/LuxR regulatory network

240 cells. But the proposed framework can model up to 2000 cells and beyond. So, the scalability of the proposed framework is validated by modeling population size up to 2000 bacterial cells.

This Section is organized as follows. Section 4.1 presents an overview of the overall QS system. Section 4.2 discusses the details of the Gillespie algorithm, sequential model and the parallel QS framework. Section 4.3 compares the experimental results. Finally, Section 4.4 closes the paper with concluding remarks.

4.1. SYSTEM OVERVIEW

Quorum Sensing (QS) system consists of a population of cells and their shared environment, defined as the concentration of autoinducers outside the cells (external autoinducer). Figure 4.1 shows a population of cells and LuxI/LuxR regulatory network within bacterium *Vibrio fischeri* [87].

In a QS system, bacteria communicate with each other through autoinducers (AHL), produced due to the synthesis of protein LuxI. These molecules are small in size and can diffuse freely through cell membrane into the environment and from the environment

back into cell. The diffusion process changes the concentration of environment, which in turn, affects the whole population. The overall system remains coupled through this diffusion process. Within a cell, the LuxR protein binds with AHL to form the monomer (LuxR.AHL). Then, through dimerization, this monomer forms $(\text{LuxR.AHL})_2$, repressing transcription of luxI gene. Diffusion rate of intra-cellular autoinducer AHL depends on the concentration of AHL itself, and extra-cellular autoinducer (AHL_{ext}). The chemical reactions and corresponding rate constants are taken from [87].

Below, a list of the chemical reactions is provided (in accordance with the reduced QS model discussed in [87]). The propensity of the LuxI expression reaction is shown in Equation 4.1 and the rate constant parameters for the reactions are listed in Table 4.1.

$$\text{LuxI} \xrightarrow{\text{d}_\text{I}} \phi$$

$$(\text{LuxR}\cdot\text{AHL})_2 \xrightarrow{\text{f(x}_3\text{,t)}} \text{LuxI} + (\text{LuxR}\cdot\text{AHL})_2$$

$$\text{LuxR} \xrightarrow{\text{d}_\text{R}} \phi$$

$$\xrightarrow{\text{tt}_{\text{LuxR}}} \text{LuxR}$$

$$\text{LuxR} + \text{AHL} \underset{\text{K}_{-1}}{\overset{\text{k}_{-1}/\text{k}_{\text{d1}}}{\longleftrightarrow}} \text{LuxR}\cdot\text{AHL}$$

$$(\text{LuxR}\cdot\text{AHL})_2 \xrightarrow{\text{d}_{\text{RA}}} \phi$$

$$2\,(\text{LuxR}\cdot\text{AHL}) \underset{\text{K}_{-2}}{\overset{\text{k}_{-2}/\text{k}_{\text{d2}}}{\longleftrightarrow}} (\text{LuxR}\cdot\text{AHL})_2$$

$$\text{AHL} \xrightarrow{\text{d}_\text{A}} \phi$$

$$\text{LuxI} \xrightarrow{\text{k}_\text{A}} \text{LuxI} + \text{AHL}$$

$$\text{AHL} \underset{\text{DV}_\text{C}}{\overset{\text{D}}{\longleftrightarrow}} \text{AHL}_{\text{ext}}$$

$$\text{AHL}_{\text{ext}} \xrightarrow{\text{d}_{\text{Ae}}} \phi$$

$$f(x_3, t) \triangleq tt_{LuxI}\left(\frac{k_{dlux} + \alpha_i x_3}{k_{dlux} + \alpha_i}\right) \tag{4.1}$$

Table 4.1. List of constant parameters [87] used in the parallel QS framework

Parameter	Description	Value
tt_{LuxR}	Protein expression rate: $LuxR$	76 $copies/min$
tt_{LuxI}	Protein expression rate: $LuxI$	219 $copies/min$
k_{-1}	Dissociation rate: $LuxR$ to AHL	10 min^{-1}
k_{-2}	Dissociation rate: $(LuxR.AHL)_2$	1 min^{-1}
α	Basal expression rate: $luxI$	0.01
k_A	Synthesis rate: AHL by $LuxI$	0.04 min^{-1}
D	Diffusion rate: AHL	2 min^{-1}
k_{d1}	Dissociation const.: $LuxR$ to AHL	100 $molecule$
k_{d2}	Diss. const.: $(LuxRAHL)$	20 $molecule$
k_{dlux}	Diss. const.: $(LuxRAHL)$ to lux	100 $molecule$
d_I	Degradation rate: $LuxI$	0.027 min^{-1}
d_R	Degradation rate: $LuxR$	0.156 min^{-1}
d_A	Degradation rate: internal AHL	0.057 min^{-1}
d_{A_e}	Degradation rate: external AHL	0.04 min^{-1}
d_{RA}	Degradation rate: $(LuxR.AHL)$	0.156 min^{-1}
d_{RA_2}	Degradation rate: $(LuxR.AHL)_2$	0.017 min^{-1}
V_{cell}	Initial cell volume	$1.1e^{-9}$ $liter$
V_{ext}	Extracellular volume	$1.1e^{-3}$ $liter$

4.2. SEQUENTIAL AND PARALLEL QS FRAMEWORKS

In this section the details of the sequential model for QS and the proposed parallel framework are discussed.

4.2.1. System Variables. Here, a system with χ reactions is considered. The reactions set is defined as $\gamma = \{\gamma_i \mid i \in \mathbb{N}, i \leq \chi\}$, set of reaction rate constants $K = \{k_i | i \in \mathbb{N}, i \leq \chi\}$, and the concentration of j^{th} reactant in reaction γ_i as ω_i^j. A molecular concentration matrix $M_{n\times m}$ is considered, where n and m are the number of cells and molecule species in the QS system, respectively. Thus, M_i denotes molecular concentration vector of the i^{th} cell. Thus, $M_{i,5}$ stores the AHL_{ext} concentration for i^{th} cell. Since AHL_{ext} is a global system variable, $M_{i,5}$ remains same for all cells.

4.2.2. Gillespie Algorithm for QS. The Gillespie Stochastic Simulation Algorithm (SSA) is used to model the signaling pathways as discussed in Section 3. Gillespie algorithm is used to simulate the reaction network of individual bacterial cell and their environment taking four parameter as inputs: the set of reaction rules (R_r), reaction rate constant (R_c), initial simulation time t and molecular concentration matrix M. Then the Gillespie algorithm probabilistically update the simulation time change the concentration of the selected reactions occuring at that time instance.

4.2.3. Sequential Model in QS. In the sequential QS model, each cell contains one copy of the regulatory network (as shown in Figure 4.1). All the reactions in the entire population, including diffusion reaction between cell and environment, is considered as a single global system as discussed in previous Section. The Gillespie algorithm is invoked in a loop, until the current time t exceeds the total simulation time T. Note that the sequential QS, when used in conjunction with the exact Gillespie algorithm, yields accurate result. Therefore, the sequential model is implemented as a benchmark of accuracy for the proposed parallel QS framework.

4.2.4. Parallel Framework. The parallel QS framework is implemented using the Multiprocessing library of Python [83]. Here, different aspect of the parallel framework is discussed.

4.2.4.1. Steps in parallel framework. As shown in Figure 4.2, the virtual master process takes 3 inputs: reaction rules (R_r), environment and constant (R_c). (1) The master spawns several parallel processes. Here, each cell is modeled as a memoryless process, termed *cell process*. In a large scale system several cell processes are assigned to a single core. (2) In each time intervals, master sends molecular concentration and environment information to each cell process and invokes Gillespie Algorithm. (3) Each cell process runs the Gillespie algorithm locally and (4) returns the updated molecular concentration to the

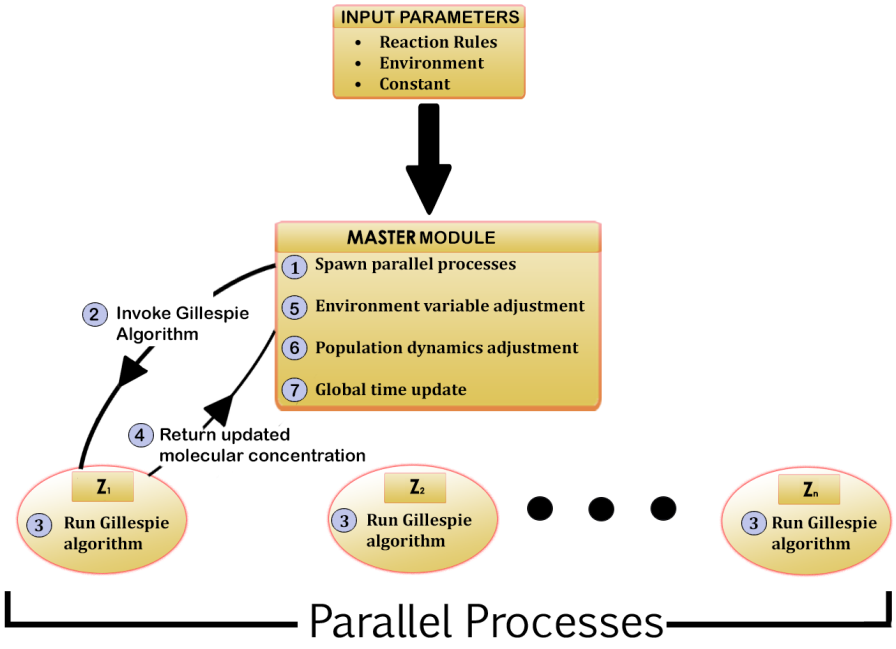


Figure 4.2. Overview of steps in the parallel QS framework

master. Following this, the master node (5) updates environment variable, (6) population dynamics and (7) global time, before returning to first step. This cycle continues until simulation duration is reached.

4.2.4.2. Master process. The master process reads and sends sampling interval Ψ, R_r, R_c and M_i to all the cell processes, at each time instance t. *Sampling interval (Ψ) is defined as the time interval between which the master process collects concentration data from all the cell processes.* After each Ψ interval, the master receives the updated M_i from each Z_i. It is noteworthy that in the sequential approach, a single reaction takes place at a time. Consequently, any change to AHL_{ext} is instantly reflected to $M_{i,5}$ (for all i). However, in the parallel approach, each cell process Z_i autonomously invokes Gillespie and updates $M_{i,5}$. Thus, the uniformity of AHL_{ext} in the system is ensured using Equation 4.2.

4.2.4.3. Cell processes. Each cell process Z_i invokes the Gillespie algorithm and returns updated M_i to the master. Two key aspects of the ParallelQS algorithm are discussed below.

```
1:  procedure ParallelQS()
2:      If ID = Master then
3:          t = 0
4:          ReadParameters(R_r, R_c, M, T, n, Ψ)
5:          While t ≤ T do
6:              While (not sentToAll()):
7:                  send(M_i, R_r, R_c, Ψ, AHL_ext(t))
8:              Endwhile
9:              While (not recvFromAll()) do
10:                 M_i, AHL^i_ext(t + 1) = recv()
11:             Endwhile
12:             J = {AHL^i_ext(t + 1)|i ≤ n}
13:             ADJUSTEXT(AHL_ext(t), J)
14:             t = t + Ψ
15:             Endwhile
16:     else
17:         (Cell process)
18:         M_i, R_r, R_c, Ψ, AHL_ext(t) = recv()
19:         gt = 0
20:         While gt ≤ Ψ do
21:             gt, M_i = GILLESPIE(R_r, R_c, M_i, gt)
22:         Endwhile
23:         send(M_i)
24:     Endif
25: end procedure
```

Time synchronization:. In each time instance, the master must wait till it has heard from all cell processes (Z_is). Following this, it increments the overall system time t by sampling interval Ψ.

Time increment:. Gillespie calculates the increment in simulation time t as $ln(1/r_1)/A$, where r_1 is random number between 0 and 1. Thus, there exists an inverse relationship between time variable t and overall propensity A (i.e., $t \propto \frac{1}{A}$). Note that the sequential model calculates A (Equations 3.1 and 3.2) based on all reactions in the system, whereas each Z_i in the parallel framework calculates A_i only based on the reactions specific to itself, implying that $A_i << A$. Hence, the time increment for the parallel framework is expected to be greater than that of the sequential approach.

Adjustment of global AHL_{ext} concentration:. The assumption here is that AHL_{ext} is homogeneously distributed within the system boundary, making it a global system variable. A cell is capable of interacting with any AHL_{ext} molecule. Master estimates the adjusted system AHL concentration at time $t + 1$, $AHL_{ext}(t + 1)$, by incrementing $AHL_{ext}(t)$ by the net difference of $AHL_{ext}(t)$ concentration from local cellular AHL concentrations, over all cells, using the equation below:

$$AHL_{ext}(t+1) = AHL_{ext}(t) + \sum_{j=2}^{n+1} (AHL_{ext}^{j}(t+1) - AHL_{ext}(t)) \tag{4.2}$$

Here $AHL_{ext}^{j}(t+1)$ is the local AHL concentration of the j^{th} cell process at time $t + 1$.

Adjustment of molecular concentration due to population dynamics. The proposed framework is capable of modeling dynamic population (i.e. cellular birth and death). Each cell Z_i undergoes division once simulation time t exceeds its division time β_i. For each Z_i, β_i is sampled from exponential distribution with $\mu = 45$ minutes. Each cell division causes the mother cell to split into two daughter cells, each containing half the molecular concentration of mother cell. Given initial volume of all cells V_{cell}, each Z_i has a *time-dependent volume*, given by –

$$V_i(t) = V_{cell} \times 2^{\frac{t}{\beta_i}} \tag{4.3}$$

Initial volume of a daughter cell is half the volume of its mother cell. The propensity of each reaction is also updated to account for time-dependent volume. In the QS system, cell density is kept constant by compensating each cell division by death of randomly picked cell (as discussed in [86]).

4.2.5. QS Model in ParCell. ParCell is a general software for modeling cell population from a single cell biochemical network. It is based on the framework discussed in Section 3. The software can seperately considers and synchronizes the environmental

species. So, this is a powerful tool to study the multicellular communication. The QS model is also implemented into the software. The implementation of QS model in ParCell is shown in Appendix B.

4.3. RESULTS

Here, 6 molecule species are considered in the reduced QS system – **(A)** LuxI, **(B)** LuxR, **(C)** $(LuxR.AHL)_2$, **(D)** AHL **(E)** LuxR.AHL and **(F)** AHL_{ext}. Parallel framework is implemented using Python 2.7 and Python Multiprocessing library [83]. Scalability experiments are performed on 50 cores of Forge high performance cluster; for other simulation experiments, Ubuntu system is used with Linux system with 8 CPUs of 1.6 GHz each.

4.3.1. Accuracy. First, how closely the dynamics of molecular concentration (matrix M) generated by parallel QS framework aligns with that of the sequential model is analyzed. On both *sequential and parallel approaches*, 50 cells are simulated for 200 minutes and sampling interval $\psi = 0.05$.

4.3.1.1. Similarity of molecular concentration. Figure 4.3 shows that the average concentration dynamics of molecules (A) - (F) (over 10 trials) are nearly identical for sequential and parallel approaches.

4.3.1.2. Accuracy at large scale. In Figure 4.4, the above model is validated against a corresponding accurate (sequential) model for a large population. In the sequential model, the intracellular reactions, cell death, and divisions associated with all cells in the population are grouped into one single list. The reactions from this list are sampled and executed following the Gillespie algorithm. As seen in the figure, despite the independent simulation of cells for discrete interval Δt, the parallel model is in perfect agreement with the sequential model. The yellow region is the standard deviation, representing cell to cell variability arising from distribution of rate constant parameters. The change of molecular concentration under cellular birth and death are shown in Appendix C.

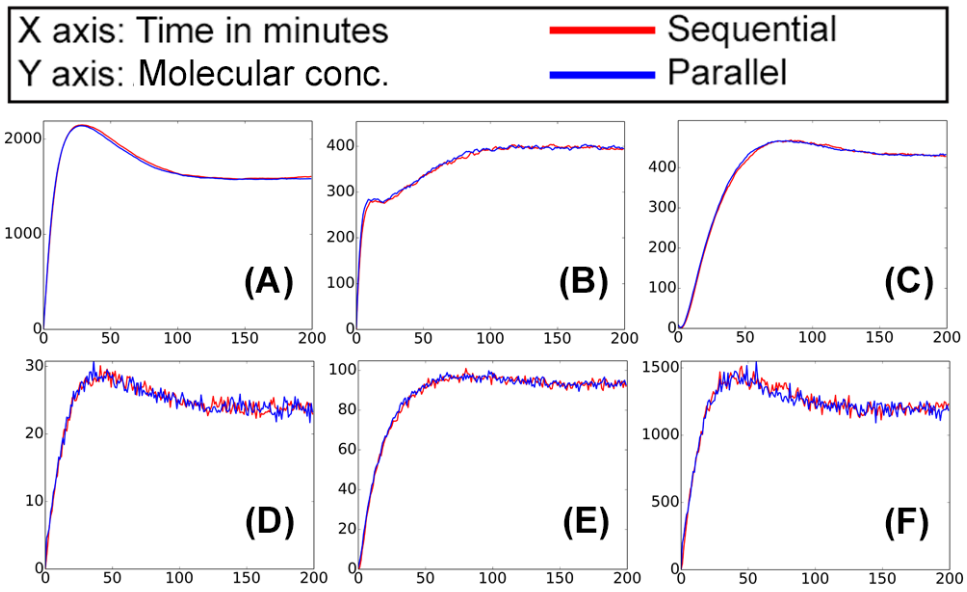


Figure 4.3. Comparison of average molecule concentration dynamics of 6 molecular species (A) LuxI, (B) LuxR, (C) $(LuxR.AHL)_2$, (D) AHL (E) (LuxR.AHL) and (F) (AHL_{ext}) for population of 50 cells

4.3.1.3. Effect of sampling interval. The choice of Ψ in the parallel framework affects the number of the data points master receives from each $Z[i]$. If Ψ is high, the framework exhibit inaccuracy in concentration dynamics due to lack of sufficient data. Here *Mean Absolute Error (MAE)* is considered as the metric of accuracy i.e. higher the MAE between two sets of plot points, greater is the dissimilarity. Since the sequential model is benchmark, the deviation of AHL_{ext} of the parallel framework (ϕ_p) from the sequential model (ϕ_S) on d data points is calculated as $MAE(\phi_s, \phi_p) = \frac{1}{d} \times \sum_d |M^s_{i,5}(d) - M^p_{i,5}(d)|$.

Simulation on 50 cells for duration of 200 minutes shows that the MAE of AHL_{ext} plots between the two approaches increases with the increase in Ψ (Figure 4.5). It is observed that a higher Ψ, though enhances running time (RT) by minimizing communication overhead, degrades accuracy.

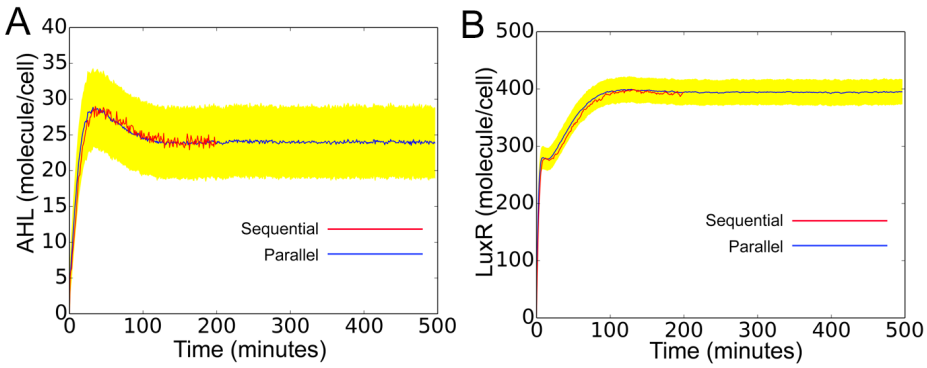


Figure 4.4. Prediction accuracy of the framework-created quorum sensing model is validated against an equivalent sequential model. Temporal evolution of (A) *AHL* and (B) LuxR. The blue and red curve represent the parallel model and the sequential model, respectively. The yellow region is the standard deviation, representing cell to cell variability arising from distribution of parameters. The temporal profiles of these concentrations are similar to those shown in [87]. Note that the sequential model was run only up to 200 seconds due to the prohibitively slow computation

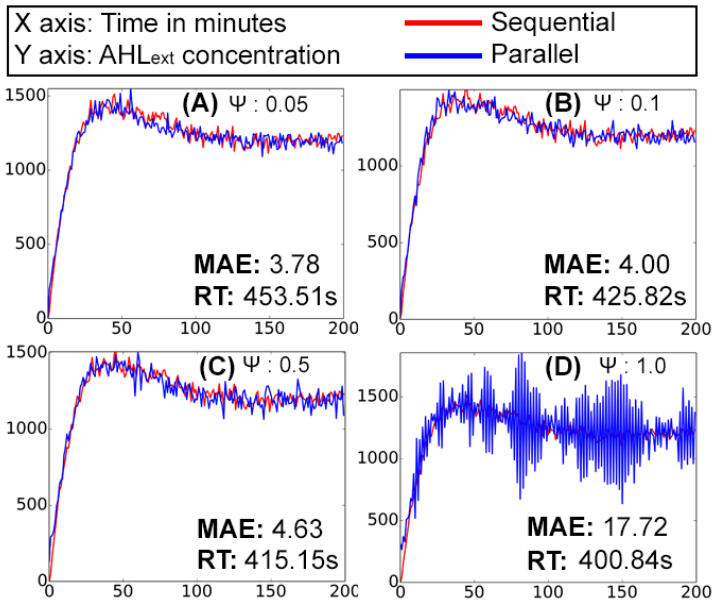


Figure 4.5. Increase in sampling interval (Ψ) increases Mean Absolute Error (MAE), but decreases running time (RT), between sequential and parallel QS framework

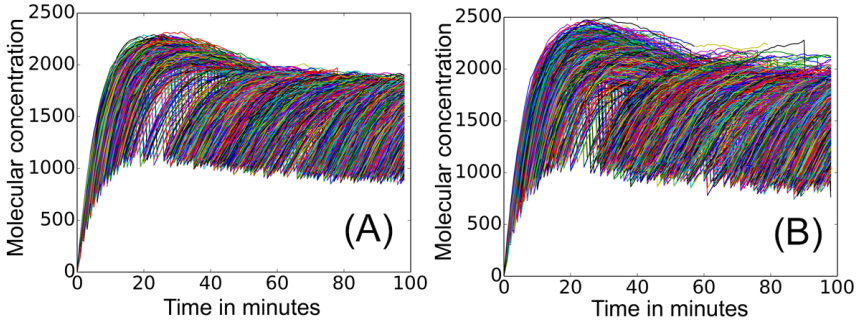


Figure 4.6. Population dynamics of LuxI concentration during cellular birth and death under conditions of (A) no noise and (B) noise with standard deviation 0.005

4.3.2. Population Dynamics. Cellular birth and death is modeled in QS for a population size 1000 cells and duration 100 minutes. In Figure 4.6 A, a single color represents the LuxI concentration of a bacterial cell over time, while the discontinuity and drop in the curves show cell death and division, respectively. Population dynamics of AHL and $LuxR$ molecules are shown in Appendix B.

The proposed parallel framework is capable of modeling noise arising from stochastic fluctuation in gene expression that can cause phenotypic variability in isogenic population [158]. In Figure 4.6 B, Cellular birth and death are shown under condition of noise generated by sampling constant parameters R_C from Gaussian distribution with standard deviation 0.005. It is noteworthy that molecular concentration in Figure 4.6 B, by virtue of the noise, exhibits greater phenotypic variability than the one in Figure 4.6 A.

4.3.3. Noise Analysis. Noise, arising from stochastic fluctuation in gene expression, can cause phenotypic variability in isogenic population [158]. Specifically, the heterogeneity induced by the noise is of particular interest in the fields of biotechnology and bio-medicine. Noise is incorporated in in the framework by sampling constant parameters R_C from Gaussian distribution with standard deviations 0, 0.025, 0.050, 0.075 and 0.1. Noise analysis is significant as it helps analyze the mean protein expression level. In Figure

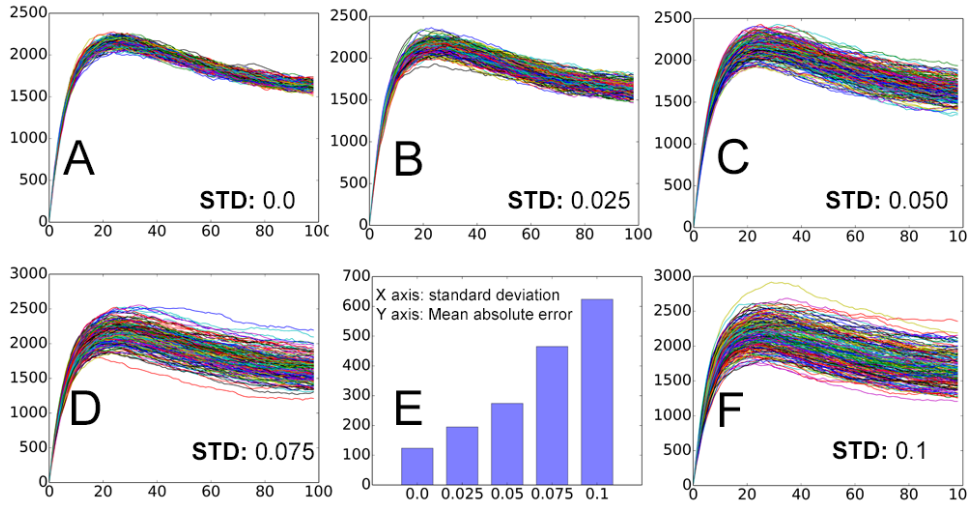


Figure 4.7. Noise analysis for different standard deviations (A) 0, (B) 0.025, (C) 0.05, (D) 0.075 and (F) 0.1. **X axis**: time in minutes; **y-axis**: molecular concentration. Figure (E) shows the phenotypic variability (mean absolute error) for varying deviation in constant parameters

4.7, the deviation from mean protein expression is shown for varying degrees of noise for 200 nodes, $\psi = 0.005$ and duration 100 minutes. Figure 4.7 E shows the mean deviation error for Figure 4.7 A-D and F.

4.3.4. Processor Utilization. The CPU utilization of the sequential and parallel frameworks is compared on a population size of 100 cells and duration of 60 minutes on an 8-CPU machine.

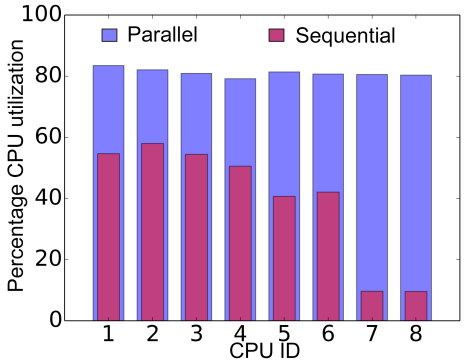


Figure 4.8. Comparison of CPU utilization for the sequential and parallel QS frameworks

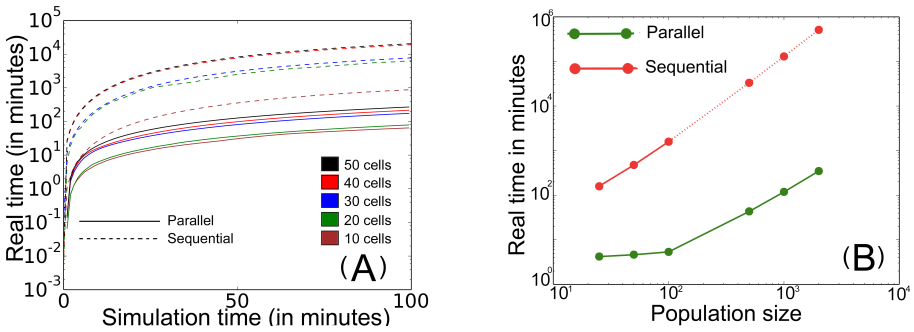


Figure 4.9. The speed up and scalability analysis of parallel QS framework. (A) Speed-up: Semi-log plot for speed-up for population sizes varying between 10 and 50. Parallel QS framework exhibits little growth in running time with increase in simulation time (B) Scalability: Log-log plot for scalability analysis for population sizes 25, 50, 100, 500, 1000, 2000

The psutil python library [159] is used to record the instantaneous CPU utilization for both frameworks. Figure 4.8 shows that the parallel framework exhibits a more uniform CPU utilization than the sequential approach.

4.3.5. Speed Up. Let the execution (or *wall clock*) time of sequential and parallel QS algorithms be ρ_s and ρ_p respectively, and the simulation time for both algorithms be T. The speedup is defined as $S_p = \frac{\rho_s}{\rho_p} = \frac{\rho_s}{T} \times \frac{T}{\rho_p} = \frac{\rho_s}{T} / \frac{\rho_p}{T}$. For sequential and parallel algorithms, a population of 10 to 50 cells are simulated for $T = 200$ minutes each. Given that T is same for both algorithms, their real (or execution) time is compared to analyze the speedup rendered by parallel QS framework. Semi-log plots are generated for execution time for population sizes 10, 20, 30, 40 and 50. Figure 4.9 A shows that the framework incurs extremely little increase in real time with growth in cell population size as compared to the sequential approach.

4.3.6. Scalability. Finally, the sequential and parallel approaches are compared on the basis of execution time for population sizes 25, 50, 100, 500, 1000 and 2000 cells. Figure 4.9 B is a log-log plot showing that the parallel framework incurs significantly lower

execution time. It is noteworthy that the sequential approach does not scale beyond 100 cells, thus the expected real time values for sequential approach (shown in dotted red line) is obtained through extrapolation.

4.4. DISCUSSION AND REMARKS

This framework is applied to model quorum sensing in bacteria, where a homogeneous distribution of environment across all cells is considered and their interaction with the environment through diffusion. As part of future work, it shall be extended to accommodate the spatial position of cells and the environmental molecules as discussed in [160, 161] to incorporate the cellular heterogeneity. There would be a minor variation in the framework once the spatial model is integrated. At present, this framework constitutes a master module which controls the activities of the cells in the system. In the spatial model, the master node will re-determine the coordinates of the cells and environmental molecules after fixed intervals of time.

This framework can be extended to model other bacteria, yeast [162], fungus and higher organisms (eukaryotic cells) [163, 164]. For example, the different proteins copy number of yeast and Hela cells are listed in Kulak et al. [165]. These modelings will follow a similar approach as quorum sensing, where bacteria synchronize themselves based on feedback response from their environment.

In this Section, a scalable parallel framework is presented for QS in bacteria. The framework is based on the general framework described in the previous Section. Simulation for varying population size, sampling interval and duration show that the framework models concentration dynamics almost as accurately as its sequential counterpart while showing significant improvement in speed-up and CPU utilization. Degradation in accuracy and improvement in running time are also studied of this model, with an increase in the sampling interval. This framework can incorporate cellular heterogeneity, phenotypic variability, and scalability by sampling the QS system parameters from Gaussian distribution and modeling

up to 2000 bacterial cells. Future works involve the extension of the framework to accommodate the spatial positioning of cells and environmental molecules to incorporate cellular heterogeneity. In the spatial model, the master process will recalculate the coordinates of the cells and environmental molecules after each sampling interval.

5. CONCLUDING REMARKS AND RECOMMENDATIONS

In this section, the concluding remarks and summary of the critical findings of this work alongside with recommendations for future action in modeling the dynamic behavior and multiscale nature of a biological system are presented.

5.1. CONCLUDING REMARKS

This work investigates the multiscale features of biological tissue. A multi-scale spatiotemporal model is proposed to study the penetration and dispersion of drug delivery nanoparticles. This scalable model can simulate tissue scale distribution of nanoparticles from particle-cell interaction and particle dynamics at the microscale. This model also considers nanoparticles as a specific size object and investigate the dynamics in tissue in the presence of cells, advection, and diffusion. Results suggest that the effect of nanoparticle size is less prominent in biological tissue due to the presence of particle-cell interaction at a fundamental scale. Then, a parallel framework is developed to study the dynamics of the cell population from a single cell biochemical reaction. This framework incorporates the multiscale features of intracellular dynamics to population dynamics. A Master-Slave communication process enables the framework to model different scale of space and time. The slave process simulated the molecular scale signaling pathways when the master process coordinated the tissue scale behavior. The framework's unique capability gives the ability to model cell to cell and cell to environment communications accurately and efficiently. Finally, the scalability, accuracy, efficiency, and performance of the framework are investigated based on an available model of quorum sensing. The result indicates that the framework can model multicellular communication for a large number of population of bacteria cells accurately and efficiently utilizing less computational resources compared to other approaches.

5.2. RECOMMENDATION FOR FUTURE WORK

At present, the multiscale spatiotemporal model for the distribution of nanoparticles can only consider specific size. But different shape and availability of specific binding site can affect the tissue scale distribution. The extension of this framework considering the effect of particle shape and specific binding site is on the developing phase.

Another extension of the model is to include drug molecules conjugated with nanoparticles. Nanoparticles can carry drug molecules in a deeper tissue space and release the drug molecules based on release kinetics. This model will further investigate the dynamics of drug molecule alongside with nanoparticles.

The multiscale spatiotemporal model proposed here do not explicitly consider the effect of charge, van der Walls force and corona effect. The impact of these fundamental features can be explored explicitly by hypothesis at the fundamental level based on the available experimental results. A future extension of the framework will explicitly incorporate these and investigate their effect on tissue-scale dynamics.

The parallel framework proposed here assumed a well-mixed system for the environmental molecules. But in real biological tissue, the particles and molecules are heterogeneously distributed. Integration of the spatiotemporal model to the parallel framework can simulate the multicellular communication in a biological tissue based on a heterogeneous distribution of signaling molecules and drug delivery nanoparticles. At present, the Master process controls the dynamic behavior of the cell population. In the parallel spatial framework, the Master will re-determine the co-ordinate of environmental particles and cells at each time interval between Master-Slave communication process. Then the slave process can simulate the reaction network based on the available environmental species around each cell. This will represent a more realistic architecture of biological tissue. As a future work, these two models will be integrated.

Furthermore, the parallel spatial framework can be incorporated into a drug delivery system. The dynamics of the cell population can be evaluated based on a constant and periodic supply of drug molecules. Experimental results at the tissue scale can be observed at tissue scale from a microfluidic experiment of tumor tissue in a chip. These experimental result can give us more knowledge on the mechanistic aspect of a drug delivery system. As a future work, a parallel spatial framework will be developed for drug delivery system.

Biological entities are structured based and has multiple binding sites. So, the interaction between two molecules can produce a large number of combinatorial modifications and complexes. This feature is known as combinatorial complexity. It is computationally intensive to include such details in a Bottom-up model. But it is essential because cellular regulation mostly depends on this complexity. As a future work, the combinatorial complexity will be included in the parallel framework.

APPENDIX A.

ALGORITHM FOR PARALLEL FRAMEWORK AND EXAMPLE MODELS

A.1. ALGORITHM

A.1.1. Master Process. (a) Launch initial cell population (parallel slave processes) on the biochemical network model with a set of default model parameter values (specified in the biochemical network model).

(b) Send messages (instructions) to each slave processes (living cells). The messages should contain one or more of the following: cell-specific new parameter values, initial concentrations (protein copy numbers), and extracellular environment. These values will replace the default model parameters and concentrations of the initial population. Wait until the slave processes communicate. The slave processes will communicate after all of them have propagated simulation (Gillespie algorithm) for a specific period.

(c) Read messages from the slave processes (cells). The messages contain the current simulation data (state variables, such as species concentrations).

(d) For each cell i, check and execute death by terminating the slave process.

(e) For each cell i, check and execute division by launching a new parallel slave process (simulation) to create a new cell.

(f) Evaluate the new rate constant for cell death.

(g) Evaluate the change in extracellular signal.

(h) Go back to step (b) and iterate until the end of the simulation time.

A.1.2. Slave Process i. (a) Receive and read the message sent by the master process. If applicable, replace the old parameters and state variables related to the cell attributes, species concentrations, and extracellular environment. Set a variable for time t to represents the current simulation time.

(b) Calculate intracellular reaction rates and sum of the reactions.

(c) Draw a uniform random number between 0 and 1. Select a reaction based on Gillespie Algorithm, and update the network.

(d) Advance time based on Gillespie Algorithm by a second uniform random number.

(e) Go back to step (b) and iterate until the time interval between Master Slave communication.

(f) Send a message to the master process. The message should contain information regarding the current state of the cell object (species, concentrations, extracellular environment, and other state variables).

A.2. DNA TRANSCRIPTION MODEL IN PARCELL

A.2.1. Model Representation in ParCell. Below is a representation of a DNA transcription model into ParCell model generator. # is used for commenting.

```
begin model
# Parameter block
begin parameter
b     20                # Burst factor
k1    0.01              # Transcription of DNA into RNA
k2    0.00577           # Degradation of mRNA
k3    k2*b              # Translation of RNA into protein P
k4    0.0001925         # production of mRNA
end parameter

# Species block
begin species
DNA
mRNA
P
end species

# Initial condition block
```

```
begin initial condition
DNA  1
end initial condition

# Reaction rules block
begin reaction rules
# 1. Transcription of DNA into mRNA
DNA -> DNA + mRNA           $ k1       $$ (constant)
# 2. Degradation of mRNA
mRNA ->                     $ k2       $$ (constant)
# 3. Translation mRNA into protein P
mRNA -> mRNA + P            $ k3       $$ (constant)
# 4. Degradation of protein P
P ->                        $ k4       $$ (constant)
end reaction rules
end model

simulate(N = 20, T = 3000, cores = 4, grain = 1, grainWrite = 1)
```

A.2.2. Protein Dynamics. The molecular dynamics for the above model are evaluated for 20 cells and 3000 seconds. Molecular dynamics and average molecular dynamics are shown in Figure A.1 and A.2.

A.3. MICHAELIS MENTEN MODEL IN PARCELL

A.3.1. Model Representation in ParCell. Below is a representation of a Michaelis menten model into ParCell model generator.

```
begin model
```

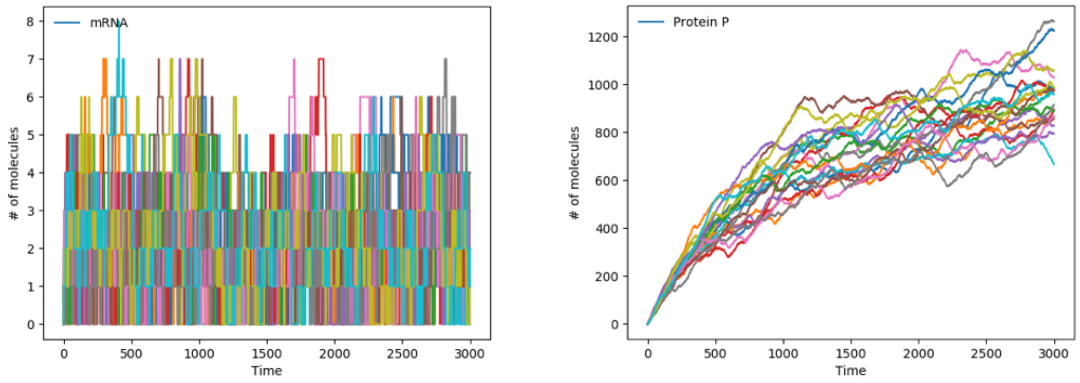


Figure A.1. Molecular dynamics of mRNA and protein P for 20 cells and 3000 seconds

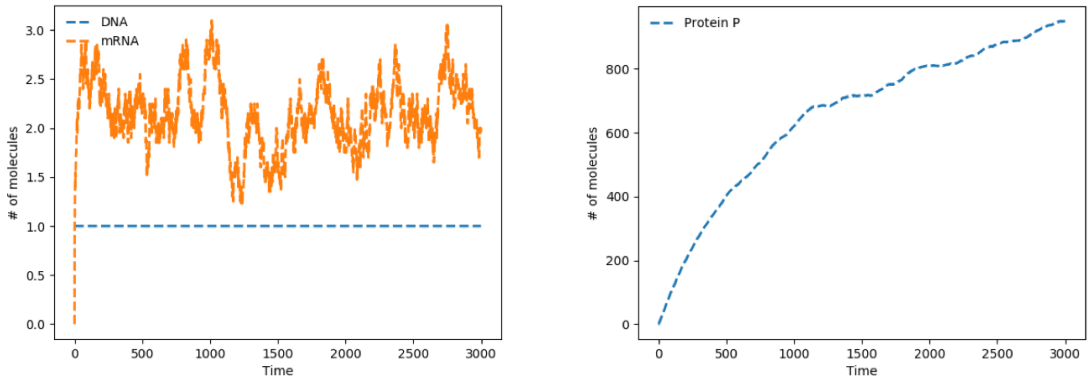


Figure A.2. Average molecular dynamics of DNA, mRNA and protein P for 20 cells and 3000 seconds

```
# Parameter block
begin parameter
kf      1e-5
kr      0.1
kp      0.05
end parameter

# Species block
begin species
E
S
ES
P
end species

# Initial condition block
begin initial condition
E     5e3
S     1e4
end initial condition

# Reaction rules block
begin reaction rules
E + S -> ES        $ kf          $$ (constant)

ES -> E + S        $ kr          $$ (constant)
```

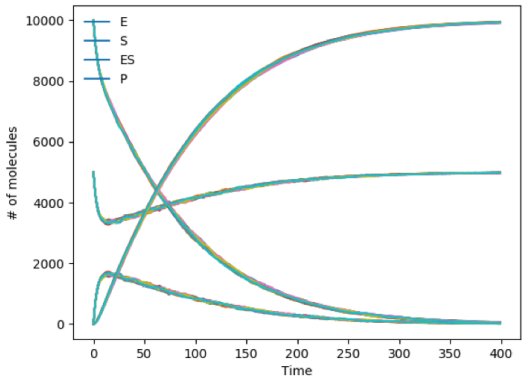


Figure A.3. Molecular dynamics of Michaelis Menten kinetics for 20 cells and 400 seconds

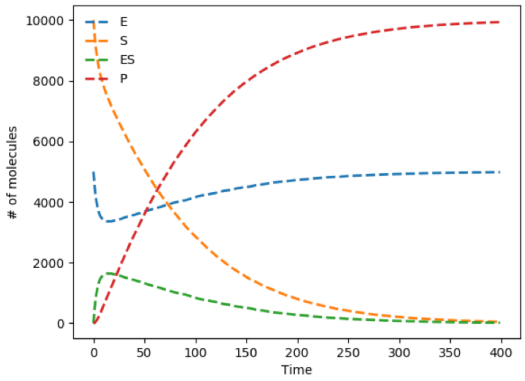


Figure A.4. Average molecular dynamics of Michaelis Menten kinetics for 20 cells and 400 seconds

```
ES -> P + E          $ kp              $$ (constant)

end reaction rules

end model


simulate(N = 20, T = 400)
```

A.3.2. Protein dynamics. The molecular dynamics for the above model are evaluated for 20 cells and 400 seconds. Molecular dynamics and average molecular dynamics are shown in Figure A.3 and A.4.

APPENDIX B.

APPLICATION OF QS MODEL IN PARCELL

B.1. QS MODEL IN PARCELL

B.1.1. Model Representation in ParCell. Below is a representation of a QS model into ParCell model generator.

```
begin model
# Parameter block
begin parameter
dI          0.027       # Degradation of LuxI
f(x3,t)     218.98      # Protein expression of LuxI
dR          0.156       # Degradation rate of LuxR
ttLuxR      76.12       # Protein expression rate of LuxR
kf1         0.1         # Association constant of LuxR and AHL(IC)
kr1         10          # Unbinding rate of of LuxR.AHL(IC)
dc          0.017       # Degradation rate of (LuxR.AHL(IC))2
kf2         0.05        # Dimerization rate of LuxR.AHL(IC)
kr2         1           # Dissociation rate of dimer (LuxR.AHL(IC))2
dA          0.057       # Degradation rate of internal AHL(IC)
kA          0.04        # Synthesis rate of AHL(IC) by LuxI
D           2           # Diffusion rate of internal AHL(IC)
DVC         2.2e-6      # Diffusion rate of external AHL(EC)
dAe         0.04        # Degradation rate of external AHL(EC)
end parameter
# Species block
begin species
LuxI
LuxR
(LuxR.AHL(IC))2
```

```
AHL(IC)
LuxR.AHL(IC)
AHL(EC)
end species
# Initial condition block
begin initial condition
(LuxR.AHL(IC))2  10
end initial condition
# Reaction rules block
begin reaction rules
#1. degradation of LuxI

LuxI  ->               $ dI          $$ (constant)

#2. Production of LuxI

      ->   LuxI        $ f(x3,t)     $$ (constant)     $$$ DRR1

#3. Degradation of LuxR

LuxR    ->             $ dR          $$ (constant)

#4. Production of LuxI

        ->  LuxR       $ ttLuxR      $$ (constant)

#5. Association of LuxR and AHL(IC)
```

```
LuxR + AHL(IC)   ->   LuxR.AHL(IC)       $ kf1     $$ (constant)

#6. Dissociation of LuxR and AHL(IC)

LuxR.AHL(IC)      ->   LuxR + AHL(IC)    $ kr1     $$  (constant)

#7. Degradation of (LuxR.AHL(IC))2

(LuxR.AHL(IC))2  ->                      $ dc      $$ (constant)

#8. Dimerization of LuxR.AHL(IC)

LuxR.AHL(IC) + LuxR.AHL(IC) -> (LuxR.AHL(IC))2  $ kf2  $$ (constant)

#9. Dissociation of (LuxR.AHL)2

(LuxR.AHL(IC))2 -> LuxR.AHL(IC) + LuxR.AHL(IC)  $ kr2  $$ (constant)

#10. Degradation of autoinducer

AHL(IC)      ->                          $ dA       $$ (constant)

#11. Synthesis of AHL(IC)

LuxI     ->     LuxI  +  AHL(IC)         $ kA       $$ (constant)
```

```
#12. Diffusion of AHL(IC) from cell to environment

AHL(IC)      ->     AHL(EC)              $ D        $$ (constant)

#13. Diffusion of AHL(EC) from environment to cell

AHL(EC)  ->    AHL(IC)                  $ DVC      $$ (constant)

#14. Degradation of AHL(EC)

AHL(EC)   ->                            $ dAe      $$ (constant)
end reaction rules
#Dependent reaction block
begin dependent reaction rules
[DRR1] *= ((100 + 0.01 * moleculeA[(LuxR.AHL(IC))2]) /
(100 + moleculeA[(LuxR.AHL(IC))2]))
end dependent reaction rules
#Environmental species block
begin environmental species
AHL(EC)
end environmental species
end model

simulate(N = 20, T = 200, cores = 8 meanDivisionTime = 45)
```

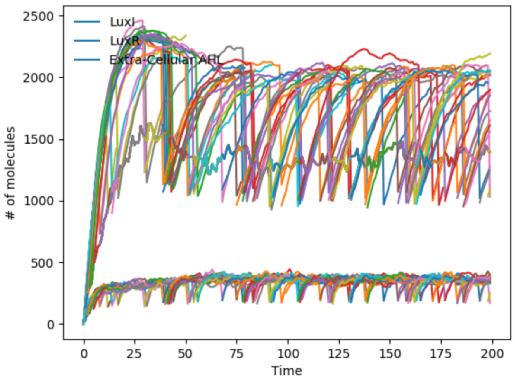


Figure B.1. Molecular dynamics of LuxI, LuxR and external AHL for 20 cells and 200 min

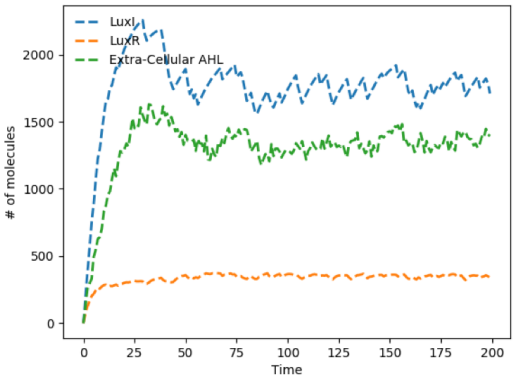
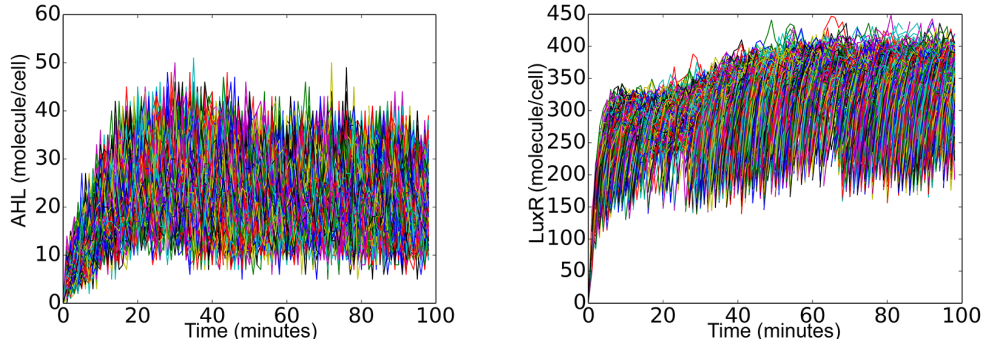


Figure B.2. Average molecular dynamics of LuxI, LuxR and external AHL for 20 cells and 200 min

The above model will run the simulation of 20 cells for 200 minute with 8 cores in machine and mean division time of 45 minute.

B.1.2. Protein Dynamics. Molecular dynamics and average molecular dynamics of LuxI, LuxR and external AHL are shown in Figure B.1 and B.2. Figure B.1 shows the trajectory of individual cells and Figure B.2 shows the average response of cell population.



(a) (b)

Figure B.3. Temporal evolution of molecular dynamics under cellular birth and death (a) AHL (b) LuxR

B.2. MOLECULAR CONCENTRATION UNDER CELLULAR BIRTH AND DEATH

Simulation is performed on a population of 1000 bacteria cells for duration of 100 minutes.

Figure B.3 shows the the evolution of concentration of AHL and LuxR molecules. The condition of constant cell density is followed for cellular birth and death as per [86].

APPENDIX C.

PUBLICATIONS

C.1. PUBLICATIONS

This dissertation consists of the following three publications, formatted in the style used by the Missouri University of Science and Technology:

Islam, Mohammad Aminul, Satyaki Roy, Sajal Das, and Dipak Barua. "Multicellular Models Bridging Intracellular Signaling and Gene Transcription to Population Dynamics." Processes 6, no. 11 (2018): 217.

Islam, Mohammad Aminul, Sutapa Barua, and Dipak Barua. "A multiscale modeling study of particle size effects on the tissue penetration efficacy of drug-delivery nanoparticles." BMC systems biology 11.1 (2017): 113.

Satyaki Roy, **Islam Mohammad Aminul**, Sajal Das, and Dipak Barua , " A Scalable Parallel Framework for Multicellular Communication in Bacterial Quorum Sensing " (Accepted in BICT 2019).

REFERENCES

[1] Zhilin Qu, Alan Garfinkel, James N Weiss, and Melissa Nivala. Multi-scale modeling in biology: how to bridge the gaps between scales? *Progress in biophysics and molecular biology*, 107(1):21–31, 2011.

[2] Martin Meier-Schellersheim, Iain DC Fraser, and Frederick Klauschen. Multiscale modeling for biologists. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(1):4–14, 2009.

[3] Giovanni Pezzulo and Michael Levin. Top-down models in biology: explanation and control of complex living systems above the molecular level. *Journal of the Royal Society Interface*, 13(124):20160555, 2016.

[4] Teresa Lehnert, Sandra Timme, Johannes Pollmächer, Kerstin Hünniger, Oliver Kurzai, and Marc Thilo Figge. Bottom-up modeling approach for the quantitative estimation of parameters in pathogen-host interactions. *Frontiers in microbiology*, 6:608, 2015.

[5] Armando Reyes-Palomares, Francisca Sánchez-Jiménez, and Miguel Ángel Medina. First steps in computational systems biology: a practical session in metabolic modeling and simulation. *Biochemistry and Molecular Biology Education*, 37(3):178–181, 2009.

[6] Andreas Schmid and Lars M Blank. Systems biology: Hypothesis-driven omics integration. *Nature chemical biology*, 6(7):485, 2010.

[7] Wendy A Comisar, Susan X Hsiong, Hyun-Joon Kong, David J Mooney, and Jennifer J Linderman. Multi-scale modeling to predict ligand presentation within rgd nanopatterned hydrogels. *Biomaterials*, 27(10):2322–2329, 2006.

[8] T Alarcon, HM Byrne, and PK Maini. Towards whole-organ modelling of tumour growth. *Progress in biophysics and molecular biology*, 85(2-3):451–472, 2004.

[9] Michael Breakspear and Cornelis J Stam. Dynamics of a neural system with a multiscale architecture. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):1051–1074, 2005.

[10] Yoonsuck Choe. The role of temporal parameters in a thalamocortical model of analogy. *IEEE Transactions on Neural Networks*, 15(5):1071–1082, 2004.

[11] James B Bassingthwaighte, Howard J Chizeck, Les E Atlas, and Hong Qian. Multiscale modeling of cardiac cellular energetics. *Annals of the New York Academy of Sciences*, 1047(1):395–426, 2005.

[12] Shayn M Peirce and Thomas C Skalak. Microvascular remodeling: a complex continuum spanning angiogenesis to arteriogenesis. *Microcirculation*, 10(1):99–111, 2003.

[13] Steven McDougall, John Dallon, Jonathan Sherratt, and Philip Maini. Fibroblast migration and collagen deposition during dermal wound healing: mathematical modelling and clinical implications. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 364(1843):1385–1405, 2006.

[14] Benjamin Donald Cumming, DLS McElwain, and Zee Upton. A mathematical model of wound healing and subsequent scarring. *Journal of The Royal Society Interface*, pages rsif–2008, 2009.

[15] Tao Sun, Salem Adra, Rod Smallwood, Mike Holcombe, and Sheila MacNeil. Exploring hypotheses of the actions of tgf-β1 in epidermal wound healing using a 3d computational multiscale model of the human epidermis. *PloS one*, 4(12):e8515, 2009.

[16] Salem Adra, Tao Sun, Sheila MacNeil, Mike Holcombe, and Rod Smallwood. Development of a three dimensional multiscale computational model of the human epidermis. *PloS one*, 5(1):e8511, 2010.

[17] Huiming Peng, Weiling Zhao, Hua Tan, Zhiwei Ji, Jingsong Li, King Li, and Xiaobo Zhou. Prediction of treatment efficacy for prostate cancer using a mathematical model. *Scientific reports*, 6:21599, 2016.

[18] Jin Chen, Jose A Gomez, Kai Höffner, Poonam Phalak, Paul I Barton, and Michael A Henson. Spatiotemporal modeling of microbial metabolism. *BMC systems biology*, 10(1):21, 2016.

[19] Zachary Fox, Gregor Neuert, and Brian Munsky. Finite state projection based bounds to compare chemical master equation models using single-cell data. *The Journal of chemical physics*, 145(7):074101, 2016.

[20] Zhiwei Ji, Jing Su, Chenglin Liu, Hongyan Wang, Deshuang Huang, and Xiaobo Zhou. Integrating genomics and proteomics data to predict drug effects using binary linear programming. *PloS one*, 9(7):e102798, 2014.

[21] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245, 2010.

[22] René Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, 1973.

[23] Le Zhang, Chaitanya A Athale, and Thomas S Deisboeck. Development of a three-dimensional multiscale agent-based tumor model: simulating gene-protein interaction profiles, cell phenotypes and multicellular patterns in brain cancer. *Journal of theoretical biology*, 244(1):96–107, 2007.

[24] Xiaoqiang Sun, Yunqing Kang, Jiguang Bao, Yuanyuan Zhang, Yunzhi Yang, and Xiaobo Zhou. Modeling vascularized bone regeneration within a porous biodegradable cap scaffold loaded with growth factors. *Biomaterials*, 34(21):4971–4981, 2013.

[25] Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–219, 2007.

[26] François Graner and James A Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, 69(13):2013, 1992.

[27] Joost B Beltman, Athanasius FM Marée, Jennifer N Lynch, Mark J Miller, and Rob J de Boer. Lymph node topology dictates t cell migration behavior. *Journal of Experimental Medicine*, 204(4):771–780, 2007.

[28] Stefania Bandini, Giancarlo Mauri, and Roberto Serra. Cellular automata: From a theoretical parallel computational model to its application to complex systems. *Parallel Computing*, 27(5):539–553, 2001.

[29] Mohammed AlQuraishi, Grigoriy Koytiger, Anne Jenney, Gavin MacBeath, and Peter K Sorger. A multiscale statistical mechanical framework integrates biophysical and genomic data to assemble cancer networks. *Nature genetics*, 46(12):1363, 2014.

[30] Adnan Morshed, Prashanta Dutta, and Robert H Dillon. Mathematical modeling and numerical simulation of the tgf-β/smad signaling pathway in tumor microenvironments. *Applied Numerical Mathematics*, 133:41–51, 2018.

[31] T Eisenhammer, A Hübler, N Packard, and JA Scott Kelso. Modeling experimental time series with ordinary differential equations. *Biological cybernetics*, 65(2):107–112, 1991.

[32] Dominik M Wittmann, Jan Krumsiek, Julio Saez-Rodriguez, Douglas A Lauffenburger, Steffen Klamt, and Fabian J Theis. Transforming boolean models to continuous models: methodology and application to t-cell receptor signaling. *BMC systems biology*, 3(1):98, 2009.

[33] Tao Peng, Huiming Peng, Dong Soon Choi, Jing Su, Chung-Che Jeff Chang, and Xiaobo Zhou. Modeling cell–cell interactions in regulating multiple myeloma initiating cell fate. *IEEE journal of biomedical and health informatics*, 18(2):484–491, 2014.

[34] Celine I Maeder, Mark A Hink, Ali Kinkhabwala, Reinhard Mayr, Philippe IH Bastiaens, and Michael Knop. Spatial regulation of fus3 map kinase activity through a reaction-diffusion mechanism in yeast pheromone signalling. *Nature cell biology*, 9(11):1319, 2007.

[35] Lakshmi Machineni, Ch Tejesh Reddy, Vandana Nandamuri, and Parag D Pawar. A 3d individual-based model to investigate the spatially heterogeneous response of bacterial biofilms to antimicrobial agents. *Mathematical Methods in the Applied Sciences*, 41(18):8571–8588, 2018.

[36] Shigeru Kondo and Takashi Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *science*, 329(5999):1616–1620, 2010.

[37] Yanhang Zhang, Victor H Barocas, Scott A Berceli, Colleen E Clancy, David M Eckmann, Marc Garbey, Ghassan S Kassab, Donna R Lochner, Andrew D McCulloch, Roger Tran-Son-Tay, et al. Multi-scale modeling of the cardiovascular system: disease development, progression, and clinical intervention. *Annals of biomedical engineering*, 44(9):2642–2660, 2016.

[38] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4): 403–434, 1976.

[39] H. Li, Y. Cao, L. R Petzold, and D. T Gillespie. Algorithms and software for stochastic simulation of biochemical reacting systems. *Biotechnology progress*, 24 (1):56–61, 2008.

[40] Nicolas Le Novere and Thomas Simon Shimizu. Stochsim: modelling of stochastic biomolecular processes. *Bioinformatics*, 17(6):575–576, 2001.

[41] Alex Sigal, Ron Milo, Ariel Cohen, Naama Geva-Zatorsky, Yael Klein, Yuvalal Liron, Nitzan Rosenfeld, Tamar Danon, Natalie Perzov, and Uri Alon. Variability and memory of protein levels in human cells. *Nature*, 444(7119):643, 2006.

[42] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural & Molecular Biology*, 9(9):646, 2002.

[43] Vahid Vahedi and Maheswaran Surendra. A monte carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges. *Computer Physics Communications*, 87(1-2):179–198, 1995.

[44] ES Oran, CK Oh, and BZ Cybyk. Direct simulation monte carlo: recent advances and applications. *Annual Review of Fluid Mechanics*, 30(1):403–441, 1998.

[45] Zachary Grant Mills, Wenbin Mao, and Alexander Alexeev. Mesoscale modeling: solving complex flows in biology and biotechnology. *Trends in biotechnology*, 31 (7):426–434, 2013.

[46] Di Su, Ronghui Ma, Maher Salloum, and Liang Zhu. Multi-scale study of nanoparticle transport and deposition in tissues during an injection process. *Medical & biological engineering & computing*, 48(9):853–863, 2010.

[47] Abhishek Majumdar, Stephen D Scott, Jitender S Deogun, and Steven Harris. Yeast pheromone pathway modeling using petri nets. *BMC bioinformatics*, 15(7):S13, 2014.

[48] Guy Karlebach and Ron Shamir. Minimally perturbing a gene regulatory network to avoid a disease phenotype: the glioma network as a test case. *BMC systems biology*, 4(1):15, 2010.

[49] Lingxi Li and Hiroki Yokota. Application of petri nets in bone remodeling. *Gene regulation and systems biology*, 3:GRSB–S2881, 2009.

[50] Simon Hardy and Pierre N Robillard. Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches. *Journal of bioinformatics and computational biology*, 2(04):619–637, 2004.

[51] Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[52] Maria I Davidich and Stefan Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, 3(2):e1672, 2008.

[53] Hila Amir-Kroll, Avital Sadot, Irun R Cohen, and David Harel. Gemcell: A generic platform for modeling multi-cellular biological systems. *Theoretical Computer Science*, 391(3):276–290, 2008.

[54] Jing Su, Le Zhang, Wen Zhang, Dong Song Choi, Jianguo Wen, Beini Jiang, Chung-Che Chang, and Xiaobo Zhou. Targeting the biophysical properties of the myeloma initiating cell niches: a pharmaceutical synergism analysis using multi-scale agent-based modeling. *PloS one*, 9(1):e85059, 2014.

[55] Ricardo Cortez. The method of regularized stokeslets. *SIAM Journal on Scientific Computing*, 23(4):1204–1225, 2001.

[56] Cliff Wong, Triantafyllos Stylianopoulos, Jian Cui, John Martin, Vikash P Chauhan, Wen Jiang, Zoran Popović, Rakesh K Jain, Moungi G Bawendi, and Dai Fukumura. Multistage nanoparticle delivery system for deep penetration into tumor tissue. *Proceedings of the National Academy of Sciences*, page 201018382, 2011.

[57] Li Tang, Nathan P Gabrielson, Fatih M Uckun, Timothy M Fan, and Jianjun Cheng. Size-dependent tumor penetration and in vivo efficacy of monodisperse drug–silica nanoconjugates. *Molecular pharmaceutics*, 10(3):883–892, 2013.

[58] Maria José Alonso. Nanomedicines for overcoming biological barriers. *Biomedicine & Pharmacotherapy*, 58(3):168–172, 2004.

[59] Rakesh K Jain and Triantafyllos Stylianopoulos. Delivering nanomedicine to solid tumors. *Nature reviews Clinical oncology*, 7(11):653, 2010.

[60] Elvin Blanco, Haifa Shen, and Mauro Ferrari. Principles of nanoparticle design for overcoming biological barriers to drug delivery. *Nature biotechnology*, 33(9):941, 2015.

[61] Sutapa Barua and Samir Mitragotri. Challenges associated with penetration of nanoparticles across cell and tissue barriers: a review of current status and future prospects. *Nano today*, 9(2):223–243, 2014.

[62] Marie Gaumet, Angelica Vargas, Robert Gurny, and Florence Delie. Nanoparticles for drug delivery: the need for precision in reporting particle size parameters. *European journal of pharmaceutics and biopharmaceutics*, 69(1):1–9, 2008.

[63] Martin Meier-Schellersheim, Xuehua Xu, Bastian Angermann, Eric J Kunkel, Tian Jin, and Ronald N Germain. Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method. *PLoS computational biology*, 2(7):e82, 2006.

[64] Martin Meier-Schellersheim and Gerhard Mack. Simmune, a tool for simulating and analyzing immune system behavior. *arXiv preprint cs/9903017*, 1999.

[65] Michael L Blinov, James R Faeder, Byron Goldstein, and William S Hlavacek. Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004.

[66] Larry Lok and Roger Brent. Automatic generation of cellular reaction networks with moleculizer 1.0. *Nature biotechnology*, 23(1):131, 2005.

[67] Pedro Mendes. Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Bioinformatics*, 9(5):563–571, 1993.

[68] Masaru Tomita, Kenta Hashimoto, Koichi Takahashi, Thomas Simon Shimizu, Yuri Matsuzaki, Fumihiko Miyoshi, Kanako Saito, Sakura Tanida, Katsuyuki Yugi, J Craig Venter, et al. E-cell: software environment for whole-cell simulation. *Bioinformatics (Oxford, England)*, 15(1):72–84, 1999.

[69] Boris M Slepchenko, James C Schaff, Ian Macara, and Leslie M Loew. Quantitative cell biology with the virtual cell. *Trends in cell biology*, 13(11):570–576, 2003.

[70] Bruce E Shapiro, Andre Levchenko, Elliot M Meyerowitz, Barbara J Wold, and Eric D Mjolsness. Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. *Bioinformatics*, 19(5):677–678, 2003.

[71] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. CopasiâĂŤa complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[72] Pawan Dhar, Tan Chee Meng, Sandeep Somani, Li Ye, Anand Sairam, Mandar Chitre, Zhu Hao, and Kishore Sakharkar. CellwareâĂŤa multi-algorithmic software for computational systems biology. *Bioinformatics*, 20(8):1319–1321, 2004.

[73] Michael Hucka, Andrew Finney, Herbert M Sauro, Hamid Bolouri, J Doyle, and Hiroaki Kitano. The erato systems biology workbench: enabling interaction and exchange between software tools for computational biology. In *Biocomputing 2002*, pages 450–461. World Scientific, 2001.

[74] Sharat Jacob Vayttaden and Upinder Singh Bhalla. Developing complex signaling models using genesis/kinetikit. *Science Signaling*, 2004(219):pl4–pl4, 2004.

[75] Seunghwa Kang, Simon Kahan, Jason McDermott, Nicholas Flann, and Ilya Shmulevich. Biocellion: accelerating computer simulation of multicellular biological system models. *Bioinformatics*, 30(21):3101–3108, 2014.

[76] T. Tian and K. Burrage. Parallel implementation of stochastic simulation for large-scale cellular processes. In *High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference on*, pages 6–pp. IEEE, 2005.

[77] I. Komarov and R. M D'Souza. Accelerating the gillespie exact stochastic simulation algorithm using hybrid parallel execution on graphics processing units. *PloS one*, 7 (11):e46693, 2012.

[78] Daniel G Harvey, Alexander G Fletcher, James M Osborne, and Joe Pitt-Francis. A parallel implementation of an off-lattice individual-based model of multicellular populations. *Computer Physics Communications*, 192:130–137, 2015.

[79] E. Kouskoumvekakis, D. Soudris, and E. S Manolakos. Many-core cpus can deliver scalable performance to stochastic simulations of large-scale biochemical reaction networks. In *High Performance Computing & Simulation (HPCS), 2015 International Conference on*. IEEE, 2015.

[80] M. Vanneschi. The programming model of assist, an environment for parallel and distributed portable applications. *Parallel computing*, 28(12):1709–1732, 2002.

[81] L. Dematté and T. Mazza. On parallel stochastic simulation of diffusive systems. In *International Conference on Computational Methods in Systems Biology*, pages 191–210. Springer, 2008.

[82] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press, 1999.

[83] R. Oudkerk J. Noller. Multiprocessing âĂŤ process-based âĂIJthreadingâĂİ interface. https://docs.python.org/2/library/multiprocessing.html.

[84] Nigel Gilbert. *Agent-based models*. Number 153. Sage, 2008.

[85] M. B Miller and B. L. Bassler. Quorum sensing in bacteria. *Annual Reviews in Microbiology*, 55(1):165–199, 2001.

[86] M. Weber and J. Buceta. Dynamics of the quorum sensing switch: stochastic and non-stationary effects. *BMC systems biology*, 2013.

[87] Y. Boada, A. Vignoni, and J. Pico. Promoter and transcription factor dynamics tune protein mean and noise strength in a quorum sensing-based feedback synthetic circuit. *bioRxiv*, page 106229, 2017.

[88] B. Bassler. Manipulating quorum sensing to control bacterial pathogenicity. *The FASEB Journal*, 29(1 Supplement):88–1, 2015.

[89] Yadira Boada, Alejandro Vignoni, JL Navarro, and Jesús Picó. Improvement of a cle stochastic simulation of gene synthetic network with quorum sensing and feedback in a cell population. In *Control Conference (ECC), 2015 European*, pages 2274–2279. IEEE, 2015.

[90] Victoria Moignard and Berthold Göttgens. Transcriptional mechanisms of cell fate decisions revealed by single cell expression profiling. *Bioessays*, 36(4):419–426, 2014.

[91] Francesco Fazi and Clara Nervi. Microrna: basic mechanisms and transcriptional regulatory networks for cell fate determination. *Cardiovascular research*, 2008.

[92] Wen Xiong and James E Ferrell. A positive-feedback-based bistable âĂŸmemory moduleâĂŹthat governs a cell fate decision. *Nature*, 426(6965):460–465, 2003.

[93] Katarzyna Anna Rejniak, Veronica Estrella, Tiangan Chen, Allison S Cohen, Mark Lloyd, and David L Morse. The role of tumor tissue architecture in treatment penetration and efficacy: an integrative study. *Frontiers in oncology*, 3:111, 2013.

[94] David J Smith. A boundary element regularized stokeslet method applied to cilia-and flagella-driven flow. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465, pages 3605–3626. The Royal Society, 2009.

[95] Edgar J Lobaton and Alexandre M Bayen. Modeling and optimization analysis of a single-flagellum micro-structure through the method of regularized stokeslets. *IEEE Transactions on Control Systems Technology*, 17(4):907–916, 2009.

[96] Heather Flores, Edgar Lobaton, Stefan Méndez-Diez, Svetlana Tlupova, and Ricardo Cortez. A study of bacterial flagellar bundling. *Bulletin of mathematical biology*, 67 (1):137–168, 2005.

[97] Bree Cummins, Tomáš Gedeon, Isaac Klapper, and Ricardo Cortez. Interaction between arthropod filiform hairs in a fluid environment. *Journal of theoretical biology*, 247(2):266–280, 2007.

[98] Hoa Nguyen, Ricardo Ortiz, Ricardo Cortez, and Lisa Fauci. The action of waving cylindrical rings in a viscous fluid. *Journal of Fluid Mechanics*, 671:574–586, 2011.

[99] Svetlana Tlupova and Ricardo Cortez. Boundary integral solutions of coupled stokes and darcy flows. *Journal of Computational Physics*, 228(1):158–179, 2009.

[100] Chin W Yong. Study of interactions between polymer nanoparticles and cell membranes at atomistic levels. *Phil. Trans. R. Soc. B*, 370(1661):20140036, 2015.

[101] Chunbai He, Yiping Hu, Lichen Yin, Cui Tang, and Chunhua Yin. Effects of particle size and surface charge on cellular uptake and biodistribution of polymeric nanoparticles. *Biomaterials*, 31(13):3657–3666, 2010.

[102] Abhilash Sasidharan, Jim E Riviere, and Nancy A Monteiro-Riviere. Gold and silver nanoparticle interactions with human proteins: impact and implications in biocorona formation. *Journal of Materials Chemistry B*, 3(10):2075–2082, 2015.

[103] Tommy Cedervall, Iseult Lynch, Stina Lindman, Tord Berggård, Eva Thulin, Hanna Nilsson, Kenneth A Dawson, and Sara Linse. Understanding the nanoparticle–protein corona using methods to quantify exchange rates and affinities of proteins for nanoparticles. *Proceedings of the National Academy of Sciences*, 104(7):2050–2055, 2007.

[104] Martin Lundqvist, Johannes Stigler, Giuliano Elia, Iseult Lynch, Tommy Cedervall, and Kenneth A Dawson. Nanoparticle size and surface properties determine the protein corona with possible implications for biological impacts. *Proceedings of the National Academy of Sciences*, 2008.

[105] Faryad Darabi Sahneh, Caterina Scoglio, and Jim Riviere. Dynamics of nanoparticle-protein corona complex formation: analytical results from population balance equations. *PloS one*, 8(5):e64690, 2013.

[106] Zhiyong Poon, Shujun Chen, Amanda C Engler, Hyung-il Lee, Evrim Atas, Geoffrey von Maltzahn, Sangeeta N Bhatia, and Paula T Hammond. Ligand-clustered âĂIJpatchyâĂİ nanoparticles for modulated cellular uptake and in vivo tumor targeting. *Angewandte Chemie International Edition*, 49(40):7266–7270, 2010.

[107] Pei-Hui Yang, Xuesong Sun, Jen-Fu Chiu, Hongzhe Sun, and Qing-Yu He. Transferrin-mediated gold nanoparticle cellular uptake. *Bioconjugate chemistry*, 16(3):494–496, 2005.

[108] Sutapa Barua and Samir Mitragotri. Synergistic targeting of cell membrane, cytoplasm, and nucleus of cancer cells using rod-shaped nanoparticles. *ACS nano*, 7(11): 9558–9570, 2013.

[109] Robby A Petros and Joseph M DeSimone. Strategies in the design of nanoparticles for therapeutic applications. *Nature reviews Drug discovery*, 9(8):615, 2010.

[110] Greg Michael Thurber and Ralph Weissleder. A systems approach for tumor pharmacokinetics. *PloS one*, 6(9):e24696, 2011.

[111] Michael I Monine and Jason M Haugh. Reactions on cell membranes: Comparison of continuum theory and brownian dynamics simulations. *The Journal of chemical physics*, 123(7):074908, 2005.

[112] Rachel A Kudgus, Chad A Walden, Renee M McGovern, Joel M Reid, J David Robertson, and Priyabrata Mukherjee. Tuning pharmacokinetics and biodistribution of a targeted drug delivery system through incorporation of a passive targeting component. *Scientific reports*, 4:5669, 2014.

[113] Twan Lammers, Fabian Kiessling, Wim E Hennink, and Gert Storm. Drug targeting to tumors: principles, pitfalls and (pre-) clinical progress. *Journal of controlled release*, 161(2):175–187, 2012.

[114] Raiyan T Zaman, Parmeswaran Diagaradjane, Sunil Krishnan, and James W Tunnell. Measuring gold nanoparticle concentrations in tissue using diffuse optical spectroscopy. In *Conference on Lasers and Electro-Optics*, page CMCC3. Optical Society of America, 2007.

[115] Faryad Darabi Sahneh, Caterina M Scoglio, Nancy A Monteiro-Riviere, and Jim E Riviere. Predicting the impact of biocorona formation kinetics on interspecies extrapolations of nanoparticle biodistribution modeling. *Nanomedicine*, 10(1):25–33, 2015.

[116] Shaolie S Hossain, Syed FA Hossainy, Yuri Bazilevs, Victor M Calo, and Thomas JR Hughes. Mathematical modeling of coupled drug and drug-encapsulated nanoparticle transport in patient-specific coronary artery walls. *Computational Mechanics*, 49(2): 213–242, 2012.

[117] Yonggang Wang, Jae-Hong Kim, Jong-Beom Baek, Gary W Miller, and Kurt D Pennell. Transport behavior of functionalized multi-wall carbon nanotubes in water-saturated quartz sand as a function of tube length. *water research*, 46(14):4521–4531, 2012.

[118] Yonggang Wang, Yusong Li, John D Fortner, Joseph B Hughes, Linda M Abriola, and Kurt D Pennell. Transport and retention of nanoscale c60 aggregates in water-saturated porous media. *Environmental science & technology*, 42(10):3588–3594, 2008.

[119] Eli Goldberg, Martin Scheringer, Thomas D Bucheli, and Konrad HungerbuÌĺhler. Critical assessment of models for transport of engineered nanoparticles in saturated porous media. *Environmental science & technology*, 48(21):12732–12741, 2014.

[120] James R Faeder, Michael L Blinov, and William S Hlavacek. Rule-based modeling of biochemical systems with bionetgen. In *Systems biology*, pages 113–167. Springer, 2009.

[121] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *International conference on concurrency theory*, pages 17–41. Springer, 2007.

[122] Dipak Barua, James R Faeder, and Jason M Haugh. Structure-based kinetic models of modular signaling protein function: focus on shp2. *Biophysical journal*, 92(7): 2290–2300, 2007.

[123] Dipak Barua, William S Hlavacek, and Tomasz Lipniacki. A computational model for early events in b cell antigen receptor signaling: analysis of the roles of lyn and fyn. *The Journal of Immunology*, page 1102003, 2012.

[124] Oksana Sorokina, Anatoly Sorokin, J Douglas Armstrong, and Vincent Danos. A simulator for spatially extended kappa models. *Bioinformatics*, 29(23):3105, 2013.

[125] Steven S Andrews. Smoldyn: particle-based simulation with rule-based modeling, improved molecular interaction and a library interface. *Bioinformatics*, 33(5):710–717, 2016.

[126] Juan R Perilla, Boon Chong Goh, C Keith Cassidy, Bo Liu, Rafael C Bernardi, Till Rudack, Hang Yu, Zhe Wu, and Klaus Schulten. Molecular dynamics simulations of large macromolecular complexes. *Current opinion in structural biology*, 31:64–74, 2015.

[127] Abdelghani Bellouquid, Elena De Angelis, and Damian Knopoff. From the modeling of the immune hallmarks of cancer to a black swan in biology. *Mathematical Models and Methods in Applied Sciences*, 23(05):949–978, 2013.

[128] Yawen Zou and Manfred D Laubichler. From systems to biology: A computational analysis of the research articles on systems biology from 1992 to 2013. *PloS one*, 13 (7):e0200929, 2018.

[129] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011.

[130] N Bellomo and B Carbonaro. Toward a mathematical theory of living systems focusing on developmental biology and evolution: a review and perspectives. *Physics of Life Reviews*, 8(1):1–18, 2011.

[131] Xiang-Hong Tan, Zhi-Bin Song, Hui Wang, Qin Wang, and Jian-Li He. Influence of adjuvant levetiracetam therapy on serum nerve cytokines and apoptosis molecules in patients with refractory partial epileptic seizure. *Journal of Hainan Medical University*, 23(11):145–149, 2017.

[132] Paolo Carpineto, Agbeanda Aharrh-Gnama, Vincenzo Ciciarelli, Enrico Borrelli, Francesco Petti, Raffaella Aloia, Alessia Lamolinara, Marta Di Nicola, and Leonardo Mastropasqua. Subretinal fluid levels of signal-transduction proteins and apoptosis molecules in macula-off retinal detachment undergoing scleral buckle surgery. *Investigative ophthalmology & visual science*, 57(15):6895–6901, 2016.

[133] Josep M Pujol, Joseph E Eisenberg, Charles N Haas, and James S Koopman. The effect of ongoing exposure dynamics in dose response relationships. *PLoS computational biology*, 5(6):e1000399, 2009.

[134] Abhishek Venkatratnam, John S House, Kranti Konganti, Connor McKenney, David W Threadgill, Weihsueh A Chiu, David L Aylor, Fred A Wright, and Ivan Rusyn. Population-based dose–response analysis of liver transcriptional response to trichloroethylene in mouse. *Mammalian Genome*, 29(1-2):168–181, 2018.

[135] Jiangyong Gu, Xinzhuang Zhang, Yimin Ma, Na Li, Fang Luo, Liang Cao, Zhenzhong Wang, Gu Yuan, Lirong Chen, Wei Xiao, et al. Quantitative modeling of dose–response and drug combination based on pathway network. *Journal of cheminformatics*, 7(1):19, 2015.

[136] Mohammad Soltani and Abhyudai Singh. Effects of cell-cycle-dependent expression on random fluctuations in protein levels. *Royal Society Open Science*, 3(12):160578, 2016.

[137] Michael L Blinov, James R Faeder, and William S Hlavacek. Rule-based modeling of biological systems using bionetgen modeling language.

[138] Walmor C De Mello. *Cell-to-cell communication*. Springer Science & Business Media, 2012.

[139] Jean-Claude Hervé and Mickaël Derangeon. Gap-junction-mediated cell-to-cell communication. *Cell and tissue research*, 352(1):21–31, 2013.

[140] Christopher M Waters and Bonnie L Bassler. Quorum sensing: cell-to-cell communication in bacteria. *Annu. Rev. Cell Dev. Biol.*, 21:319–346, 2005.

[141] Ilana Kolodkin-Gal, Ronen Hazan, Ariel Gaathon, Shmuel Carmeli, and Hanna Engelberg-Kulka. A linear pentapeptide is a quorum-sensing factor required for mazef-mediated cell death in escherichia coli. *science*, 318(5850):652–655, 2007.

[142] Sathish Kumar and Hanna Engelberg-Kulka. Quorum sensing peptides mediating interspecies bacterial cell death as a novel class of antimicrobial agents. *Current opinion in microbiology*, 21:22–27, 2014.

[143] S Yu Jessica and Neda Bagheri. Multi-class and multi-scale models of complex biological phenomena. *Current opinion in biotechnology*, 39:167–173, 2016.

[144] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3): 7280–7287, 2002.

[145] Gary An, Qi Mi, Joyeeta Dutta-Moscato, and Yoram Vodovotz. Agent-based models in translational systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(2):159–171, 2009.

[146] Megan M Olsen and Hava T Siegelmann. Multiscale agent-based model of tumor angiogenesis. *Procedia Computer Science*, 18:1016–1025, 2013.

[147] Le Zhang, Zhihui Wang, Jonathan A Sagotsky, and Thomas S Deisboeck. Multiscale agent-based cancer modeling. *Journal of mathematical biology*, 58(4-5):545–559, 2009.

[148] Michael W Sneddon, James R Faeder, and Thierry Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with nfsim. *Nature Methods*, 8(2):177–183, 2011.

[149] Dipak Barua and William S Hlavacek. Modeling the effect of apc truncation on destruction complex function in colorectal cancer cells. *PLoS computational biology*, 9(9):e1003217, 2013.

[150] Matthew S Creamer, Edward C Stites, Meraj Aziz, James A Cahill, Chin W Tan, Michael E Berens, Haiyong Han, Kimberley J Bussey, Daniel D Von Hoff, William S Hlavacek, and Posner G Richard. Specification, annotation, visualization and simulation of a large rule-based model for erbb receptor signaling. *BMC systems biology*, 6(1):107, 2012.

[151] Andre Levchenko and Ilya Nemenman. Cellular noise and information transmission. *Current opinion in biotechnology*, 28:156–164, 2014.

[152] Adam Humphries, Biancastella Cereser, Laura J Gay, Daniel SJ Miller, Bibek Das, Alice Gutteridge, George Elia, Emma Nye, Rosemary Jeffery, Richard Poulsom, et al. Lineage tracing reveals multipotent stem cells maintain human adenomas and the pattern of clonal expansion in tumor evolution. *Proceedings of the National Academy of Sciences*, 110(27):E2490–E2499, 2013.

[153] Abul K Abbas, Andrew H Lichtman, and Shiv Pillai. *Basic immunology: functions and disorders of the immune system*. Elsevier Health Sciences, 2014.

[154] Qian Qi, Yi Liu, Yong Cheng, Jacob Glanville, David Zhang, Ji-Yeun Lee, Richard A Olshen, Cornelia M Weyand, Scott D Boyd, and Jörg J Goronzy. Diversity and clonal selection in the human t-cell repertoire. *Proceedings of the National Academy of Sciences*, 111(36):13139–13144, 2014.

[155] Amar B Singh and Raymond C Harris. Autocrine, paracrine and juxtacrine signaling by egfr ligands. *Cellular signalling*, 17(10):1183–1193, 2005.

[156] Mariusz Z Ratajczak, Gabriela Schneider, and Janina Ratajczak. Paracrine effects of fetal stem cells. In *Fetal Stem Cells in Regenerative Medicine*, pages 47–56. Springer, 2016.

[157] Wouter-Jan Rappel. Cell–cell communication during collective migration. *Proceedings of the National Academy of Sciences*, 113(6):1471–1473, 2016.

[158] Y. Boada, Alejandro V., and J. Picó. Engineered control of genetic variability reveals interplay among quorum sensing, feedback regulation, and biochemical noise. *ACS synthetic biology*, 6(10):1903–1912, 2017.

[159] G. Rodola. psutil: cross-platform lib for process and system monitoring in python. https://pypi.org/project/psutil/.

[160] Radek Erban and S Jonathan Chapman. Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions. *Physical biology*, 6(4):046001, 2009.

[161] Haluk Resat, Michelle N Costa, and Harish Shankaran. Spatial aspects in biological system simulations. In *Methods in enzymology*, volume 487, pages 485–511. Elsevier, 2011.

[162] Tae-Hyuk Ahn, Pengyuan Wang, Layne T Watson, Yang Cao, Clifford A Shaffer, and William T Baumann. Stochastic cell cycle modeling for budding yeast. In *Proceedings of the 2009 Spring Simulation Multiconference*, page 113. Society for Computer Simulation International, 2009.

[163] Chandrasekaran Komalapriya, Despoina Kaloriti, Anna T Tillmann, Zhikang Yin, Carmen Herrero-de Dios, Mette D Jacobsen, Rodrigo C Belmonte, Gary Cameron, Ken Haynes, Celso Grebogi, et al. Integrative model of oxidative stress adaptation in the fungal pathogen candida albicans. *PLoS One*, 10(9):e0137750, 2015.

[164] Emalie J Clement, Beata J Wysocki, Ghada A Soliman, Tadeusz A Wysocki, and Paul H Davis. Dynamic modeling and stochastic simulation of metabolic networks. *bioRxiv*, page 336677, 2018.

[165] Nils A Kulak, Garwin Pichler, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nature methods*, 11(3):319, 2014.

VITA

Mohammad Aminul Islam was born in Dhaka, Bangladesh. He graduated with a Bachelor of Science in Chemical Engineering degree in June 2014 from Bangladesh University of Engineering and Technology. He was accepted and received funding as a graduate research assistant at Missouri University of Science and Technology in Spring of 2015. He was enrolled for a Ph.D. degree program in Chemical Engineering from 2015-2019. During his time in Missouri S&T, he acted as the vice president of the Language Club and executive member of Council of Graduate Studies. In July 2019, he received his Doctor of Philosophy degree in Chemical Engineering from Missouri University of Science and Technology (Rolla, Missouri, USA).