DOCTORAL THESIS

CMU-CB-21-100

in the field of

COMPUTATIONAL BIOLOGY

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Macromolecular Self-Assembly: Simulation and Optimization

Marcus Thomas

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Thesis Committee:
Russell Schwartz
James Faeder
Frederick Lanni
Timothy Lezon

This research was funded in part by the National Science Foundation grant MCB-1616492 and the National Institutes of Health grant T32EB009403. It was also supported by the Carnegie Mellon Computational Biology Department.

ACCEPTED:

THESIS COMMITTEE CHAIR DATE

DEPARTMENT HEAD DATE

APPROVED:

DEAN DATE

Macromolecular Self-Assembly: Simulation and Optimization

by

Marcus Thomas

Submitted to the Computational Biology Department
on Jan 4, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis develops computational methods for the investigation of self-assembly systems in biology as well as methods for the simulation of reaction-diffusion chemistry. We discuss the current state of the field with respect to modeling self-assembly and its importance to systems biology generally. Our contributions come in the form of pipelines for model inference based on comparisons of *in silico* experiments with physical experiments monitoring assembly progress. A new black-box parameter optimization methodology suitable for noisy objective values, and using multiple Gaussian processes, is presented. We also discuss the current landscape of course-grained simulation methods for reaction-diffusion chemistry and their limitations. A novel algorithm generalizing the stochastic simulation algorithm to continuous space is presented. We describe its physical justification as well as its improvements over the state of the art in certain respects, e.g. run time efficiency. At the end, we describe our applied work in collaboration with the Faeder and Murphy Labs (University of Pittsburgh and CMU, respectively) on an immune cell signaling project. While not directly related to self-assembly or the methods described previously, this collaboration allowed us to design a kinetic model from scratch and develop an optimization framework tailored to real experimental data.

Thesis Supervisor: Russell Schwartz
Title: Professor

Acknowledgments

There are many people that have helped me on this PhD journey in different ways. I have expressed my appreciation to most in person. Here I would like to give my thanks to my advisor, Russell Schwartz, for patiently helping me to learn, make mistakes and become a better scientist. I'd also like to thank the members of my committee - James Faeder, Frederick Lanni, and Timothy Lezon for their helpful discussions and feedback.

Contents

Chapter 1

Self-Assembly Systems

Molecular self-assembly is the dominant form of chemical reaction in living systems, yet efforts at systems biology modeling are only beginning to appreciate the need for and challenges to accurate quantitative modeling of self-assembly. Self-assembly reactions are essential to nearly every important process in cell and molecular biology and handling them is thus a necessary step in building comprehensive models of complex cellular systems. They present exceptional challenges, however, to standard methods for simulating complex systems. While the general systems biology world is just beginning to deal with these challenges, there is an extensive literature dealing with them for more specialized self-assembly modeling. This chapter will examine the challenges of self-assembly modeling, nascent efforts to deal with these challenges in the systems modeling community, and some of the solutions offered in prior work on self-assembly specifically. The chapter concludes with some consideration of the likely role of self-assembly in the future of complex biological system models more generally. [1]

1.1 Introduction

Self-assembly reactions account for the overwhelming majority of the reaction events occurring in the cell. Most eukaryotic proteins function normally in complexes and self-assembly of these complexes is a key step in nearly all major cellular functions [11]. Examples of processes critically dependent on self-assembly include genome replication [210, 274, 240, 28]; gene transcription and

[1] This chapter is based on work published here [265].

transcript degradation [28, 161, 181]; protein synthesis and degredation [162, 81]; cell movement and shape control [286, 73, 58, 116]; cell-to-cell communication including gap-junction assembly and regulation [261]; formation of membrane complexes such as pore-forming toxins [19]; and mechanotransduction [282, 288, 12]. Through these processes, the assembly and disassembly of molecular complexes and machines plays a crucial role in essentially all regulatory processes in cell biology. Given the centrality of self-assembly to cell biology, one cannot hope to develop truly comprehensive quantitative models of systems biology without tackling self-assembly. Yet self-assembly has until recently been largely absent from major efforts at developing general systems biology modeling tools (e.g., [209, 117, 88, 98, 253, 254, 157, 227, 258]) or handled only with one-off special cases for particular systems of importance (e.g.,[87, 283, 140]). Even the most ambitious efforts at large-scale biochemical modeling largely focus on traditional enzymatic chemistry or transcriptional dynamics and only implicitly model the self-assembly reactions involved in those processes (e.g., recent comprehensive models of whole-cell or whole-organism transcriptional and metabolomic modeling [262, 25]). This situation is beginning to change as some major systems biology tools (e.g., [104, 103, 83]) and modeling efforts [140] have begun to incorporate methods suitable to complex self-assembly, but major challenges remain.

These challenges of self-assembly modeling largely arise from the extremely large space of possible pathways accessible to the intermediate species of a self-assembly reaction network. The number of possible reaction trajectories by which a set of free monomers can assemble into a complex grows in general exponentially in the complex size, leading to an enormous combinatorial explosion in pathway space for even moderate-sized assemblies and astronomical numbers for large complexes, such as virus capsids or cytoskeletal networks. This is problematic for experimental study of assembly systems, as it is rarely possible to discriminate experimentally among these pathways except at a coarse level, particularly for highly symmetric or repetitive structures. It likewise creates problems for the most popular modeling methods. Mass action differential equation (DE) models are generally unsuitable for non-trivial assemblies because they require either extensive simplifications [105, 84, 179] or enormous numbers of equations and variables to account for the many possible intermediates [128]. Brownian dynamics (BD) models, even highly coarse-grained [234, 24, 79, 14], are likewise challenged by the large numbers of reactants and long timescales typical of self-assembly systems, requiring themselves great simplifications

of reaction processes that generally make them unsuitable for accurate quantitative modeling [86]. Methods based on Gillespie's stochastic simulation algorithm (SSA) can provide an effective balance between DE and BD, but face their own challenges because the underlying reaction networks are too large to model explicitly [39, 87, 94]. For similar reasons, self-assembly networks are extremely challenging for experimental characterization [42, 47, 132, 148, 203, 311, 312] and model inference as well [152, 298]. For example, the high computational cost and large numbers of intermediate species make it computationally infeasible to learn models via prevailing Bayesian parameter inference schemes [101], which require large numbers of simulation trajectories.

Over the recent decades, however, a specialized literature on self assembly modeling has grown for handling a number of challenging systems of independent importance. Cytoskeletal assembly (i.e., actin and microtubule assembly) has been the subject of extensive modeling work, leading to many seminal results in the basic biophysics of molecular assembly processes. Viral capsid assembly [105] has a long history as one of the primary model systems for macromolecular self assembly, both from an experimental and a computational perspective. Another key model system is amyloid aggregation, the basis for many major public health threats, including Alzheimer's disease, Huntington's disease, Parkinson's disease, prion disease, and type II diabetes. Fig. 1-1 shows a few examples of important model systems for self-assembly and models through which they have been studied. The practical importance of these and other systems has led them to attract their own modeling communities to find solutions to the special challenges of molecular self-assembly to computational modeling. In these fields, one can find studies both anticipating the challenges beginning to face broader systems biology efforts and often offering at least partial solutions to these challenges.

The remainder of this chapter will consider in more detail both the special difficulty of self-assembly modeling and the literature addressing it. It will first discuss some of the important roles of self-assembly in cellular biochemistry as well as the role of systems modeling methods in understanding these systems. It will then discuss some of the successful approaches to self-assembly modeling that have emerged through this literature, as well as continuing challenges. It will conclude with consideration of how quantitative self-assembly modeling may shape future efforts in modeling biological systems more generally.

Chapter 2 describes our work on viral capsid assembly systems. We develop a pipeline for

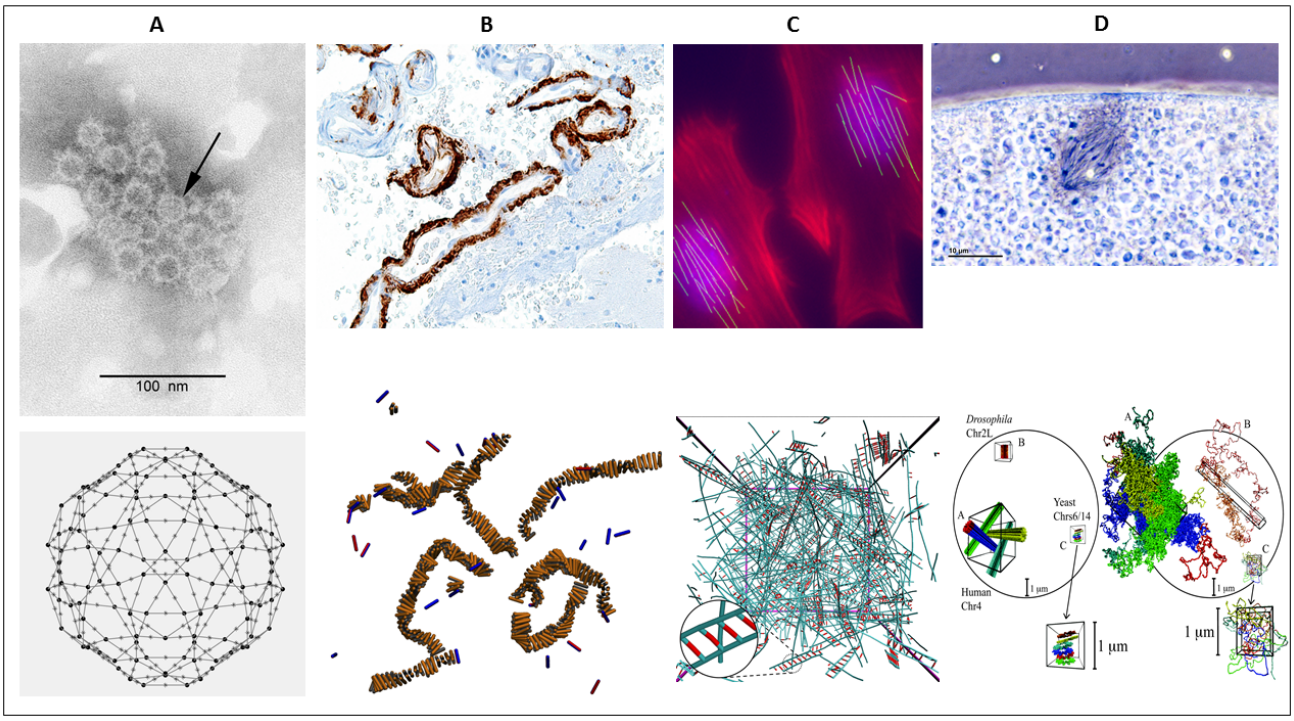


Figure 1-1: Example model systems for self-assembly simulation. (A) Viral capsid assembly. Top: Hepatitis virus [31], Bottom: Coarse-grained SSA simulation of HPV assembly [299]. (B) Amyloid aggregation. Top: High magnification micrograph of cerebral amyloid angiopathy with senile plaques in the cerebral cortex (amyloid beta, as seen in Alzheimer disease) [33], Bottom: Coarse-grained Monte Carlo simulation of amyloid aggregation with two state monomer model [18]. (C) Cytoskeletal assembly and disassembly. Top: Highly oriented actin fibers in shear stress cultivated rat cells [2], Bottom: BD simulation of actin cytoskeleton composed mainly of actin and actin crosslinking proteins [145]. (D) Genome organization. Top: Chromatin fibers during Mitosis, Xenopus egg [32], Bottom: BD simulations of nucleosome structure and dynamics [15]. For more on nucleosome assembly, see [28].

simulation based data fitting, i.e., for comparing experiments that track assembly progress with *in silico* experiments with constructed from assembly simulations in order to infer assembly model parameters. The simulations are primarily carried out using the stochastic simulation algorithm, though we also explore a differential equation model. In considering the limitations of these and other simulation methodologies, we were led to explore whether an entirely novel methodology could be more suitable to the requirements of self-assembly modeling. Chapter 3 describes the essential features of a new particle level spatial simulation algorithm which, with further development, may posses sufficient resolution and efficiency. Chapter 4 describes some tangentially related kinetic modeling and optimization work in the area of immune cell signalling.

Chapter 5 provides a summary and explores directions for future work.

1.2 The role of self-assembly in general cell biology

Self-assembly is everywhere in biology, beginning with the most fundamental processes of molecular biology, all of which depend on the self-assembly of specialized complexes, structures, or molecular machines. Examples of self-assembled molecular machines fundamental to molecular biology include DNA polymerases (replication), RNA polymerases (transcription), the spliceasome (splicing), the ribosome (translation), and the proteasome (protein degradation). Each of these processes is critical in different ways to the regulation of complex biological systems and thus has been the focus of specialized modeling efforts. For example, the transcription complex is one of the most well studied systems in molecular biology, with experimental work on the interaction of classic 1D and 3D diffusion of transcription factors [109] inspiring kinetic models of the recruitment process [139]. More specialized examples of self-assembly continue to be elucidated, with prominent recent examples including the RISC complex involved in miRNA [235, 168, 198, 126] and the Cas9-gRNA complex [56] implementing the CRISPR/Cas system [199, 259].

Within eukaryotic biology specifically, a more specialized set of self-assembly systems have evolved critical roles. The cytoskeleton is an unusually large, dynamic, and complicated molecular assembly, making it a crucial target of modeling efforts. The cytoskeleton itself is essential to intracellular transport [213, 214], cell movement and shape control [9, 212], mechanotransduction [287], and cell division [114], among many other functions. Furthermore, the dynamic process of assembly and disassembly is central to each of these functions. Actin and microtubule assembly and disassembly have been key model systems for self-assembly from the early days of molecular biology [129, 195, 89, 77, 238, 178, 291, 29, 90] and have inspired numerous computational models (e.g., [226, 190, 80, 251, 85, 228]). Transport processes in the eukaryotic cell frequently depend on other kinds of specialized self-assemblies, in addition to the cytoskeleton. For example, much eukaryotic transport involves the assembly of specialized machinery for construction and scission of cargo-carrying vesicles, such as the clathrin and COP-I/COP-II coat systems [197, 78], which have inspired their own modeling literature (e.g., [61, 171, 124]).

Beyond its role in general cell and molecular biology, self-assembly is crucial to a number of

disease-specific processes. Amyloid diseases are perhaps the prime example of a disease specifically of self-assembly, where aberrant assembly is the mechanism of illness. Numerous such diseases are known, including many major public health threats. Perhaps best known are Alzheimers (characterized by aggregates of the Aβ peptide and the Tau protein[176, 136]), Huntington's disease (characterized by aggregates of the Huntingtin protein [163]), Parkinson's disease [236], amytrophic lateral sclerosis [280], type II diabetes, and a variety of known prion diseases such as Creutzfeldt-Jakob [194, 45]. Alzheimer's and dementia, for example, are strongly associated with aging and affected roughly 36 million people in 2010 [296, 297]. It is becoming increasingly clear that the ability to form the amyloid state is a widespread, generic property of proteins [150] making the process of amyloidogenesis an important topic of theoretical study. From a physical perspective, the main question is what forces stabilize the aggregates into the oligomer (small soluble disordered clusters) and fibrillar (long, many-chain highly structured β-sheet-containing aggregates) states associated with neurotoxicity [232]. For a broader discussion of these forces, see [159, 158, 200]. From a computational perspective, the focus is both on identifying the structure of oligomeric intermediates and fibers but also elucidating the kinds of assembly pathways available. This is an especially challenging computational problem due to the intrinsic disorder in the system.

Viral illnesses form another broad class of self-assembly-driven illness, in which the assembly of large complexes (i.e., the viruses themselves) is the mechanism of the disease process. Virus assembly is of obvious medical importance, given the millions who die each year from viral illnesses, e.g., 1.5 million from AIDS alone [62]. A fundamental understanding of this crucial aspect of the viral life cycle and infectivity may offer avenues for therapeutics or vaccines [312]. Additionally, there many factors making viruses appealing to the modeling community, including the deep experimental literature on their assembly and a high degree of symmetry in the final structure that allows for large complexes to be produced from small numbers of distinct subunit types. Viral assembly modeling has thus become a subfield in itself. Virus assembly has been a crucial platform for many basic advances in self-assembly modeling, including the use of DE [309], BD [234, 206, 106, 185], and SSA [307, 113, 143] methods. It has likewise been a platform for developing a variety of specialized versions of these modeling methods, such as rule-based approaches to simulating extremely large reaction networks [127] and derivative-free optimization approaches to

model inference [152, 298]. Viral capsids have been a focus of intense theoretical study into the basic biophysics of self-assembly [105, 106, 70, 260, 34] as well as for identifying potential new avenues for assembly-mediated treatment [154, 53, 155, 260, 289, 304, 149, 71, 121, 239, 205].

1.3 The challenge of modeling of self-assembly reaction networks

At the root of much of the difficulty of modeling self-assembly is the extraordinarily large number of intermediates and pathways potentially accessible to a self-assembly system. Large number of reactants present problems in different ways to most conventional modeling and model inference methods (see Section 1.3.1 below). They likewise present a challenge to experimental characterization of such systems, as there is no practical way to monitor huge numbers of distinct molecular species. While details vary by geometry, in general the number of possible intermediates (partially built structures) one might encounter on the way to a complete assembly will blow up exponentially in the assembly size. This problem has probably been most intensively studied in the virus assembly literature, as it is particularly pronounced for large, highly symmetric structures, of which viral capsids are a prime example. Even a coarse-grained model of an icosahedral virus capsid, consisting of just twelve subunits, has 750 possible intermediate structures [174]. For real viral structures, which typically have several hundred proteins, the numbers of potential intermediates will be astronomical. Similar problems will arise to a lesser degree with large, asymmetric assemblies (e.g., the ribosome [151, 180]) as well as with larger but less symmetric assemblies such as the cytoskeleton. While the number of species possible for a linear filament is small, once one allows for branching [305], numbers of possible branched filaments or networks can blow up exponentially in the structure size as well. Note that this is not a unique challenge of self-assembly, as similar issues arise in other combinatorially explosive systems, such as signaling networks [23, 115].

A related concern for modeling, particularly with respect to self-assembly in cell biology, is the issue of small copy numbers [91, 273] resulting in an inherently discrete and stochastic reaction system. The issue occurs for many cellular systems involving reactants that occur in

just a few copies per cell, but is especially an issue for self-assembly because the large number of intermediates guarantees that most are present in zero or one copies at any given time [196]. The issue is exacerbated by the fact that self-assembly reactions are frequently nucleation-limited, meaning that they are characterized by slow and relatively rare nucleation events followed by comparatively rapid polymerization. Nucleation-limited growth is well established for several of the major model systems in self-assembly, such as virus capsids [203, 306], amyloids [156], and actin and tubulin fibers [21]. A large body of theory suggests the nucleation-limited growth is crucial to their robust operation [222, 204, 203, 72, 268, 201]. In nucleation-limited systems, nearly every species is unpopulated at most times. Small copy numbers are problematic computationally in part because they mean that discretization errors inherent to efficient continuum models became substantial. In part, they are problematic because they mean that self-assembly must be treated as a stochastic system, forcing the use of less efficient simulation methods than the continuum approximations usable when all species are well populated [91, 273, 68] (see Section 1.3.3).

A second major challenge of self-assembly reactions is their long timescales (see Figure 1-2), and in particular the large gap between timescales of the full assembly reaction and the individual polymerization steps of assembly. Full assembly reactions of large complexes in vitro may have timescales measured in minutes to days (although assembly in vivo may be substantially faster [167, 60, 243]) while individual reaction steps are typically many orders of magnitude faster [252, 311]. In part, this is a side effect of nucleation-limited growth mentioned above: nucleation reactions are necessarily much slower than the subsequent elongation reactions [234, 310]. Furthermore, the nucleation reactions themselves may in fact require extensive trial-and-error involving much faster formation and breakdown of transient partial intermediates [310, 306, 257]. Large timescales, and a large dynamic range of timescales, are challenges for essentially all standard modeling methods, whether that manifests in a need for large numbers of timesteps in a continuum method or large numbers of discrete events for a stochastic simulation.

A third class of challenge arises from the fact that self-assembly reactions are unusually sensitive to the many ways in which the physical biology of the cell differs from that of in vitro models. For example, physical confinement — by the cell membrane, subcellular compartments, or other large structures such as the cytoskeleton or genome — is commonly neglected in modeling

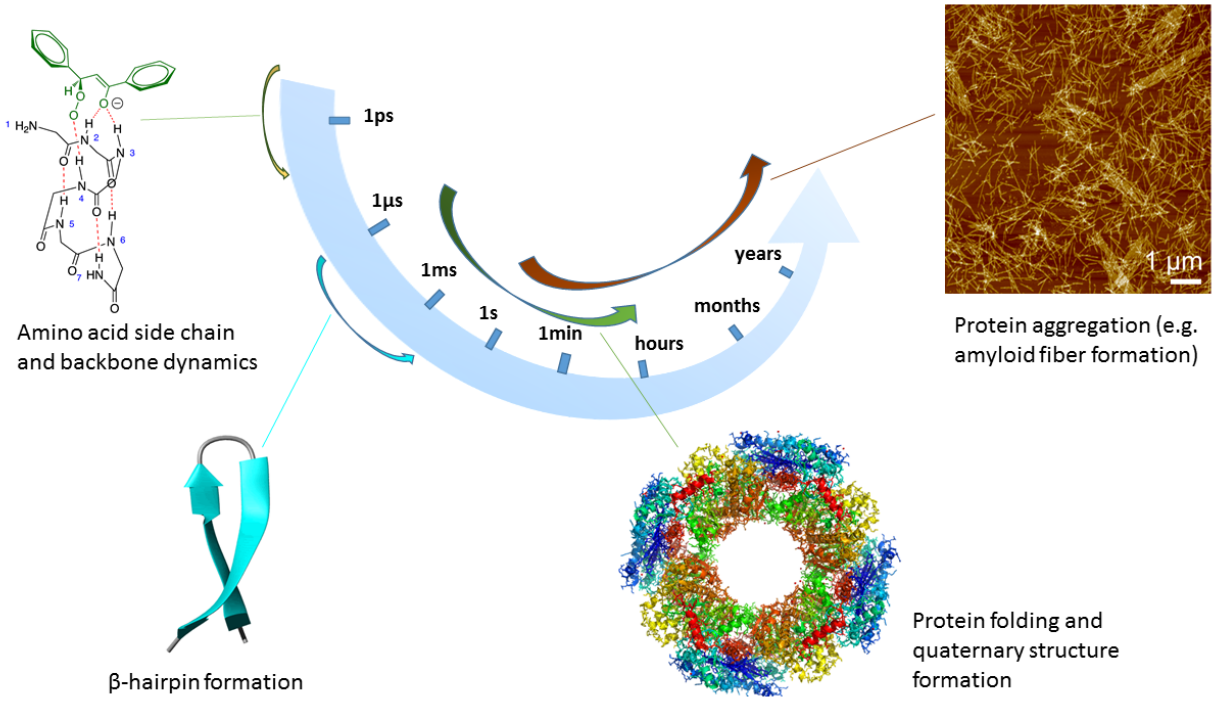


Figure 1-2: Timescales for protein dynamics and aggregation. The figure illustrates some of the basic biological processes applicable to self-assembly and their relevant timescales. It is based on material from [8, 30, 294, 293].

reaction systems yet cannot be ignored when dealing with reactions that result in products comparable in size to the spaces in which they form. A related issue is that self-assembly processes are also well known to be unusually sensitive to macromolecular crowding [211, 173, 110], a key distinguishing feature of the cellular environment. Numerous theoretical and experimental studies have suggested both the need for and the challenge to correcting simulation methods to account for the effects of crowding on assembly processes (e.g., [186, 233]). Examples include the effects on several aspects of DNA replication such as helicase activity and the sensitivity of DNA polymerase to salt [1], on protein-protein binding affinity and specificity [146], on the kinetics and morphology of amyloid self-assembly [167], on the stochasticity of gene expression machinery [111], and on viral capsid assembly [243, 60].

1.3.1 Modeling methodologies

Despite the difficulties they present to modelers, a variety of modeling methods have proven valuablefor self-assembly. Table 1.1 describes a few of the primary methods that have emerged for self-assembly modeling. While most are drawn from older techniques for more general reaction chemistry modeling, in the self-assembly context they often present novel challenges or require specialized adaptations. This section covers three of the most successful methodologies that have been developed for self-assembly, some of the particular challenges they have faced in the self-assembly context, and how they have been adapted to meet those challenges.

Table 1.1: Common modeling methodologies for self-assembly. The table lists principal techniques for self-assembly modeling, some systems biology software packages implementing them, and some notable applications in self-assembly modeling.

Reaction Representation	**Description**	**Software Packages**	**Applications**
Law of Mass Action (Deterministic)	Expresses any well-mixed chemical system as a collection of coupled non-linear first order differential equations which typically must be numerically integrated. PDEs must be used when space is explicitly included	BioNetGen [22], COPASI [117], VCell [209], DBSolve [99]	Virus Assembly: [179, 241, 42, 105], Metabolomic Networks[128]
SSA/Gillespie Approaches	Provides a way to simulate kinetically correct trajectories consistent with the Chemical Master Equation	Moleculizer [165], BioNetGen [22], VCell [209], DESSA [307]	Virus assembly: [257, 143]
Spatial Stochastic	Usually combine Gillespie or Stochastic Langevin with diffusion or subunit geometry	MCell [253], StochSim [157], VCell [209], Smoldyn [6], SRSim [104]	Geometric Constraints with Diffusion: [103], Amyloid-Beta: [269]
Rule-Based	Primarly network-free rule-based methods which may incorporate stochasticity and spatial modeling	RuleMonkey [55], BioNetGen, ML-Space [20], VCell [231], SRSim [104]	Multivalent ligand-receptor interactions: [302], Prion Aggregation [219], Virus Assembly: [234, 307]
Brownian Dynamics	An explicitly spatial model where brownian motion is computed with the Langevin equation	Smoldyn [6], MCell [253]	Multiscale Reaction-Diffusion [86], Virus Assembly: [234, 106, 185, 74, 75, 24], Crowding/Amyloids: [292], Clathrin Cage Formation: [124]

Mass action differential equation (DE) models

Much modeling of reaction systems classically has arisen, at least initially, from DE models based on the chemical Law of Mass Action. Such models represent any generic chemical reaction

network

$$a_{11}X_1 + a_{12}X_2 + \ldots + a_{1n}X_n \underset{k_r}{\overset{k_f}{\rightleftharpoons}} b_{11}X_1 + b_{12}X_2 + \ldots + b_{1n}X_n$$

in terms of a system of differential equations of the form

$$\frac{dX_i}{dt} = (k_{i+}b_{1i} - k_{i-}a_{1i})X_1^{a_{11}}X_2^{a_{12}}\cdots X_n^{a_{1n}} + \ldots$$

$$+(k_{i+}b_{mi} - k_{i-}a_{mi})X_1^{a_{m1}}X_2^{a_{m2}}\cdots X_n^{a_{mn}}$$

Accumulating these contributions across a full set of reactions and reactant species defines a system of differential equations modeling the time evolution of all reactants in the system. Such DE models were the basis of many of the earliest cell simulation systems, such as E-cell [270], ProMoT/Diva [98], Virtual Cell [229], GEPASI [172] and others. Later extensions of these models allowed for consideration of spatial heterogeneity via partial differential equation (PDE) reaction-diffusion models:

$$\frac{\partial X_i}{\partial t} = d_i\nabla^2 X_i + (k_{i+}b_{1i} - k_{i-}a_{1i})X_1^{a_{11}}X_2^{a_{12}}\cdots X_n^{a_{1n}} + \ldots$$

$$+(k_{i+}b_{mi} - k_{i-}a_{mi})X_1^{a_{m1}}X_2^{a_{m2}}\cdots X_n^{a_{mn}}$$

for reactant-specific diffusion coefficients d_i.

DE models provided a basis for some of the first approaches to modeling many self-assembly systems. Classic results on molecular assembly of polymers derived from such models include [192, 191] and they were integral to seminal models of microtubule polymerization [63]. They likewise were used for early attempts at more complex systems, such as the first dynamic models of viral self-assembly [312, 310], where they provided early insights into the parameter space of self-assembly [310]. They continue to prove valuable in that context for such problems as interpreting complex experimental data [105, 241, 42]

The most substantial challenge to DE models on self-assembly systems is computational tractability, as such models need to keep explicit track of all species that might be present in a given simulation. While that number grows only linearly in assembly size for linear polymers, it blows up exponentially in size for more complex structures such as viruses. In practice, the

solution to that problem has typically been to simplify: either manually via simplified versions of structures or conflation of subsets of structures [309, 310] or through automated methods for pruning low-usage pathways [76]. While there is good empirical evidence that such strategies can yield quantitatively accurate models [312, 76], degrees of accuracy can be sensitive to structure and pathways used [174]. DE models further provide no good solution for the problem of modeling discretization of small copy number reactions.

1.3.2 Brownian dynamics (BD) models

The challenges self-assembly modeling presents to DEs led to an alternative approach based on Brownian dynamics (BD) particle models. In a BD approach, we explicitly model a finite set of assembly subunits in three dimensional space. These subunits diffuse through space under a model of Brownian motion, implemented by a variant of damped Langevin dynamics [79]. Models of binding dynamics can be implemented either by discrete reactions occuring upon particle collisions or via short-range binding forces, leading to gradual agglomeration of particles over the course of a simulation. BD models have the considerable advantage over DE models that one need not devote computational resources to any species not present at a specific instant in time. Run time thus depends on the number of particles modeled, not the number of species they might in principle form.

Such models have perhaps been most pronounced in their use with viral capsid systems, perhaps because their exceptionally large space of intermediates makes them especially challenging for DE models. Through viral capsid work, they have been the basis of numerous important insights into the basic biophysics of self-assembly. BD models were introduced to capsid studies nearly two decades ago [234], have seen a series of important methodological advances since [206, 106, 185] and continue to be the basis of new approaches and applications (e.g., [14, 79, 24]). They have also seen important roles in modeling various other challenging assembly systems, such as clathrin [124]. Insights arising from BD models include understanding the importance of nucleation limited growth to ensuring robust assembly and preventing kinetic trapping [108], the sensitivity to numerous parameter variations [74], and the potential sources of misassembly [74, 106]. In more recent years, these models have been extended to issues difficult to model with other methods, such as understanding the role of the genome in RNA virus assembly [75].

The advantages of BD methods, however, come with some significant tradeoffs. First, the large size and long timescale of assembly reactions generally requires substantial structural simplifications. Second, such models typically can accommodate only modest numbers of particles, ranging up to a few thousand per simulation for state-of-the-art methods [24, 107, 220]. For relatively large structures, that may be too few to capture more than a small fraction of possible assembly trajectories. Third, they generally cannot produce quantitatively correct assembly rates, because of the large gap between diffusion rates and assembly rates. Effectively, systems need to be shifted into domains of extremely rapid assembly, through unrealistically high binding rates or concentrations, in order to yield computationally tractable simulations of the complete assembly process. Some more advanced versions of this approach can somewhat mitigate these issues, for example the use of Green's function reaction dynamics (GFRD) to reduce the computational time needed to compute trajectories of particles between collision events [278].

1.3.3 Stochastic simulation algorithm (SSA) methods

Just as BD models were introduced to self-assembly modeling to address the weaknesses of DE models, so have models based on the stochastic simulation algorithm (SSA) [91] (also known as "Gillespie models" after their inventor) been adopted to address the weaknesses of BD models. In an SSA model, we represent a system at an instant in time by discrete counts of molecular species (monomers or partial assemblies). Simulation progress proceeds via reaction events, which for a self-assembly system will largely consist of single binding or dissociation reactions. Classically, one assumes a uniform, well-mixed system, in which reaction times can be approximated with exponential waiting time distributions [91]. The SSA approach can also accommodate spatial heterogeneity through modeling as an array of well-mixed, discretized spatial compartments, a variant known as spatial SSA, e.g. [255, 7].

SSA models offer considerable advantages but also involve important tradeoffs with the previously considered methods. They can be implemented to have run times independent of the number of potential species, unlike DE models, and can thus handle arbitrarily large reaction networks [94]. However, their run time does depend on the number of discrete particles present, limiting them to finite numbers of protein copies, as do BD models. They are, however, typically much more efficient than comparable BD models since they do not need to model diffusion

explicitly [94] and are practical over a much broader range of parameter domains [257]. In addition, they provide an explicit quantitative model yielding kinetically correct samples from a set of reactions and associated rate constants. However, because they do not explicitly model space, they do not easily handle steric constraints that are important to such processes as aberrant assembly [234], interaction of proteins with a flexible genome [75, 303], or any form of continuous flexibility in proteins or complexes [217, 116].

SSA methods have needed some special adaptations to deal with the challenges of self-assembly. Probably the most important advance is the use of rule-based modeling (e.g., [246]), a strategy independently developed for the self-assembly field under the name local rule modeling [234] and later introduced to SSA models under that name [307]. Rule-based models allow one to avoid explicitly constructing the reaction network, an infeasible task for all but trivial self-assembly systems, and rather represent only the current state of the system and its immediate neighbors [82, 307]. This reduces run time from dependence on the size of the network to dependence only on the number of species and reactions present at any instant in time. While steric constraints are a challenge for rule-base models, that challenge has been overcome for some systems, e.g., in modeling multivalent ligand-receptor interactions [177]. Further improvements to queuing methods for discrete event implementations of SSA [127, 64] made it possible to accelerate run time by eliminating quadratic time/memory dependencies in the standard algorithms. Additionally, a set of more specialized theory has been developed to deal with the problem of extreme divergence between timescales of monomeric reactions versus the complete assembly process. Generic methods for accelerating SSA can be helpful, e.g., [208, 38], as well as more specialized variants specifically for self-assembly [175]. Other improvements include hybrid methods combining SSA with ideas from agent-based modeling [3].

While SSA methods have not yet seen as wide use as BD in the self-assembly field, they have proven to have important applications for which neither DE nor BD methods are suitable. Because of their ability to handle complex geometries and long time scales, SSA models have proven valuable for exploring parameter dependence of assembly systems by making it practical to sample large numbers of trajectories over long time scales [299] and to sample trajectories from particularly complex geometries or pathway sets [143]. They have also become a valuable platform for fitting models to experimental data, where their ability to fit an explicit timescale,

to function over wide parameter ranges, and to model complex geometries are all crucial features [299, 298, 243, 244].

1.4 Self-assembly in broader systems biology modeling

In recent years, efforts at systems biology modeling have begun increasingly to recognize the importance of self-assembly to comprehensive modeling of complex biochemical systems. For example, a number of general systems biology simulation tools have begun to incorporate handling of self-assembly in various ways. An early example was Moleculizer [165], which incorporated basic models of assembly reactions via a rule-based SSA model with special purpose corrections accounting for altered diffusion rates of growing species. Similar kinds of models have become important more generally in modeling tools, such as RuleBender [301], which have made it possible to integrate similar rule-based SSA models into other tools for systems biology modeling. The Virtual Cell [230] has recently added handling of self-assembly reactions, using a special-purpose extension based on a form of coarse-grained BD models of self-assembly [6, 59], as well as explicit handling of rule-based modeling[231]. The most recent version of the E-Cell [270] simulation environment (ECell4) has also been updated to include capabilities for modeling self-assembly such as a network-free rule model [82] and a spatial SSA method [255]. While none of these systems yet incorporates all of the specializations found in such methods in self-assembly specific contexts, they represent important steps towards generic tools for modeling complex reaction networks that include but are not specific to self-assembly.

This need for handling the kind of combinatorial explosive reaction network that characterizes self-assembly is also beginning to be reflected in systems biology language design. For example, the Systems Biology Markup Language (SBML) [120, 119], which has become the de facto standard for specifying models in systems biology, has been updated in more recent versions to accommodate the kind of network-free rule-based models needed for self-assembly work [118]. While it has long been possible to generate SBML from a rule specification through external tools, such as BioNetGen [83], native support of the modeling language is necessary to achieve the benefits of network-free modeling needed to make complicated self-assembly modeling tractable. Handling of steric constraints that become imporant in formation of more complicated assemblies

remains a hard problem for the field, however, and is so far handled only in more specialized self-assembly simulation languages [307].

Recent years have also seen claims of the first true whole-cell simulations [140, 223], an effort that necessarily involves modeling numerous processes that depend on self-assembly. In practice, such efforts have not relied on a general-purpose simulation engine suitable to both self-assembly and more conventional reaction chemistry, favoring instead general purpose methods ill-suited to self-assembly coupled to special-purpose handling of particular kinds of self-assembly. The landmark work of Karr et al. [140] establishing a comprehensive simulation of M. genitalium biochemistry, relied on a series of special-purpose modules, several of which involved ad hoc methods for specific examples of self-assembly, such as macromolecular complexation and ribosome assembly. Nonetheless, even this kind of special-purpose handling remains the exception in similar efforts at comprehensive modeling of whole-cell reaction networks (e.g., [262, 25]).

Self-assembly is a greatly important but long neglected issue in the quantitative modeling of biological systems. While it is conventionally seen as a specialized form of chemistry, it is in fact the dominant form of reaction in living systems. It poses distinctive challenges for modeling methods, though, that prevailing methods in systems biology cannot handle. Self-assembly modeling has, however, been studied intensively in many more specialized contexts, leading to an appreciation of these challenges and a variety of ways they can be addressed. As more general systems biology efforts are beginning to embrace the necessity of accommodating self-assembly, this specialized literature can provide guidance and at least partial answers to some of the biggest obstacles these efforts will encounter. This chapter was intended to provide a brief overview of the particular challenges of self-assembly modeling, how they have been approached to date, and how these methods have been used in the past and are beginning to be incorporated into comprehensive models of systems biology. Our hope is that better awareness of obstacles and solutions already identified by self-assembly modelers can assist the broader systems modeling community in anticipating and navigating the same issues.

An appreciation for the past literature allows us to predict some of the future paths comprehensive systems modeling efforts are likely to follow. For the most part, where general efforts systems biology modeling has considered self-assembly, it has been as special cases with special-purpose methods for specific systems (e.g., [24, 75, 299, 242, 224, 131, 146, 250, 220, 14, 308, 298]). Given

the many examples of self-assembly in cell biology, it is safe to say this is not a sustainable solution; rather general systems biology efforts will need to start to think of self-assembly as the normal case that must be accommodated and integrated into simulation design via both model specifications and simulation algorithms. Modeling methods that will work for both self-assembly and for other kinds of chemistry exist [20, 283, 3, 209, 22, 165, 258, 55], but will need to become the standard for modeling tools and languages. More foresighted efforts in a variety of systems modeling contexts can help point the way (e.g., [165, 270, 6, 140, 59, 231]), although most remain behind the state-of-the-art in modeling of self-assembly specifically.

At the same time, there are many challenges for which good solutions do not yet exist. For example model inference [50] remains an extremely difficult problem for self-assembly systems [312, 152, 299, 298], where the Bayesian methods usually favored by the field [101] are unusable in practice, and it is likely advances in both biotechnology and inference algorithms will be needed to address it. The field is beginning to tackle this challenge, e.g., with BioNetFit [263], and its genetic algorithm to provide curve fitting capabilities compatible with ODE (BioNetGen) and Network Free (NFSim) model specifications. Promising results were presented for steady-state dose-response and time series oligomerization data, though it is not clear how the method copes with more complicated assembly dynamics. There are also, as yet, no universally good methods for modeling hard self-assembly systems. Each of the major approaches covered here — SSA [94], BD [234], and DE [309] — has tradeoffs that make them unsuitable for some questions. It remains to be seen whether more general solutions might arise from advances in one or more of these methods, clever hybrid approaches, or some wholly new ideas. It is also worth noting that self-assembly systems are challenging to characterize experimentally, for similar reasons to their challenge to modelers. The solutions to that issue, as well, are likely to lie in pooled efforts by experimentalists and computational researchers to advance experimental biotechnology and model-fitting algorithms in complementary ways. Indeed, self-assembly may be a particularly valuable test case for addressing the hard problems in building detailed and predictive quantitative models of complex biological systems, where the field can begin to think of modelers and experimentalists not as two communities but as two inseparable pieces of the future practice of biological discovery.

1.5 Contributions of this Thesis

Chapter 2

Viral capsid assembly has a long history as a model system for investigating properties of self-assembly systems. Since the 1950s there has been intense effort to understand capsid structure (e.g., Watson & Crick's 1958 *The structure of small viruses*, and Caspar & Klug's 1962 *Physical principles in the construction of regular viruses*) and much experimental work to elucidate details assembly pathways. In the 1990s, Berger et al. [16] put forward a general computational theory for assembly able to account for a wide range of experimental and theoretical findings. We adopt this local rule theory and develop a pipeline for parameter inference using simulation based data fitting. To briefly summarize, we simulate parameterized local-rule models of protein subunits assembling into complete capsids. From these simulations, *in silico* experiments are generated mathematically and compared to "real" experimental data. An optimization algorithm then searches for the parameter set resulting in the closest match between the simulated and real experiments. Our main contributions were: algorithms for generating small angle scattering data from assembly, and a novel Bayesian optimization framework based on representing the objective function with Gaussian processes.

Chapter 3

The second major project was initially motivated by the need for more spatial realism in our self-assembly pipeline. Many of the existing spatial methods were not suitable for the long time scales and large number of reactions required to simulate assembly. Our main contribution here is a novel algorithm for the efficient simulation of reaction-diffusion systems - a generalization of the stochastic simulation algorithm to continuous space. This is only a first stage towards a method capable of handling the challenges of self-assembly reaction chemistry. However, initial results show improved efficiency over more established methods in a benchmark reaction-diffusion test.

Chapter 4

The last major project was a collaboration with James Faeder's Lab at the University of Pittsburgh and the Robert Murphy's Lab at Carnegie Mellon University. Our contribution was the development of a kinetic model capable of explaining the spatiotemporal patterns of a number of proteins involved in immune cell signaling. This data, derived from fluorescence intensities, came in the form of probabilities for finding a given protein in each of the roughly 7000 voxels in a standardized cellular volume. We also designed an optimization algorithm tailored to this data set and to the differential equation system used in its simulation.

Chapter 2

Optimization of a Rule-Based Assembly Model

2.1 Background

Self-assembly chemistry is an essential part of nearly every important function of a living cell, yet has long proven exceptionally challenging due to their large sizes, long time scales, and explosive pathways spaces. Simulations can provide a way to examine details of assembly unavailable to direct experimental observation, but are computationally demanding for complex assemblies leading to a body of specialized simulation methods specifically for simulating molecular self-assembly chemistry [309, 234, 206, 127, 307, 106, 113, 143, 185, 174]. Furthermore, learning parameters needed by these simulations is itself a very difficult problem for assembly systems, likewise requiring specialized methods [1].

We previously showed that it is possible to simulate realistic scales and parameter ranges of complex self-assembly reactions, with specific focus on virus capsid assembly as a model system, by using coarse-grained, rule-based models [234]. These rule-based models were originally implemented via Brownian particle models [234, 206, 106, 185] and later via fast stochastic sampling algorithms [307, 113, 143], approaches that have since seen widespread use in modeling capsids and other complex reaction systems. Accurately parameterizing such models from experimental data, though, remains challenging. Standard methods for model fitting in biochemistry, particularly the

[1] This chapter is based on work published here: [266]

Bayesian model-inference methods that have become the favored approach in the field [295], are unusable for non-trivial self-assembly systems due to their exceptionally high computational cost, large pathway space, and inherent stochasticity [265]. In past work, we showed that it was possible to learn detailed quantitative parameters of these models via simulation-based model fitting to static light scattering (SLS) measurements of bulk assembly in vitro [299, 152], primarily by bringing to bear specialized optimization techniques from the field of Derivative-Free Optimization (DFO) [57]. Together, these contributions made it possible to infer the subunit-level pathway space of real capsids assembling in vitro, which in turn can be applied to explore how pathway usage might differ under more realistic models of the intracellular environment [243, 244]. The reliability of such inferences is uncertain, however, due to limits of the data in precisely and uniquely identifying a specific model and the difficulty of accounting for model uncertainty with these classes of methods. The present work focuses on improving parameter fitting methods in terms of potential experimental techniques to which one can fit models and parameter inference algorithms that can be applied for the fitting.

Computationally, we seek to bring to self-assembly the advantages of Bayesian model inference in exploring the space of possible solutions.

We approach the problem of quantifying uncertainty in parameter estimation by constructing a probabilistic model of the objective function using Gaussian process (GP) models [207]. This GP method is a variant of a technique called kriging [147] that has previously proven valuable in other contexts for solving computationally demanding model inference problems under uncertainty. GP models are defined by mean and covariance (kernel) functions, and specify a prior on the space of possible functions. As simulations at successive parameter values are completed, the prior is updated, forming the posterior which is used in prediction. New data points for sampling are selected based on the current properties of the process and user defined trade-offs between exploration and exploitation of the parameter space. This iterative non-parametric Bayesian approach is better able to handle uncertainty in parameter assignments than our previously used optimization techniques which were based on local surrogate functions. The GP formalism also allows for predictions at test points using global information about the smoothness and self-similarity of the objective.

We simultaneously seek to expand the repertoire of data sources to which these methods can

be applied, with specific focus on moving from the static light scattering (SLS) of prior work to small angle X-ray/neutron scattering (SAXS/SANS). SAXS has already proven valuable for reconstructing kinetics of capsid assembly systems (e.g., SV40 VP1 pentamers encapsidating short RNA molecules [148], and distinguishing closed shells from incomplete intermediates during P22 assembly [272]) while SANS has been applied to similar reconstruction problems of other protein assemblies, such as the Huntington amyloid [252]. Time resolved SAXS has also been used to study the dynamics of conformational change in viruses [36, 160, 170].

Here, we develop and implement our GP optimization framework and demonstrate it using synthetic SAXS data of known ground truth. We implement both stochastic (SSA [91]) and deterministic (ordinary differential equation (ODE) [309]) models of virus-like assembly systems of known parameters. We then demonstrate that we can accurately reconstruct the original models from simulated SAXS data derived from these systems. While our stochastic and deterministic models are applicable to virus like assembly systems, the parameter inference framework is quite general and can be expected to be appropriate to any system for which model predictions are costly to evaluate and noisy.

2.2 Overview and Objective

The overall goal of our method is to learn a set of model parameters, specifically kinetic rate constants for distinct self-assembly reaction events, that define a quantitative model of assembly which is maximally consistent with a set of experimental data. We assume the data here to be SAXS or SANS waveform data, explained below, which we will canonically reference as SAXS data. We particularly develop a class of methods designed to learn a stochastic process which is utilized to obtain an optimal assembly model specification. The process probabilistically models an objective function quantifying the difference between a ground truth SAXS experiment, for which we have (synthetic) data, and a candidate experiment determined from a simulation trajectory at a single hypothetical point in parameter space.

We define our objective function as the root mean square deviation (RMSD) between the respective sets of intensity curves over designated time points. We generalize this objective for use with two model types in common use in this field, SSA-based stochastic models and ODE

continuum models, each of which can work with the same basic inference framework with some specialized modifications. For the stochastic assembly framework, no two trajectories' reactions will occur at the same time points. We can get around this problem because assembly is a Markov process in which the system state remains unchanged between any two successive reactions. Thus, for each time point in the ground truth experiment, we select the closest later time point from the candidate experiment when computing the RMSD. Within the continuous time ODE framework, we can directly specify the time points to consider via interpolation relative to a finite difference numerical integration. We note that we do not generalize to coarse-grained Brownian particle models despite their widespread use, because they cannot be parameterized straightforwardly in terms of reaction rate constants like SSA and ODE models.

Our central task in model inference is to approximately minimize this objective as efficiently as possible. Note that our use of stochastic processes to represent the objective means we are effectively specifying a probability density over possible rate parameters. This contrasts with our prior work [300, 152, 299, 243] fitting the rate parameters directly or with conventional Bayesian optimization that directly samples over the parameter space. While this may seem a complicated approach, this complication is the key to kriging methods gaining the advantages of a Bayesian model in estimating model uncertainty while simultaneously getting the high efficiency needed by the application. Noise variance in the stochastic case is significant and no single assembly trajectory, or resulting SAXS experiment, can be taken as representative for a given model specification. We determine the representative by simulating multiple trajectories, translating each into a SAXS experiment, and then taking their element-wise mean. The objective's empirical noise level is also approximated by computing the RMSD of each repeated simulation and calculating their variance directly.

2.3 Data Sources

Scattering experiments consist of a wave source (x-rays/neutrons for SAXS/SANS, respectively) directed towards a sample. After interacting with the sample medium, some fraction of the incident waves scatter away while the remainder are absorbed. The intensity of scattered radiation is measured at a detector as a function of the scattering vector, q. Small-angle scattering roughly

corresponds to measured intensities at low q values, allowing the investigation of microscopic features with spatial resolution ranging from a few angstroms to a few microns. Mathematically, the scattered intensity I(q) is the Fourier transform of the electron density correlation function, therefore signal is observed only if the contrast in electron density is different from zero. However, scattering experiments do not provide localized information about the sizes, shapes and pairwise distances of the molecular constituents. Instead, the intensity is representative of the entire sample, providing a spatial and temporal average over the duration of the measurement (temporal resolution as low as 100ps [46]). Due to these limitations, scattering provides only bulk, indirect evidence of assembly dynamics. See [100, 46] for a detailed treatment. There are also numerous examples of small-angle scattering with other protein systems, e.g., [281, 225, 35, 130, 102, 141, 65].

In silico SAXS experiments are constructed from simulated assembly trajectories by extrapolating the solution scattering of a single protein subunit obtained from CRYSOL. CRYSOL [256] is a program for evaluating the solution scattering from macromolecules with known atomic structure and accepts as input PDB structure files. The present work focuses on fixed structures for the dimer subunits of cowpea chlorotic mottle virus (CCMV) formed from the $A-B$ and $C-C$ chains (PDB 1za7, [249]), as well as a model of the pentamer subunits of generic dodecamer assembly. [51] examines the energetic effects of allowing the subunits to come from a distribution of possible configurations rather than a single PDB structure.

CRYSOL ouputs a vector of scattering intensities corresponding to q values from 0 to 0.5 in steps 0.01. As the higher q values correspond to observations of smaller features of the system, possibly beyond experimental reliability, there is a question as to the correct range to consider. However, in the context of our purely computational experiments we will not consider the issue and use the default range returned by CRYSOL.

The full SAXS intensity is determined as a function of the form factor, $F(q)$, and the structure factor, $S(q)$. Loosely speaking, the form factor is determined by the internal structure of the elementary particles in the system (i.e., the protein subunits), and the structure factor provides information on larger scale spatial correlations among the elementary particles.

$$I^{SAXS}(q) = \Delta\rho^2 V^2 |F(q)|^2 S(q) \tag{2.1}$$

In Eq.2.1, V is the elementary particle volume, and $\Delta\rho$ the electron density contrast between particle and solution. These two terms, together with the form factor, are returned by CRYSOL as the single subunit scattering. Because our simulators do not directly model diffusion through space, we do not have any information on the relative positions of the various intermediates present at each reaction step. Our mathematical extrapolation of the subunit scattering to the full system scattering therefore relies on a dilute assumption, allowing the scattering contributions of each intermediate to be summed. Within the context of a single intermediate, we do have access to relative subunit positions and so we calculate a structure factor for each.

$$S(q) = \frac{1}{N}\Sigma_{j,k}e^{-iq(R_j - R_k)} \tag{2.2}$$

In Eq.2.2, i is the imaginary unit, $\sqrt{-1}$, and the summation is over every pair, (j,k), of the N subunits present in the intermediate, located at positions R_j, R_k.

The full waveform, $I^{SAXS}(q)$, specified as a function of q over a defined range and step size, serves as the input to our model inference. One might in principle fit to multiple waveforms for a given system, for example from monitoring assembly at distinct concentrations, although for simplicity we assume here that we are fitting to a single SAXS experiment.

2.4 Stochastic Simulation Model

The simulation-based data fitting approach used here depends on fitting a model to a data set through an intermediate simulation. That is, one assesses quality of fit of a parameter set based on how well the true experimental data matches simulated experimental data, derived by simulating assembly with the parameter set and then generating SAXS/SANS data from the output of that assembly simulation. In the present work, the "true" data is also simulated, which is necessary to have a data set with known ground truth. We implement two versions of the full pipeline here, one for a stochastic simulation class and one for a deterministic one, in each case using the same techniques for creating true data and for fitting to those data.

Stochastic simulations are run using DESSA [307] which implements a version of the Gillespie algorithm [91] for coarse grained, complex self-assembly systems [127]. Reaction chemistry is

represented as a continuous time Markov model of the possible reaction trajectories (see [307] for details) available to an initial collection of protein subunits that undergo association and dissociation reactions according to a local rule model [16, 234]. The local rules describe interactions between protein subunits in terms of the positions, affinities, and kinetics of their binding sites, with the binding rate constants the only free parameters. This simulator does not explicitly model diffusion in space, nor is it based on a lattice or compartment model. Instead, the intra-capsid geometry is modeled through the local rules, with diffusion implicitly a function of the kinetic rates under the assumption that the system is well mixed.

Because DESSA deals directly with expected wait time constants for reactions (T) rather than reaction rate constants (k), conversion between the two is useful. The unimolecular case (e.g., dissociation of species $S_1 : S_1 \rightarrow 2S_1$) is easy. The reaction rate k_{uni} has units $\frac{1}{[s]}$, so the expected wait time T_{uni} is simply the inverse of the rate constant. The bimolecular (e.g., association: $S_1 + S_2 \rightarrow S_1 : S_2$) molar reaction rate constant is defined as $k_{molar} = \frac{N_A * \Omega}{T_{bi}}$, where Ω is the system volume, N_A is Avogadro's number and T_{bi} is the expected reaction waiting time, again allowing one to derive the rate constant from the inverse of the waiting time and vice versa.

Finally, because the search space can span multiple orders of magnitude along each dimension, we work in log-scaled units. We specifically treat the ground truth (GT) parameter values T_{GT} as the origin of CCMV's 10-dimensional parameter space (corresponding to the 10 subunit binding sites), with the conversion between real space (T) and log-scaled space (x) given by $\mathbf{T} = 10\hat{}(log(\mathbf{T}_{GT})/log(10) + \mathbf{x})$. Search points are constructed by the algorithm as modifications of the ground truth point which is located at $\mathbf{x} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. In all results figures, the $\mathbf{x}$ values of points are displayed, rather than their $\mathbf{T}$ values.

2.5 ODE Model

Simulating assembly using ODEs requires a distinct differential equation describing the time evolution of the concentration of each potential intermediate, from monomeric subunit to complete capsid. We began, like Endres & Zlotnick [76] and Misra & Schwartz [174], by considering a model of $T = 1$ assembly from pentameric capsid subunits, for which only monomer/monomer and monomer/oligomer reactions are possible. We justify this restriction of the pathway space

based on the observation that, except in cases of extremely high rates or concentrations [308], the equilibrium concentration of monomers is much larger than that of the intermediates. With this simplification, we were able to compute each species in the assembly tree (all of the structurally unique partial assemblies of a given size), their forward/backward reaction degeneracies, and their relative stabilities. Following [174], we represent the molar concentration of the k^{th} unique species of size j as [j,k], and represent the forward reaction degeneracy between (j,k) and (m,n) after monomer addition as $a_{j,k}^{m,n}$. The corresponding backward reaction degeneracy is $b_{m,n}^{j,k}$. The relative stability of (j,k) w.r.t. (m,n) is approximated as

$$s_{j,k}^{m,n} = exp(-\Delta G * (c_{m,n} - c_{j,k})/RT) \tag{2.3}$$

were $c_{j,k}$ is the number of bonds formed within species (j,k).

The differential equation for the time evolution of $[j,k]$ is as follows.

$$\begin{aligned}\frac{d[j,k]}{dt} &= k_{on}\Sigma_{m,n}(b_{m,n}^{j,k}s_{j,k}^{m,n}[m,n] - a_{j,k}^{m,n}O(m-j)[j,k][m-j]) \\ &- k_{on}\Sigma_{p,q}(b_{j,k}^{p,q}s_{j,k}^{p,q}[j,k] - a_{p,q}^{j,k}O(j-p)[j-p][p,q])\end{aligned} \tag{2.4}$$

In Eq.2.4, $O(m-j)$ denotes the symmetry of the monomer subunits which satisfy $m = j+1$ (or oligomer subunits if $m - j >= 1$ were allowed). The full set of these equations for all defined j and k define an ODE model for time evolution of the complete reaction system.

For the equations to be correct, it is necessary to identify assemblies that are isomorphic to one another, a special case of the graph isomorphism problem. While a general algorithm for detecting isomorphism of subsets of icosahedral assemblies is provided in [174], we provide here an efficient variant customized for this application. Our new algorithm for identifying all structurally unique intermediate oligomers and computing the forward/backwards degeneracies for each relevant pair is shown in Figure 2-1. It iteratively constructs the state space by adding a pentagonal monomer to each free binding site of the current oligomer, and tests the resulting oligomer set for isomorphism. Only the unique structures are saved, i.e., those which are not pairwise isomorphic under some transformation in SO(3). For each isomorphic structure generated, the appropriate $a_{j,k}^{m,n}$ is incremented. The isomorphism testing subroutine is outlined in Figure 2-2. It relies on

the fact that without loss of generality, we can enforce that all species contain the same initial monomer (which we call 'face 1'), and that the implicit dodecahedron - of which all oligomers are a part - can be oriented relative to a fixed location in space. For convenience, the centroid of face 1 is treated as the fixed location. There are 11 3D rotations leaving the dodecahedron in an orientation equivalent to the original, but which successively place each face in the fixed location. Further, for each of these orientations, due to the pentagonal symmetry of the subunits, there are 4 2D rotations leaving the centroid of the face in the fixed location unchanged. When determining if two oligomers are isomorphic, each of the 12*5 orientations of the first oligomer are computed successively and the resulting coordinates are compared with the second oligomer for identity. The isomorphism subroutine runs in time $O(|F| * |E/F|)$ where $|E/F|$ is the number of edges per face. Finally, we note that many viruses possess icosahedral symmetry. Being dual to the dodecahedron and sharing the same symmetry group, the same set of rotations apply to the icosahedron.

Figure 1 Identifying distinct intermediates

```
Compute origin centered dodecahedron as graph G(V,E,F)
        % V: coordinates of the 30 vertices.
        % E: pairs of indices in V denoting edges.
        % F: 5-tuples of indices in V denoting the 12 pentagonal faces.

Provide arbitrary fixed label for each face (i.e. 1,2,...12).
Compute centroid of each face.
        % Note the location of say, face 1 centroid.
        fixedLocation ← centroid(face1)
Compute list of potential binding partners of each face

% Iterate through species sizes.
for n = 2 : | F|

        if n == 2, test dimers for isomorphism, update a_{j,k}^{m,n}. else
        for each unique oligomer of size n-1
                Construct all possible oligomers of size n, s.t.
                no two share the same face list.

                Save unique subset of these oligomers after
                testing for isomorphism.
```

Figure 2-1: Pseudocode for identifying distinct intermediates.

Figure 2 Testing isomorphism subroutine

```
% Inputs: oligomer 1 and {oligomer set}
for R in {set of | F| *| F/E | rotations}

        Compute all orientations, R*(oligomer 1)

        Consider only rotated oligomers with a face in the fixedLocation

        If any of the rotated oligomers' coordinates
        match any of those in {oligomer set}

                return the member of {oligomer set}
```

Figure 2-2: Pseudocode for determining if oligomers are isomorphic.

2.6 Modeling the Objective as a Gaussian Process

Gaussian processes have a long history of use in disparate fields for related tasks including interpolation and prediction. For example, in geostatistics it has been known as kriging since the early 1970s [207]. The modern interpretation is that a GP assigns a probability distribution to a space of functions, the most important properties of which (e.g., smoothness) are determined by the GP covariance function. Due to its non–parametric nature, overfitting is less of a concern than it is with other regression models.

In analogy with the Gaussian distribution, the GP prior is completely defined by its mean function and covariance function [207].

$$F(x) \sim GP(m(x), k(x, x')) \tag{2.5}$$

The covariance function, $k(x, x')$, is of central importance. It defines a notion of similarity between points in the input space in terms of their objective values, enabling prediction at test points. It is technically a kernel function and must be symmetric. Further, when this kernel function is evaluated at a set of points, the resulting matrix must also be positive semidefinite. Combining the prior with new observations (i.e., training on pairs $\{\mathbf{x},F(\mathbf{x})\}$) leads to the posterior distribution over functions, representing our updated beliefs about possible candidate functions.

Once the form of the mean and covariance functions are specified initially, training is synonymous with covariance *hyper*parameter optimization (note that the model is still nonparametric

because the hyperparameters define a class of GPs rather than a particular instance). In other words, the hyperparameters obtained are those that minimize the negative log marginal likelihood of the data under the GP class specified by the covariance function. This optimization step is usually very efficient and should not be confused with the larger optimization problem of minimizing the objective function.

Gaussian Process Optimization

At a high level, the method will work iteratively by using the GP to identify candidate parameter sets at which to run additional simulations, which in turn are used to refine the GP fit. After training, the GP can be queried for the *best* new expensive point(s) to evaluate. This model includes uncertainty at test points, as it not only provides an estimate of the mean value of the objective, but also provides an estimate of the variance. Assuming the input space has not yet been thoroughly explored, the algorithm will benefit from objective evaluations at additional parameter points. In identifying candidate points to simulate, we need to balance exploration of unexplored (high variance) areas with exploitation of regions known to have low objective values. Several *acquisition functions* (AF) have been designed to handle this tradeoff at the expense of yet another (usually inexpensive relative to $F(\mathbf{x})$) optimization. We have chosen the lower confidence bound (LCB) as the acquisition function to be minimized due to it's simplicity of evaluation.

$$a_{LCB}(\mathbf{x}) = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x}) \tag{2.6}$$

$$\mathbf{x}_{new} = argmin_x \ \ a_{LCB}(\mathbf{x}) \tag{2.7}$$

In Eq.2.6, μ denotes the mean prediction at each input and σ the corresponding standard deviation. The user defined parameter κ balances the tradeoff (higher and lower for exploration and exploitation, respectively). It has been shown that choice of statistical model is often far more important than choice of acquisition function [237]. Other popular choices include *probability of improvement (PI)*, *expected improvement (EI)*, *entropy search*, and *Thompson sampling* [247, 237].

AF minimization can be achieved using derivative free optimization packages such as DIRECT [133, 27], SNOBFIT [123] and MCS [122], or methods such as sequential quadratic programming

and quasi-Newton solvers. We, however, chose a simpler approach. We draw samples in relevant areas of the search space, evaluate the AF, and directly select the minimizer. This randomized procedure is repeated many times with the resulting set of minimizers coordinate-wise averaged.

Multi-GP Model Optimization

In many regression contexts, domain-specific knowledge can be applied to constrain the class of statistical model used to fit the experimental data. For example, it may be known from physical principles that observations should be distributed linearly with some corruption from random measurement error, suggesting the use of a linear regression model. In our case, GP regression allows a great deal more flexibility in principle but we lack prior knowledge about which, if any, GP class accurately models the process generating a particular set of observations. With sufficient training data, the kernel maximizing the likelihood of those data while also predicting the correct noise level is often the best choice. However, our focus is optimization of the objective with as few function evaluations as possible. A novelty of the proposed method is the assumption that using multiple statistical models for experimental data generation at once, rather than in a one-off fashion, may allow us to more efficiently discover structure, e.g., the locations of local minima. Additionally, during early rounds of search, this strategy can provide an avenue for more thoroughly exploring the input space since the acquisition functions corresponding to different kernels may be minimized by different points.

1. Matern 3/2 (ARD): $\sigma^2(1+\sqrt{3}\sqrt{r}) * exp[-\sqrt{3}\sqrt{r}]$
2. Matern 5/2 (ARD): $\sigma^2(1+\sqrt{5}\sqrt{r}+(5r)/3) * exp[-\sqrt{5}\sqrt{r}]$
3. Rational Quadratic (ARD): $\sigma^2(1+r/(2\alpha))^{-\alpha}$
4. Rational Quadratic (ISO): $\sigma^2(1+s/(2\alpha))^{-\alpha}$
5. Gabor (ARD): $h(x_1-x_2); h(t) = exp[-\sum(t^2/P^2)] * cos[2\pi\sum(t/p)]$
6. Neural Network: $\sigma^2 \arcsin[x_1^T P x_2/\sqrt{(1+x_1^T P x_2) * (1+x_1^T P x_2)}]$
7. Square Exponential (ARD): $\sigma^2 exp[-r/2]$
$r = (x_1-x_2)^T * P^{-1} * (x_1-x_2); s = (x_1-x_2)^T * (\ell * I)^{-1} * (x_1-x_2)$
P is the diagonal matrix of ARD lengthscale hyperparameters.
ℓ is a scalar lengthscale hyperparameter; I is the unit matrix.
α is a shape hyperparameter for the rational quadratic kernel.
p is a vector of period hyperparameters.
In the Gabor kernel, operations are performed element-wise.

Table 2.1: Kernel Functions, $k(x_1, x_2)$

The kernel functions used in the present work are listed in Table 2.1. We sought to include a range of traditional kernels as well as a few less common choices. All except the Neural Network (NN) covariance are stationary in the sense of depending on the relative difference $\mathbf{x} - \mathbf{x}$' rather than on the absolute locations in parameter space. The Square Exponential (SE) covariance leads to extremely smooth candidate functions (i.e., infinitely differentiable). This smoothness may not be realistic for objective functions associated with physical processes, but it is the most widely used in machine learning. The Matern class covariance with hyperparameter $\nu = 7/2$ (not used) leads to candidate functions very similar to the SE. As ν moves through $5/2$ and $3/2$, the respective candidate functions become rougher. Values of ν below $3/2$ are not recommended for regression, and non half-integer values lead to very complicated forms for $k(x, x')$. The Rational Quadratic (RD) kernel can be viewed as a mixture of many SE kernels, each with a distinct lengthscale hyperparameter, and is very general. The NN covariance allows us to perform regression with the equivalent of an infinitely wide single layer network using the error function as the hidden unit. The Gabor covariance enables the discovery patterns in the data which incorporate some periodicity, and extrapolation based on the pattern. Our decision to include these kernels and not others is somewhat arbitrary beyond the fact that they impose a diversity of assumptions on data generation. Future work may consider more principled methods for the number and types of kernel to be used.

Figure 2-3 (A) visualizes the main aspects of our global optimization method during a single round of search and (B) illustrates how the optimization fits into the overall data set preparation and parameter inference pipeline.

2.7 Results

Gaussian process model specification and hyper-parameter optimization was performed using code released by Rasmussen, Nickisch, Williams and Duvenaud [40, 69].

Stochastic Simulation Model Results

We begin the model fitting by sampling a selection of points in parameter space uniformly at random from a hypersphere, a contrast to our earlier methods that begin with a regular grid

search [300] that is motivated by prior work showing random sampling to be more efficient when the objective surface has low effective dimensionality compared to the parameter space [17]. For the present experiments, the hypersphere is centered on the ground truth point. For each sampled point, we run a set of simulation trajectories, project SAXS outputs, and compute the associated RMSDs relative to the input data. The resulting data points are then used in initial GP kernel hyperparameter training, updating the prior over objective functions to a posterior. In subsequent rounds of search, the posterior density estimated by the GP from the previous round becomes the new prior density from which we select further parameter points for evaluation to produce an updated posterior.

To provide a comparison with a more traditional solver, we used SNOBFIT (Stable Noisy Optimization by Branch and FIT) [123] a Matlab-based solver that combines a branching strategy with localized quadratic response surface fitting for fast, continuous optimization of black box functions satisfying a number of technical and design criteria. We favor SNOBFIT based on prior work showing it to be effective on capsid assembly simulation [299]. Its major advantage over competitor methods, including early stochastic process based methods such as DACE and SPACE [221, 135] as well as more traditional iterative modeling methods such as DIRECT and UOBYQA [134, 202], is its ability to handle all of the following cases: function values are expensive to evaluate; function values may be available only at approximately the requested points; the function values are noisy; the objective is non-convex; no gradients are available; there are hidden constraints; there are soft constraints; parallel function evaluation is desired; function values may be obtained extremely infrequently; and, the objective function or the search region may change during optimization. In the present work, the comparison is in terms of the number of function evaluations necessary to recover the ground truth parameter vector. Each time SNOBFIT is called, it uses function evaluations from previous rounds as well as newly evaluated points to return a user-specified number of function minimizers to be evaluated evaluated in the next round. These minimizers belong to one of 5 classes: 1-3 being local estimates, and 4-5 global estimates. Plots indicate the local/global classification of each returned point.

We first show results of a search of a small parameter space, corresponding to a hypersphere of radius 3 logs around the ground truth. For this search, we used an initial sample of 100 points, with 300 trajectories per point sampled. Figure 2-4 shows RMSD as a measure of search

progress for our method (top/middle) and for SNOBFIT (bottom). The κ parameter listed for our method balances the degree to which the search favors exploration of uncertain regions (higher κ values) versus exploitation of low variance regions. After one round of optimization, no kernel is clearly superior in discovering the correct parameter set. After two rounds, in each case one of the seven kernels shows a near optimal fit, although it is surprisingly a different kernel for each choice of κ. The three best scoring points are displayed in Figure 2-4 (middle) with 95% confidence intervals on each dimension. These intervals are kernel-dependent and so we used the kernel responsible for recovering the point in their calculation. To estimate confidence intervals in a particular dimension, we use the GP model to sample a series of RMSD values from a sequence of points in the vicinity of the chosen optimum in that dimension, with the spacing for the sequence determined in part by the applicable kernel lengthscale hyperparameter, l. Specifically, we scanned a region of $20l$ at a density of $0.001l$. The regression provided us with predictions of $\mu(RMSD)$ and $\sigma^2(RMSD)$ at each point in the sequence from which we drew 10,000 random samples to estimate the fraction of times each point in the sequence would be predicted to yield the optimum RMSD. We then chose the minimal symmetric window of points around the optimum so as to account for 95% of the probability density of minimum RMSDs, providing an estimated 95% confidence interval.

SNOBFIT with default settings was unable to recover low RMSD parameter sets from either its locally weighted (classes 1 and 3) or global (class 4) optimizations after seven rounds. Figure 2-5 shows another measure of search progress, the distance between predicted parameter points and the ground truth parameter set. Here we can see that the same points that approximately minimized the RMSD are also in fact close to the ground truth.

We further sought to compare the results to a more conventional kriging search by evaluating how well the method would perform using only a single kernel. Figure 2-6 are search results for which only a single kernel is used across all rounds of optimization, with the same 100-point training set as in Figs. 4 and 5. In the multi-GP search, it was the Matern 3/2 kernel that discovered the lowest RMSD ($1.6 \times 10^{10} \pm 2.2 \times 10^{11}$) point after 121 total function evaluations. In the single-GP searches, the Gabor-ARD kernel was able to obtain the slightly lower value of $0.7 \times 10^{10} \pm 2.2 \times 10^{11}$ after only 106 total function evaluations. The remaining searches produced minima in the range 5.2×10^{10} – 2.4×10^{11} with similar noise levels. We can conclude that while

a single kernel may lead to slightly better performance when seeking the minimum of a particular objective function, it is not obvious how to select this kernel beforehand, or to what extent the choice depends on the particular training examples seen. The multi-kernel approach does nearly as well as the best single-kernel approach without the need for advanced knowledge of how to select an appropriate kernel for a particular system.

We next consider a search of a larger space, corresponding to a hypersphere of radius 9 logs using 71 initial training points and 100 trajectories per point sampled. For this search, we allowed the optimization to run for 20 rounds. The results show fairly high concordance among solutions, although with high variability in estimates of parameters p7 and p10. The results suggest the method is effective at finding low-RMSD solutions, although as might be expected, the solutions are sensitive only to a subset of the parameters. Figure 2-7 summarizes search progress to this point.

As the predictions for each kernel begin to repeat round after round (e.g., beginning shortly after the 150th function evaluation in Figure 2-7), it may be useful to re-evaluate points the algorithm deems good. As more simulated experiments are averaged at a point, the corresponding objective becomes more accurate, potentially allowing better discrimination between similarly good points and better generalization at nearby points. We have settled on a number of criteria for the selection of the most useful set of points to re-evaluate with more simulations. First, the set should be a subset of previously evaluated points. This allows the utilization of previously run simulations. Second, no two points should be "close" (as defined by a kernel's lengthscale hyperparameter). Third, the set should prioritize higher scoring points.

These criteria suggest a selection subroutine analogous to agglomerative, hierarchical clustering, with the highest scoring cluster representatives chosen for re-evaluation. The resulting set of points was limited to 16 and is shown in Figure 2-8. See Figure 2-9 for pseudocode of the selection subroutine. In this case, re-evaluation of each of the 16 points with 1000 additional simulated experiments did not alter their relative ordering. It is interesting to note that for the smaller search space, the lowest RMSD points tend to also be closest to the ground truth in Euclidean distance. However, for the larger search space, the lowest RMSD are among the furthest from ground truth. This is again an illustration of the fact that the objective function is not equally sensitive to changes in each dimension. To obtain more precise estimates of the global minimum,

one strategy would be to begin new small (e.g., hyperspheres of radius 3 logs) searches at low noise levels, and centered successively on each of the top 16 points.

ODE Model Results

We next examined the utility of the solver for deterministic optimization using an ODE model of capsid assembly represented as a dodecamer, as in [76]. Here, we follow the assumption that each step in the oligomerization reaction may have an independent rate, but equating all oligomers of a given size. That is, we assume there is a single oligomer of size N that has a defined rate of transition to size $N+1$, but allow that the transition from N to $N+1$ may have a different rate than that from N' to $N'+1$ for $N \neq N'$. We here examine two cases: a 6 parameter model (grouping [1,2],...[11,12]) and the full 12 parameter model in which each oligomer has its own on-rate. We arbitrarily define the ground truth for the differential equation model to be a parameter vector in which each element (reaction rate) has the value 100 (in real space as opposed to the log space used in the stochastic simulations) and we conduct the parameter search in a hypersphere of radius 100 around this ground truth value.

Figures 2-10 and 2-11 show the results of the six and twelve parameter models. Each subfigure shows RMSD as a function of the number of search rounds for each kernel. We note that the ground truth in each case has an RMSD of exactly zero, yet moving a small distance away necessitates a minimal RMSD in the realm of 10^7 due to the way SAXS experiments are evaluated. The objective surface is roughly constant in a neighborhood surrounding ground truth, with a very steep descent in its immediate vicinity. Thus, we should expect the accuracy of the approximate global minimizer to depend on the size of this surrounding neighborhood. In each assembly model (6 or 12 parameters), different kernel functions are able to identify this neighborhood with varying amounts of training data.

To provide comparison to a competitive existing black box global minimizer, we use Multilevel Coordinate Search (MCS) [122], a more appropriate choice than SNOBFIT when solving for a deterministic objective. MCS is based on the DIRECT method and can be classified as *branch without bound* in the sense that it sequentially partitions the search space. As an improvement on DIRECT, the balance between global and local search is handled through a multilevel approach (partitioning the space along a single coordinate only). The method is guaranteed to converge if

the objective is continuous in the neighborhood of a global minimizer. Because MCS is designed as a MATLAB caller, taking the black box function as an input, we were not able to easily asses its performance in terms of the number of function evaluations. Rather, it runs until convergence (or a stopping criteria is met) and outputs the minimizer, objective value, number of function evaluations, number of function evaluations used in local search, and other algorithm parameters. Figure 2-12 shows the results of MCS searches of increasing search space size. The ground truth is again a 12D vector with each element 100. The plot shows that MCS performs well when we have relatively tight bounds on the global minimum, in fact far better than our GP method, but poorly when those bounds are relaxed. A good strategy for solving deterministic systems of similar dimension may therefore be to narrow down the search region using the GP approach, and then apply MCS for a more accurate solution.

2.8 Discussion

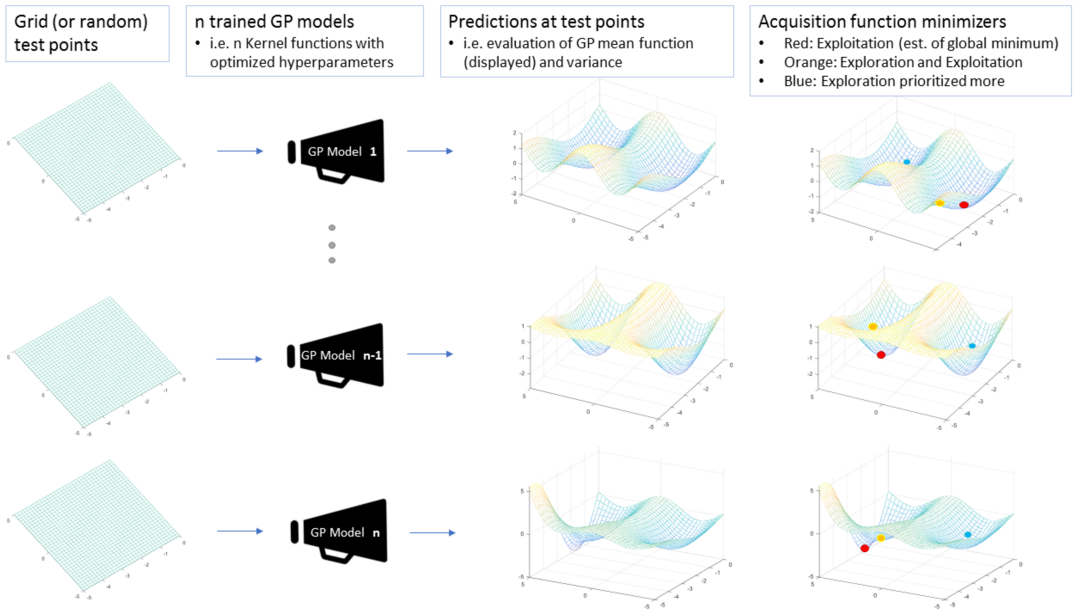
While our work provides a proof-of-concept demonstration of the multi-GP strategy, it offers many avenues for improvement. For example, our current goal is efficient global optimization with respect to the number of function evaluations, yet when considering the vast variation in resources required for a given evaluation (in terms of simulation time as well as memory), it may make sense to define efficiency with respect to total search time instead. To give some idea of the time required for stochastic assembly, evaluation of the ground truth point with 300 trajectories takes on the order of 30 min, while distant points in parameter space can span the range of hours to a week. One way to accomplish this may be to separately model the expected evaluation time, and take this into account during AF minimization. Another avenue for improvement concerns the empirical noise variance in RMSD at evaluated points; information to which we have access but do not directly utilize in GP regression. Modeling this variance itself as a GP may improve the ability of the LCB, which is constructed with the standard deviation at test points, to explore the space.

Furthermore, like all black box search methodologies, ours requires many design choices which balance competing factors including run time, cluster architecture, available memory, and the

details of simulating molecular assembly. We attempted to bias the search as little as possible, defining the search bounds as a hypersphere surrounding the known ground truth and selecting initial training points randomly within the region, and refraining from enforcing hyper-priors on the kernel hyperparameters. In acquisition function minimization our sampling methods were simple, again based around randomly selecting points from hyperspheres, and more sophisticated sampling strategies might lead to more efficient optimization.

Finally, it also important to note that this method is limited to learning models of a system under experimental conditions, typically in vitro, which may be quite far from conditions of the functional system in vivo. Many extrinsic factors might perturb system behavior in vivo, such as the presence of other molecules interacting with the system or generic effects, such as molecular crowding. Prior work has explored the question of how to "correct" a rule-based system learned in vitro for some effects one would expect in vivo (e.g., crowding [243]). Such approaches cannot account for all possible differences, though, and addressing that issue is a hard problem that would need to be solved on a system-specific basis.

A



B

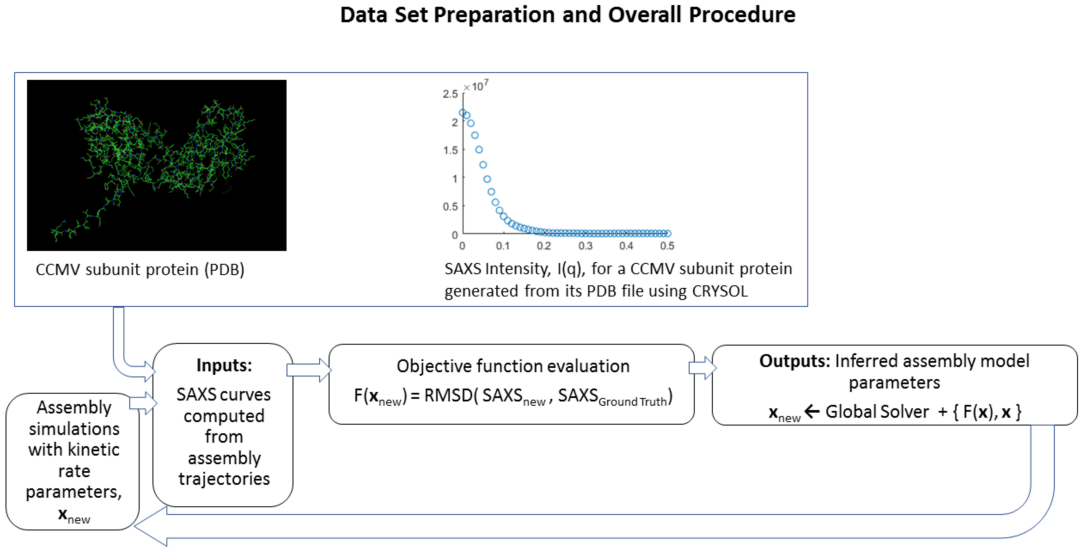


Figure 2-3: High-level overview of our multi-GP optimization strategy. (A) Visualization of a single round of our multi-GP model optimization. (B) Overall parameter inference pipeline incorporating the multi-GP optimization method of (A).

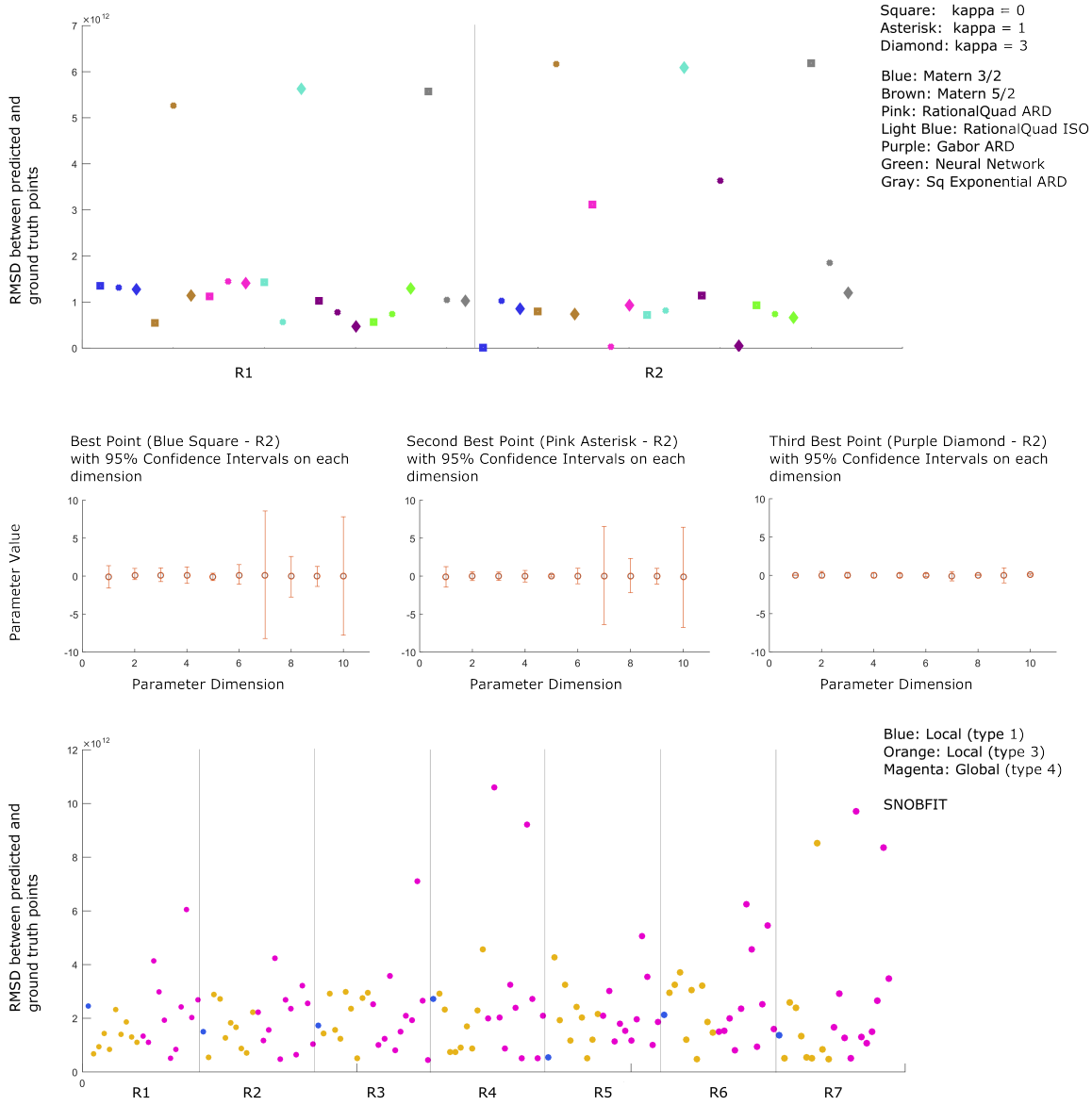


Figure 2-4: Comparison of objective values for our multi-GP optimization (top/middle) and SNOBFIT (bottom). Both methods use the same training set of 100 randomly sampled inputs, and both return 21 points for evaluation in a subsequent round. In the second round (R2 region) of search, our method recovers 3 low RMSD points, i.e., the blue square, pink asterisk and purple diamond. These three points, displayed in the middle figures with 95% confidence intervals for each dimension, minimize acquisition functions for distinct GPs and exploration/exploitation trade-off parameters. Displayed for SNOBFIT are 7 rounds of search in which it fails to recover equally low RMSD points. With default settings, SNOBFIT returned points of three types (distinguished by color), two of which result from local searches, and the remaining from a global search.

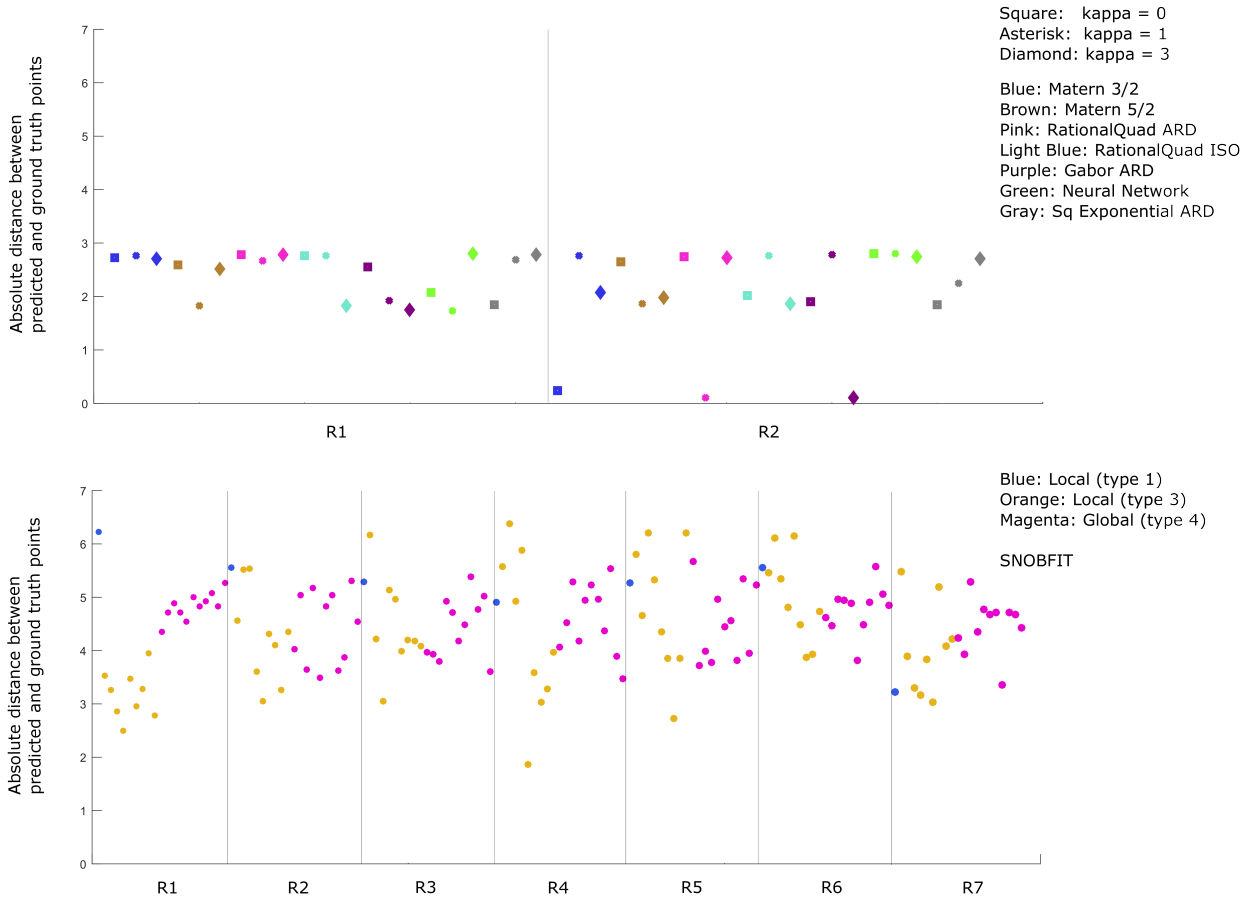


Figure 2-5: Comparison of error in predicted parameters for our multi-GP optimization (top) and SNOBFIT (bottom). Both methods use the same training set of 100 randomly sampled inputs, and both return 21 points for evaluation in a subsequent round. In the second round (R2 region) of search, our method recovers 3 points very close to the ground truth, i.e., the same blue square, pink asterisk and purple diamond. Displayed for SNOBFIT are 7 rounds of search in which it fails to recover equally close points.

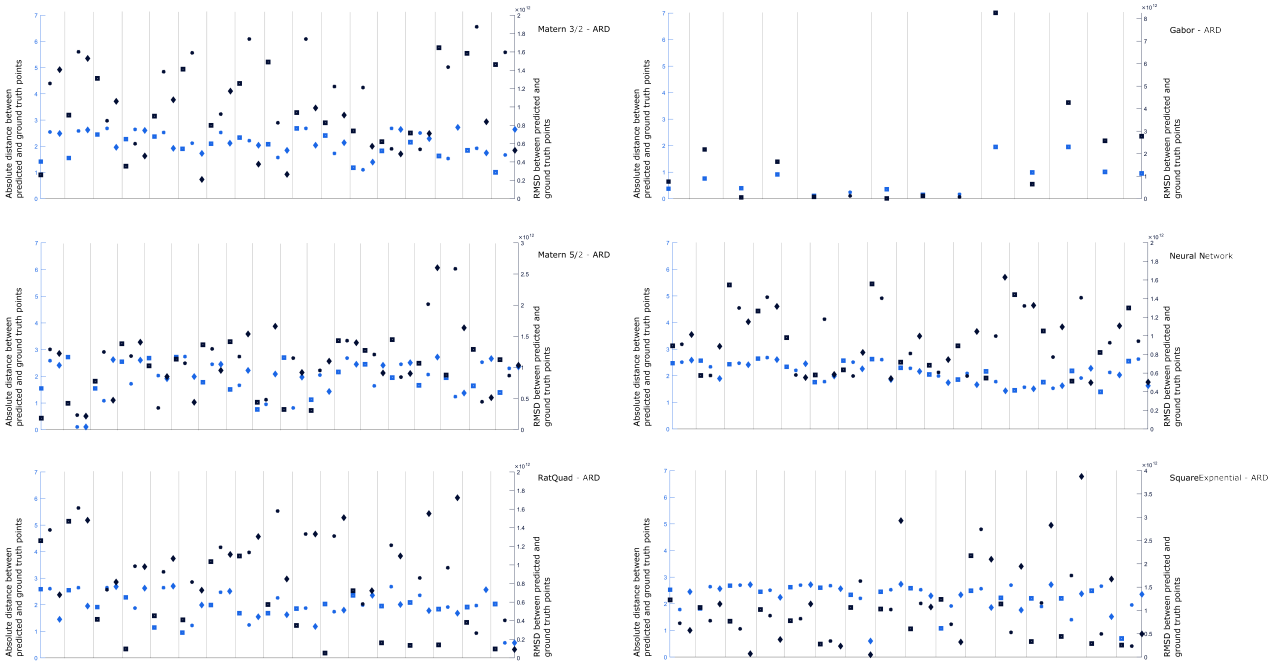


Figure 2-6: Results for single-GP parameter searches in which a single kernel function is used during all rounds. Displayed for each search are the errors in objective value (RMSD - right y-axis, dark blue) and error in predicted parameters (Distance - left y-axis, light blue) for each round.

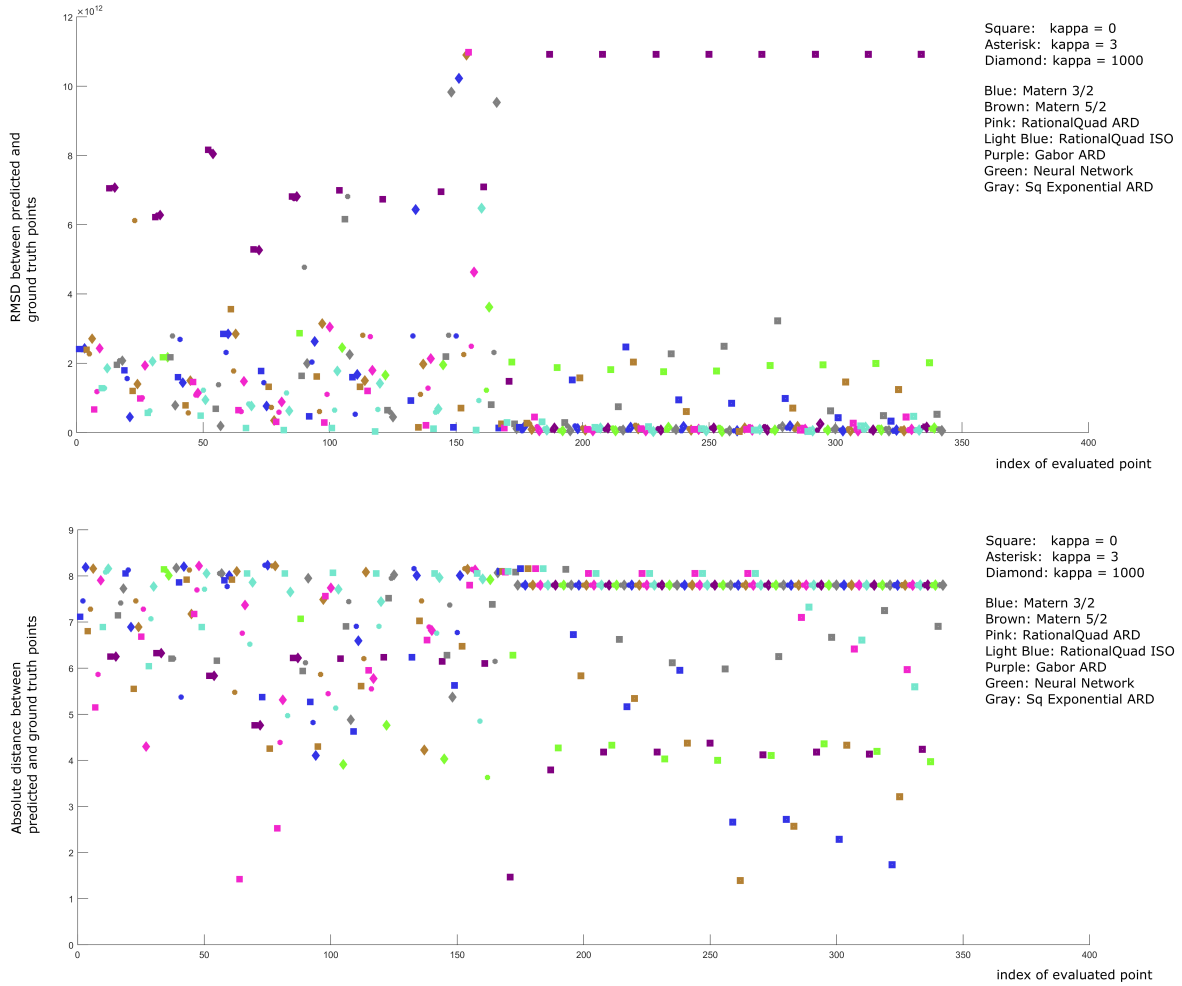


Figure 2-7: Results from 20 rounds of optimization for a large (hypersphere of radius 9 logs) search space. Individual rounds are not delineated due to variable numbers of points returned from acquisition function minimization in different rounds. Shown are (top) the RMSD values, and (bottom) the errors in predicted parameter sets as the search progresses. Note that the best points found (very low RMSD) often correspond to comparatively distant parameter sets. This is a result of the fact that the objective can be insensitive to large displacements in some of the input dimensions but not others.

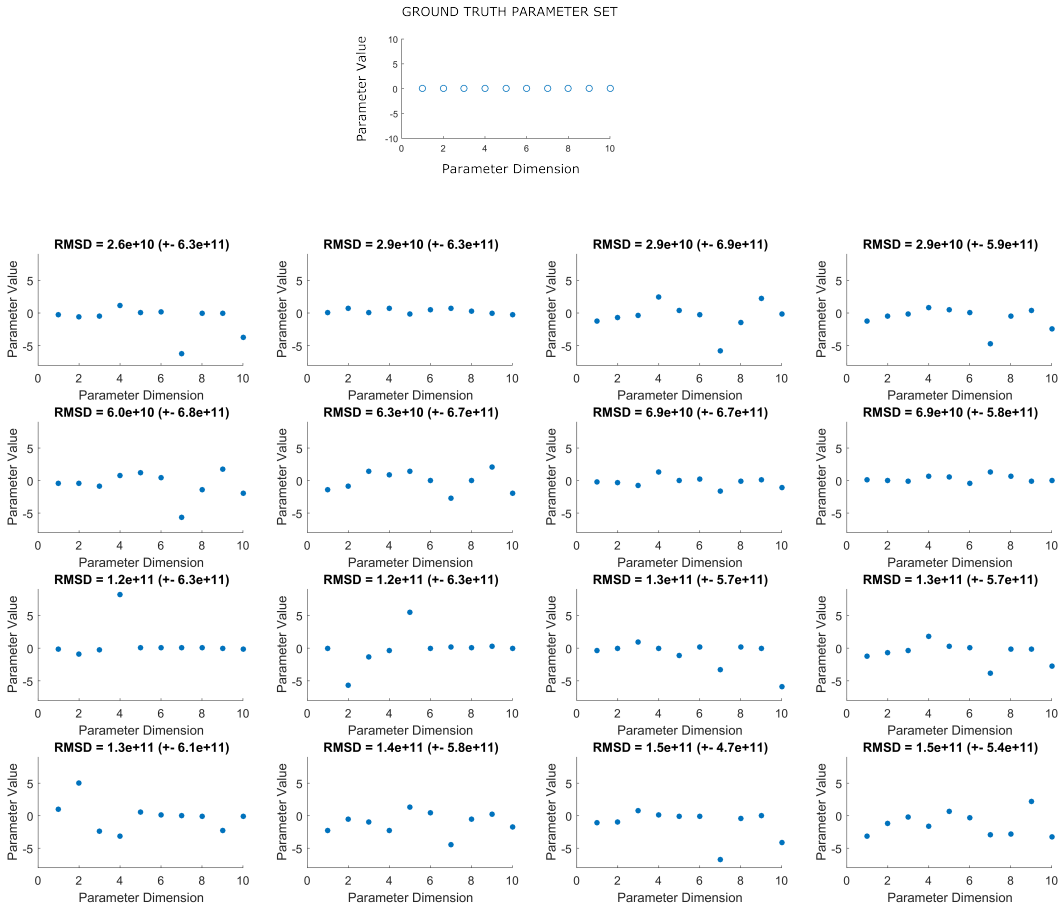


Figure 2-8: 16 points returned by re-evaluation selection subroutine. The points were chosen from among the 50 top scoring points over all rounds.

Figure 8 Selecting Points for Re-Evaluation

Sort all points by decreasing objective.
Initialize $\{\}_{remaining}$ as the list of the top n points.
Define n_{max} as the max number of points to return for re-evaluation.
Initialize $\{\}_{Re-Eval}$ the list containing the best scoring point.

while $\neg$ $empty(\ \{\}_{remaining}\ )$; do

    $x \leftarrow \{\}_{Re-Eval}[end]$

    % C is a binary matrix. Rows correspond to points.
    % Cols correspond to kernel derived lengthscales defining "closeness".
    % If kernel j considers point i close, then $C(i,j) = 0$, else $C(i,j) = 1$.
    % (Note: Isometric kernels have a single lengthscale hyperparameter,
    % ARD kernels have a lengthscale for each input dimension.
    % Thus, for ARD kernels, two points are close if they are close in
    % **all** dimensions.)
    $C \leftarrow EvalCloseness(x, \{\}_{remaining})$

    % If majority of kernels consider a point close, we consider it close.
    % Remove it from consideration.
    $\{\}_{remaining} \leftarrow \{\}_{remaining} - \{\}_{close}$

    % Append first (i.e. best scoring) point to re-evaluation list.
    $\{\}_{Re-Eval} \leftarrow \{\}_{Re-Eval} + \{\}_{remaining}[1]$

done
$n_{RE} = min(\ n_{max},\ |\{\}_{Re-Eval}|\ )$
$\{\}_{Re-Eval} \leftarrow \{\}_{Re-Eval}[1 : n_{RE}]$

Figure 2-9: Pseudocode for selecting previously evaluated low RMSD points for re-evaluation at a lower noise level.

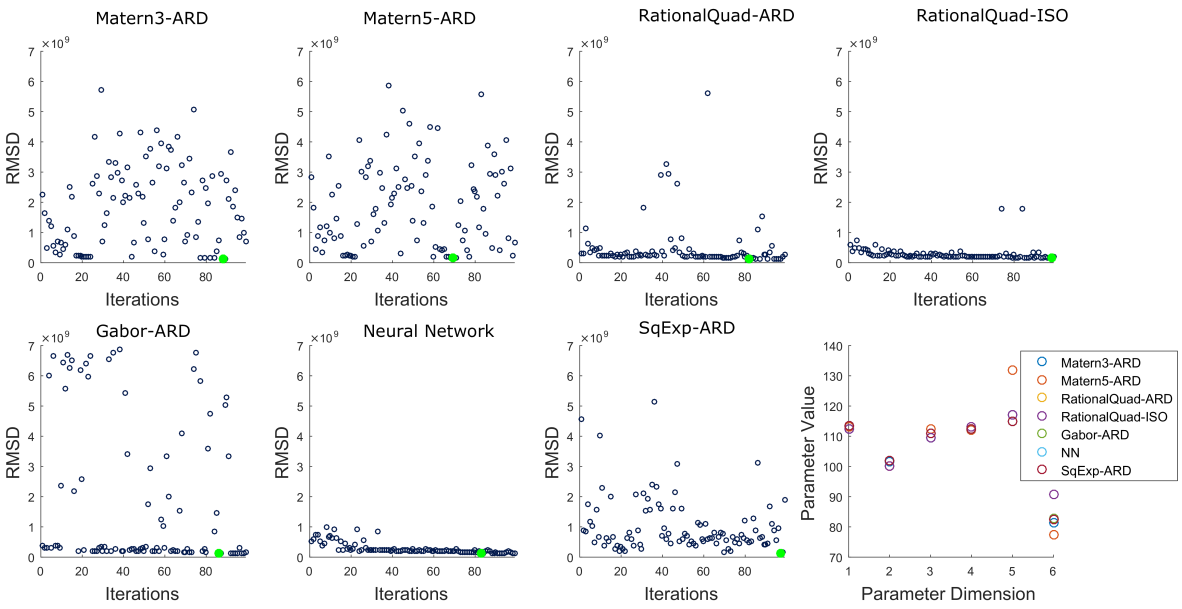


Figure 2-10: Model-fitting results for a 6-parameter ODE model. The results reflect 100 rounds of search with 893 points evaluated following initial training on 200 points randomly selected from a radius 100 hypersphere. Each round results in a new predicted minimizer for each GP model. The correspoinding RMSD is displayed. The lowest RMSD for each GP model over all rounds is plotted as a large green filled circle. In the final subfigure, the inputs with the lowest RMSD for each GP model are displayed.

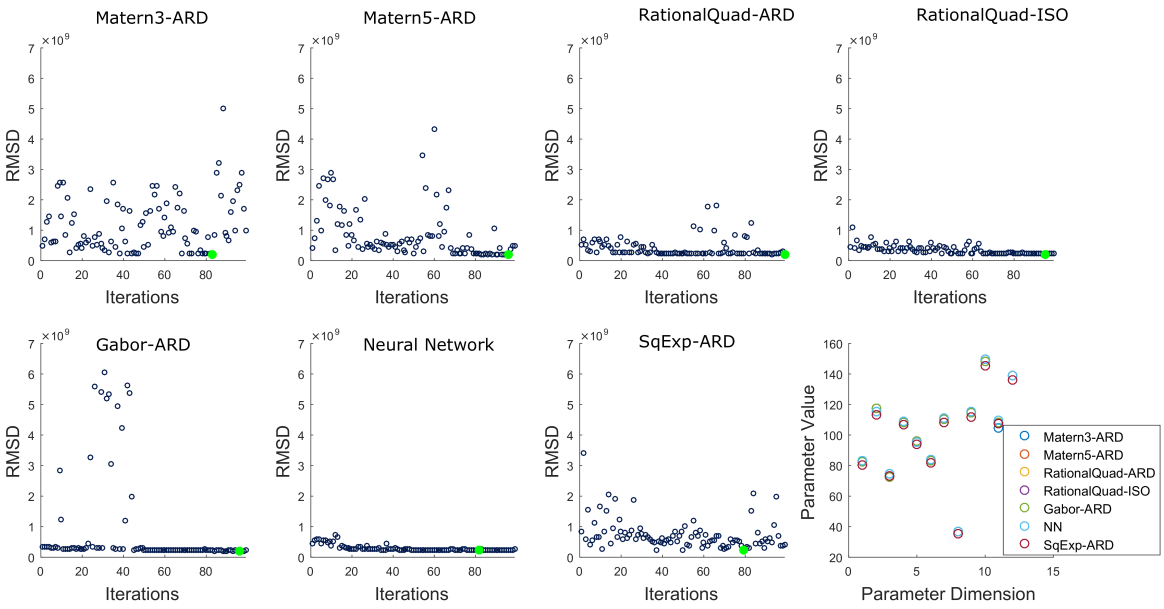


Figure 2-11: Model-fitting results for a full 12-parameter ODE model. The results reflect 100 rounds of search (893 points evaluated following initial training on 200 points randomly selected from a radius 100 hypersphere). Each round results in a new predicted minimizer for each GP model. The corresponding RMSD is displayed. The lowest RMSD for each GP model over all rounds is plotted as a large green filled circle. In the final subfigure, the inputs with the lowest RMSD for each GP model are displayed.

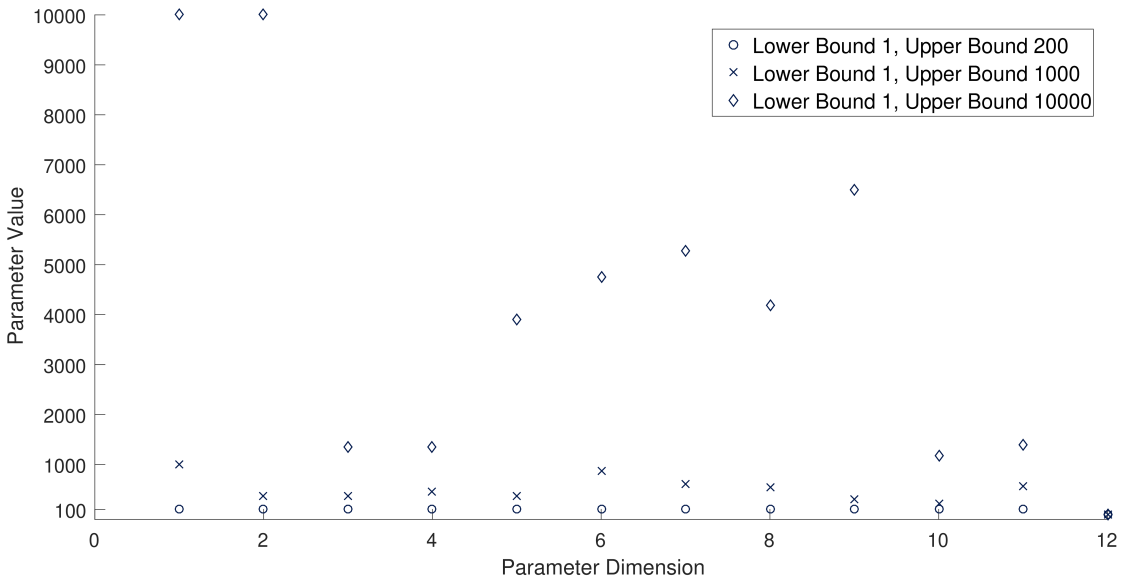


Figure 2-12: Results for ODE model-fitting using three separate MCS searches with varying search space sizes: 200 (1200 function calls, max allowed 7200), 1000 (878 function calls, max allowed 7200), and 10,000 (1439 function calls, max allowed 7200). Each dimension is lower bounded by 1 to enforce that all rate constants are positive. Only the smallest search space resulted in the correct minimizer being found.

Chapter 3

A Novel Algorithm for Particle-Level Spatial Stochastic Simulations

3.1 Introduction

Simulation methods have become a valuable adjunct to experimental work, facilitating the interpretation of experimental data and inferences about experimentally unobservable aspects of biomolecular dynamics [142], yet accurate simulations remain challenging for many biochemical processes crucial to living systems. The need for improvements in simulation technology is particularly acute for macromolecular assembly systems, which are central to nearly all cellular processes, yet frequently not directly observable experimentally due to their small scale and rapid dynamics [265]. Intractability of experimental approaches is particularly acute for understanding self-assembly *in vivo*, which may operate quite differently from purified *in vitro* models due to such effects as spatial confinement [48, 166, 285], macromolecular crowding [137, 243], and influences of extrinsic cellular factors [277]. The challenges of developing simulations that are both accurate and efficient, especially for hard-to-model systems like self-assembly, has led to extensive work on models and algorithms for biochemical simulation seeking to balance computational efficiency with fidelity to the complexity of the underlying biology. [1]

The Gillespie Stochastic Simulation Algorithm (SSA) [92, 95] was particularly influential in establishing a computational framework for efficient sampling of chemical reaction trajectories,

[1] This chapter is based on work currently under reivew: [267]

especially for small copy-number settings typical of biochemistry in the cell. The SSA has proven a valuable tool for understanding the kinetics of reaction networks, i.e., tracking the evolving populations of interacting reactant species, when older methods based on deterministic differential equation systems are too inaccurate or computationally infeasible [96, 290, 182]. Many improvements have been made to efficiency of the basic method either via approximations or for particular spaces of model parameter [93, 208, 37, 127, 175, 5, 246, 67, 164]. Yet the SSA is not explicitly spatial and instead treats the reactants as uniformly distributed at all times, aside from transient fluctuations. To better capture spatial heterogeneity, extensions of the SSA have been developed based on the reaction diffusion master equation (RDME), typically partitioning the reaction volume into compartments or voxels for which the usual well-mixed assumption applies in each compartment [13, 125]. In these spatial Gillespie models, reactants can react within a compartment or diffuse to an adjacent compartment. However, there is an inherent conflict between accuracy (smaller compartments imply higher spatial resolution) and the well-mixed assumption (better satisfied with larger compartments and/or diffusion rates). In fact, even in the limit of fast diffusion rates, RDME may not converge to the Chemical Master Equation (CME) underlying the Gillespie algorithm [245].

Brownian dynamics (BD) methods provided an opposite extreme of efficiency/realism tradeoffs for such modeling, allowing detailed, off-lattice spatial dynamics but at a cost of much greater computational cost. Coarse-grained BD methods have been widely used in self-assembly modeling, as they can deal well with systems with complicated spatial heterogeneity or geometrically intricate structures [234, 26, 106, 144, 43, 44, 10, 66]. However, their need to explicitly model diffusion trajectories of single particles creates high computational demands due to the large gap between timescales of diffusive motion versus those of typical molecular assembly processes. Smoldyn [6] is one such simulation package in which molecules diffuse with ideal Brownian motion and react upon collision. It has been developed extensively since its initial release in 2003, e.g., the inclusion of rule-based modeling, volume exclusion handling, on-surface diffusion, single particle tracking, and integration with BioNetGen. However, Smoldyn's diffusion accuracy is tied to the length of its fixed time steps.

Green's function reaction dynamics (GFRD [278, 279]) provided an alternative approach to capture spatial heterogeneity in simulating reaction-diffusion systems while taking advantage of

SSA-like efficient discrete event simulation without requiring spatial discretization. Instead of generating sample trajectories from the CME or RDME through MCMC, or numerically solving the many-body Smoluchowski equation as in Brownian Dynamics, GFRD analytically solves the Smoluchowski equation for single molecules and molecular pairs in terms of Green's functions. These Green's functions describe the probability of finding a molecule (pair) at a certain location and time given a known position(s) at an earlier time. A maximum time step is chosen such that, with high probability, at most two molecules come into contact, a requirement for analytical tractability. This single/pairwise interaction assumption becomes more valid with smaller time steps, introducing a trade-off between accuracy and efficiency. Reactions are incorporated through the boundary conditions, and the method combines into a single step propagation through space and reactions between particles. eGFRD [248] is a more recent exact algorithm which removes the accuracy/efficiency trade-off by including the concept of "protective domains" first developed by Oppelstrup et al. [193]. These domains are geometrically simple mathematical boundaries enclosing single molecules or pairs, each of which requires a distinct Green's function solution yielding next event types (domain escape or reaction) and waiting times. Because the time steps are now domain specific, eGFRD is an asynchronous algorithm allowing increased efficiency in some circumstances, although the additional mathematical complexity comes at significant computational expense.

The Small Voxel Tracking Algorithm (SVTA) developed by Gillespie et al. [97] is another particle based simulation algorithm for reaction-diffusion systems. While SVTA is based on the same underlying physics as eGFRD, its implementation is instead based on a discrete space. Instead of protective domains, SVTA constructs one and two particle "corrals", within which single molecules and molecule pairs hop between voxels and potentially interact. More specifically, it is the center of each molecule that hops since the voxel size is typically smaller than the molecular radius. These small voxel dimensions rule out the use of traditional bimolecular propensity functions that rely on the well-mixed assumption. Because the system state evolves on the time scale of diffusion hops, SVTA does not need to analytically sample locations on the protective domains, an easy task only when the domain is a sphere or other simple shape. It can simply keep track of when a diffusion hop places a molecule's center in a voxel identified with the corral. SVTA therefore bypasses the need for a suite of domain specific Green's functions in favor of

implementing individual diffusion steps on a lattice.

Similarly, the Microscopic Lattice Method (MLM) of Chew et al. [49] simulates lattice based diffusion with reactions. However, MLM aims at optimizing efficiency by simulating molecules of equal size, and requires that voxels dimensions are larger than molecular radii. Additionally, without corrals or protective domains, MLM relies on periodic boundaries to control the simulation volume and number of molecules. A direct comparison between Chew's MLM and Gillespie's SVTA is unavailable, however, the trade off seems to be that the former is potentially more efficient while the latter can simulate more complex biochemistry.

Despite these advances, the most challenging systems remain out of reach of molecular simulation methods without substantial simplifications [265]. New advances in models and algorithms for efficient but physically realistic simulation remain a pressing concern if the field is to continue to move towards solving the grand challenge of truly comprehensive and predictive models of whole-cell biochemistry.

We develop an alternative methodology intended to reduce the computational complexity of eGFRD while maintaining discrete event based system updates. Our goal is not to present a fully optimized algorithm, but rather to explore the use of time-dependent reaction propensities as a basis for reaction-diffusion simulation in continuous space.

3.2 Theoretical Framework

In this section, we present some theoretical concepts that will be useful subsequently in explaining our model and its relationship to prior work. Consider the bimolecular association reaction system:

$$A + B \Rightarrow C$$

governed by

$$\frac{d[A](t)}{dt} = \frac{d[B](t)}{dt} = -k(t)[A](t)[B](t) \tag{3.1}$$

where A and B are hard-sphere species with radii r_A and r_B and diffusion coefficients D_A and D_B. There are two traditional treatments of diffusion influenced reactions. The first was introduced by Smoluchowski [284] and later extended by Collins and Kimball (CK) [54]. At time t=0, a single

particle of species A is considered fixed at the origin and an initial surrounding concentration gradient is set up for the mobile species B molecules. They showed that

$$k(t) = \Phi(t)/c_0 = (4\pi R^2 D/c_0)(\partial c/\partial r)_{r=R} \tag{3.2}$$

where $\Phi(t)$ is the probability flux across a boundary sphere for the A particle at radius R, and c_0 is the initial uniform concentration for species B. The simultaneous diffusion of both species is incorporated by setting D as the sum of their respective diffusion coefficients. In this picture, the concentration gradient for the mobile B species $c(r,t)$, defined as the concentration of the B species at distance r from the origin at time t after the initial condition, is found by solving the diffusion equation

$$\partial c/\partial t = D\nabla^2 c \tag{3.3}$$

subject to initial condition $c(r,0) = c_0$ and the radiation boundary condition $D(\partial c/\partial r)_{r=R} = \kappa c(R,t)$ where κ is a specific reaction rate. The solution $c(r,t)$ is a complicated function and obeys the relation

$$k(t)/k_i = \frac{c(R,t)}{c_0} \tag{3.4}$$

where k_i is the limiting value $k(t \Rightarrow 0)$. Naqvi et al.[184] (sections III.-IV.) updates this by replacing the diffusion equation with a discrete random walk model from which is obtained in the limit of sufficiently long time and distance scales

$$k(t)/k_0 = \frac{c(R+\Delta,t)}{c_0} \tag{3.5}$$

with Δ equal to two thirds the scattering mean free path.

The second treatment is due primarily to Noyes [188] and considers an isolated pair of reactive molecules separating from a nonreactive encounter. They showed that

$$k(t) = k_0\Big[1 - \int_0^t h(t')dt'\Big] \tag{3.6}$$

where k_0 is defined as "the rate constant applicable for an equilibrium molecular distribution"[189] and $h(t)dt$ is the "probability two molecules separating from a nonreactive encounter at time zero

will react with each other between t and $t + dt$" [188]. This can be recast into the form ([184] Eq.47)

$$k(t)/k_0 = S(t; r_0 = R_0, R) \tag{3.7}$$

where R_0 denotes the distance between two molecules separating from a nonreactive encounter at time zero, and the survival probability $S(t; r_0, R)$ is defined as

$$S(t; r_0, R) = 1 - \int_0^t p(t'; r_0, R) dt'. \tag{3.8}$$

These two major approaches, based on the diffusion equation and particle-pair standpoint respectively, can be shown to be equivalent under certain assumptions and by a lengthy derivation (see [184], sections IV and V). Our method is most easily identified with the theoretical framework of Noyes, but with a different emphasis on instantiating the physical model so as to enable efficient stochastic off-lattice particle simulations. We describe the novel features of our model in more detail below.

The function $h(t)$ appearing in Noyes' fundamental relation can be inferred as the special case

$$h(t) = p(t; r_0 = R_0, R) \tag{3.9}$$

To be clear, r_0 is the separation distance immediately after a nonreactive encounter. Naqvi argues that $r_0 \neq R$, the reactive contact distance defined in the boundary condition, but instead $r_0 = R_0 = R + \Delta$. The exact expression for $p(t; r_0, R)$ depends on various assumptions, e.g., that the discrete random walks taken by the particles are accurately described by a continuous diffusion equation. In this case, one needs to make further assumptions about initial conditions and boundary conditions.

In the CK picture, the reaction rate evolves only during the time window beginning with the initial condition and ending with a reaction. The assumption here is that immediately after a reaction, the system returns the concentration surrounding the product molecule to the fixed initial value. As such, the formalism may not be suitable to an event-driven, explicitly spatial simulation. Chew et al. [49] with their microscopic lattice method address this issue by deriving their lattice parameters as analogues to the effective or steady state reaction rates in

the continuum CK/Noyes theory. This ensures the model behaves similarly to the theory over suitably long time scales.

While our treatment of diffusion influenced reactions is similar to the particle pair approach in Noyes theory, there are notable differences. Instead of using probabilistic arguments to derive reaction rate functions suitable for a differential equation model, we use them to derive reaction propensities suitable for a discrete event SSA. Our conception is as follows: Given a collection of molecules in an explicit and bounded 3d space, and assuming a maximum diffusion time before which we observe their positions, reaction waiting times can be randomly sampled using pairwise propensity functions. The probability density we focus on is not $h(t) = p(t; r_0 = R_0, R)$, but rather $p(t; r, R)$ where r is interpreted as the initial separation, and R is the separation below which a reaction can occur.

In the remainder of the chapter we describe the model and implementation, which we refer to as the Diffusion-Based Embedding of the Stochastic Simulation Algorithm in Continuous Space (DESSA-CS) method, in reference to an earlier space-free method [307] based on an accelerated SSA algorithm [127], and demonstrate its effectiveness in comparison to prior alternatives through application to a Michaelis-Menten model.

3.2.1 Background on Green's Functions

Whereas thermodynamics is concerned with the evolving temperature in a region, reaction-diffusion chemistry is concerned with the evolving position probability for a diffusing molecule. Additionally, the evolving probability of a reaction between two molecules may be considered. The equations for temperature and probability are formally interchangeable, but the terms will have different interpretations. Any system requires three ingredients to obtain its solution: an equation governing the time evolution (a diffusion/heat equation), an initial condition and one or more boundary conditions.

The 1d problem is based on an instantaneous plane source, the 2d problem on an instantaneous line source and the 3d problem on an instantaneous point source.

Our starting point in 3d is a solid material bounded by some surface. The Green's function (GF) is understood as *the temperature at (x,y,z) at time t due to an instantaneous point source of strength unity generated at the point P(x',y',z') at time τ, the solid being initially at zero*

temperature, and the surface being kept at zero temperature (p353 [41]). Translating this into diffusion language, we have a molecule (source) at point P(x',y',z') at time τ, the probability (temperature) of finding it anywhere else in the volume (solid) is zero initially. As a consequence of diffusion, this probability - whose integral (strength) at every instant is unity - will spread throughout the volume. The molecule is never found at (or beyond) the boundary surface since the probability is kept at zero there.

In the literature on heat transfer, an additional heat source may be considered at the boundary surface - one which depends on time and spatial position. In fact there are 4 typically encountered types of boundary condition (See p18 [41]): (1) Prescribed surface temperature. (2) No flux across the surface. I.e., $\frac{\partial T}{\partial n} = 0$, at all points of the surface where the differentiation is w.r.t. the outward normal to the surface. (3) Prescribed flux across the surface. (4) Linear heat transfer at the surface. This is also called 'radiation' boundary condition. If the flux across the surface is proportional to the temperature difference between the surface and the surrounding medium (i.e., temperature flux $= H(T_{surface} - T_{surroundings})$), then the boundary condition is

$$K\frac{\partial T}{\partial n} + H(T_{surface} - T_{surroundings}) = 0.$$

H is the surface conductivity. K is the thermal conductivity of the solid. $\frac{\partial}{\partial n}$ denotes differentiation in the direction of the outward normal to the surface.

"Radiation Boundary Condition": Describing the Boundaries of a Simulation Volume

When the heat source is within the solid, $H/K > 0$ is assumed as this corresponds to loss of heat across the surface into the surrounding medium. $H/K < 0$ would correspond to a *supply* of heat at the surface at a rate proportional to the temperature difference between the surface and the surroundings. See footnote in [41] p19. However, reflecting simulation box walls for diffusing molecules can be described with $H/K < 0$.

"Radiation Boundary Condition": Describing Reactions

However, the boundary conditions do not have to correspond to physical boundaries, i.e., cell membrane, reaction container, etc. They can instead be used to examine the evolution of the probability within an imagined region (i.e., protective domains of eGFRD [248]), or to examine the probability governing radial separation of two diffusing molecules where the boundary condition is associated with the minimum separation due volume exclusion (i.e., MLM [49]), or to examine the flow of probability into a reaction.

The GF used in the microscopic lattice method of Chew et al. [49] comes from Section 14.7 IV. of Carslaw & Jaeger [41] and also Eq. 3.10 of Naqvi et al [183]. It is the solution for the following problem. *The region bounded internally by the sphere* $r = a$. *Unit instantaneous spherical surface source at* $r = r'$ *at* $t = 0$. *Boundary condition at* $r = a$: $k\frac{\partial T}{\partial r} - hT = 0$ *where* $k \geq 0$, $h \geq 0$ (p368 [41]). In this case, because the region is bounded internally by a spherical surface, the heat source is outside that sphere which accounts for the minus sign. [We understand this as $T_{surroundings} = T_{r<a} = 0$, therefore $h(T_{surface} - T_{surroundings}) \rightarrow hT$] It should be noted that this is a 1d problem - that of a radial flow of heat. Such a 1 dimensional framing is appropriate because the question Chew et al were interested in was the radial separation between two diffusing molecules. Our new algorithm, DESSA-CS, also uses 1d GFs based on radial separation. This framing has been used historically by Naqvi, Noyes, and Collins & Kimball. Framings requiring GF solutions in 2 and 3 dimensions also exist, e.g., eGFRD.

Reaction-diffusion algorithms that use Green's functions tend not to focus on h explicitly, but it's helpful to consider it to better understand what's going on. From a comparison of Appendix A of [49] (variables k_a, R, D) with the original variables used by Carslaw & Jaeger [41] (variables h, k), we arrive at $h = k_a$ and $k = 4\pi R^2 D$ for the microscopic lattice method. A positive h which depends on the reaction parameter k_a makes sense. At the boundary surface representing contact between two molecules, we are *loosing* heat (probability) from the system. That lost probability passes into the sphere at $r = a$, and can be interpreted as going into the reaction channel.

Beijeren and colleagues have investigated h from the perspective of Noyes' theory of diffusion based reaction kinetics and derived it in terms of fundamental quantities. See Equation 21 from

[276] where they use the variable κ in place of h.

$$h = \kappa = \frac{P_{react}\nu_{AB}}{g(\sigma)\rho_B}$$

P_{react} is defined as the probability of a reaction in a collision between A and B molecules, ν_{AB} is the collision frequency of A and B molecules, $g(\sigma) = g(r = \sigma)$ is particle pair correlation function describing the relative motion of two molecules diffusing in a liquid medium. The radial separation σ is that of the centers of mass at the collision distance. The last variable, ρ_B is the local density of B molecules surrounding an A molecule.
In the following sections where greens functions are used, the notational switch $h \rightarrow c$ is made to maintain consistency with our published simulation paper.

3.3 Methods

Algorithm 1 summarizes our general procedure for off-lattice spatial stochastic simulation. It makes use of a discrete event structure similar to the stochastic simulation algorithm, with the addition of routines for sampling reaction locations. This sampling is based on diffusion spheres containing n_{sigma} standard deviations of the Gaussian distributions describing each particle, similar to GFRD. The resulting positions (due to reactions and position-only updates) are therefore restricted to be within the diffusion spheres, no matter the choice of n_{sigma} (typically 3-5).

In contrast with existing simulation methods in which the boundaries of the simulation volume are either periodic or reflective, we utilize an alternate approach. The state of each molecule is given by the mean and variance of its Gaussian probability distribution, therefore we do not have access to precise positions or velocities. As such, the action of a periodic boundary condition is not well defined. Our approach to wait time sampling is to allow the diffusion spheres of molecules near the boundary to extend a small distance beyond the boundary, typically a small fraction of the container length. When sampling reaction locations, we implement a reflective boundary procedure designed to keep the molecules within the simulation volume while respecting the physics of diffusion.

Algorithm 1 DESSA-CS procedure

1: Initialize Event Queue: For each assembly, consider self events (unimolecular reaction, position-only update) and pair events (bimolecular reaction) and add to the queue the earliest self event and pair event for each assembly.
2: Main Loop:
3: **repeat**
4: Extract the next event on the queue.
5: **if** event is bimolecular and valid **then**
6: sample location for product given waiting time; update data structures; add next self event(s) to the queue; add next potential bimolecular events to the queue.
7: **else if** event is unimolecular event and valid **then**
8: sample locations for both products; update data structures; add next self event(s) to the queue; consider each product and add next potential bimolecular events to the queue.
9: **else if** event is position-only update and valid **then**
10: sample location; update data structures; add next position-only update to the queue; add next potential bimolecular events to the queue.
11: (Apply boundary conditions to product(s) if necessary, before adding new events to the queue.)
12: **until** max allowed simulation time or max allowed number of reactions is reached

3.4 Sampling Bimolecular Reaction Waiting Times

Consider a set of K possible bimolecular reactions, i.e., distinct pairs of individual molecules each represented as a point particle, and assume each molecule traverses an explicit 3d space by diffusion. For each molecule pair, k, there exists a reaction propensity $a_k(t; s)dt$ describing the probability of an encounter and subsequent reaction of that pair, within some small time interval $[t, t + dt)$ after the most recently executed event at time s. The waiting time, t_{wait}, before the next reaction of reactant pair k can be sampled via the equation [4]

$$\int_0^{t_{wait}} a_k(t \mid s)dt = ln(1/r_k) \tag{3.10}$$

which determines the time at which the integrated propensity equals an exponentially distributed random variable. r_k is the random number sampled for molecular pair k uniformly from the unit interval for use in sampling an exponential waiting time by the transformation method. Because each of our propensity functions are unique to their associated molecular pair, the reaction channels defined in the original SSA and in Anderson's modified next reaction method [4] at the the species level are now defined at the molecule pair level.

3.4.1 Point Particles

At the moment a given molecule's state is updated, the probability density describing its center of mass is concentrated at a single point, i.e., a Dirac delta function centered on that point. As time progresses, the probability density spreads as a Gaussian. This is the free diffusion Green's function solution of the diffusion equation [278]. The positions of two molecules A and B are therefore described by two independent Gaussian random variables, $x_A(t) \sim\ N[\mu_A, \Sigma_A(t)]$ and $x_B(t) \sim\ N[\mu_B, \Sigma_B(t)]$. In order to evaluate Pr(*encounter* & *reaction* | t), the joint probability of an encounter and a reaction during the interval $[t, t + dt)$, we factor the joint probability as Pr(*encounter* | t) * Pr(*reaction* | *encounter*). The latter factor is expressed using a time-independent *intrinsic* reaction rate constant, c, such that $c\,dt$ is the constant encounter conditioned reaction probability over a small time interval. Note that c is specific to this formalism and not equivalent to the microscopic reaction rates used in Smoluchowski or Collins-Kimball theory.

In evaluating the former factor, Pr(*encounter* | t), we assume the initial positions of A and B are known and ask the following question: given a sampled position $\mathbf{x}_A$ of molecule A taken after time t, what is the probability a sampled position $\mathbf{x}_B$ of molecule B after time t will be close to A? Here "close" means at a distance less than a threshold denoting contact or an encounter.

This question can be answered in the language of distributions of quadratic forms in random variables. We define the quadratic form $Q(t)$ as the squared Euclidean distance between Gaussian random variates $\mathbf{x}_A$ and $\mathbf{x}_B$.

$$X_{B-A}(t) \sim\ N\big(\mu_B - \mu_A, [\Sigma_A(t) + \Sigma_B(t)]\big)$$

$$Q(t) = X_{B-A}(t)^T\, X_{B-A}(t) \tag{3.11}$$

Thus,

$$Pr(encounter \mid t) = CDF_{Q(t)}(R^2_{enc}) \tag{3.12}$$

$$= Pr\big(Q(t) < R^2_{enc}\big) \tag{3.13}$$

where R^2_{enc} is the square of the encounter threshold distance. Theorem 4.2b.1 of Mathai & Provost[169] provides a formula in terms of an infinite power series expansion which we use for evaluation.

$$CDF_{Q(t)}(R^2_{enc}) = \sum_{h=0}^{\infty}(-1)^h z_h(t)\frac{(R^2_{enc})^{(3/2)+h}}{\Gamma\big((3/2)+h+1\big)} \tag{3.14}$$

The coefficients $z_h(t)$ are defined recursively and depend on $\mu_{AB} = \mu_B - \mu_A$, and $\Sigma_{AB}(t) = (\Sigma_A + \Sigma_B)$. Convergence is defined by no change to 5 places after the decimal for 20 successively higher order approximations. For very small t and large initial separation, the approximation can oscillate wildly about zero. In these parameter regions where numerical instability is detected, we set the CDF to zero.

With isotropic diffusion, the reaction propensity given R^2_{enc} and $d = norm(\mu_{AB})$ after time t, and with intrinsic rate c, can be reparameterized as a function of the variance v of $X_{B-A}(t)$ rather than of time directly. This variance is simply the diagonal element of $\Sigma_{AB}(t)$. The reparameterized reaction propensity, denoted $a_k(t)$, is then given by:

$$a_k(t)dt = a_k(v \mid d_k, R^2_{enc,k}, c)dv \tag{3.15}$$

The time to next reaction can now be determined by evaluating

$$argmin_v \quad \int a_k(v)dv \geq ln(1/r_k) \tag{3.16}$$

and inferring t_{wait} from the variance value, should it exist. Figure 3-3 visualizes the wait time sampling procedure. One added complication is that the DESSA-CS algorithm is event driven. After each event, potential new reactions are considered for the product(s) of that most recently executed event. This implies that the position of the product (e.g., reactant A) is known precisely,

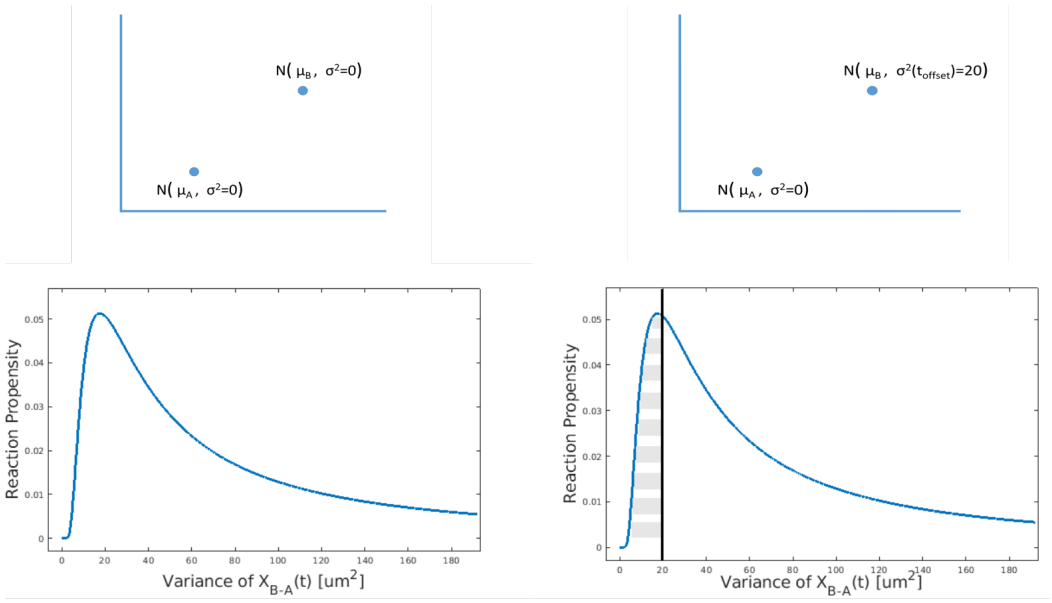


Figure 3-1: The figures on the left depict two molecules, A and B, described as Gaussians with means separated by $d = 7.235\mu m$. The point-particle reaction propensity grows as the variance increases, reaching a peak just before $20\mu m^2$ and then decreases monotonically. The figures on the right are more typically encountered in the algorithm. The most recent reaction for A was just executed and wait times are being sampled for the $A + B$ reaction. B has already been diffusing for a time t_{offset}, thus, propensity function integration begins not at zero variance, but instead at variance equal to $20\mu m^2$.

while its potential partner (e.g., reactant B) has been diffusing for a time t_{offset} and thus has its position represented by a Gaussian random variable. Any integrated propensity up through $v(t_{offset})$ must therefore be discounted when sampling the variance at which a reaction occurs. See Figure 3-1 for an illustration. The sampling procedure is described in Algorithm 2. For finite sized molecules, the procedure is similar, except the integrated propensities are expressed directly in terms of times rather than variances. A Matlab implementation of the algorithm is available on GitHub [264].

Our propensity function describing $\Pr(encounter\ \&\ reaction \mid t)$ is equivalent to $p(t; r, R)$ from the Noyes theory under the assumption that the molecules are dimensionless point particles for which there is no minimum separation distance. In this case, there is no need to go beyond the free diffusion Green's function solution to the diffusion equation as there are no boundary conditions enforcing a minimum pairwise separation.

3.4.2 Particles with Finite Size

Assume both particles are spherical and R defines the center-to-center distance at contact. In this context, the $p(t; r, R)$ described in the Theoretical Framwork section above is expressed as

$$p(t; r, R) = p(r = R, t; r_0, 0) * c\, dt \tag{3.17}$$

where c now denotes the absorbing/radiation boundary condition parameter, and $p(r, t; r_0, 0)$ is the Green's function solution to the following boundary value problem. Assume $p(r, t; r_0, 0)$ obeys a diffusion equation, and the initial separation between molecules is r_0. This is expressed with the initial condition

$$p(r, 0) = \frac{\delta(r - r_0)}{4\pi r^2}. \tag{3.18}$$

The two boundary conditions on $p(r, t; r_0, 0)$ ensure that the molecular separation never reaches infinity, and that at contact, the probability of a reaction is accounted for.

$$\lim_{r\to\infty} p(r, t) = 0 \tag{3.19}$$

$$4\pi R^2 D \left.\frac{\partial p(r, t; r_0, 0)}{\partial r}\right|_{r=R} = c\, p(R, t; r_0, 0) \tag{3.20}$$

From Chew et al. [49], and Jaeger & Carslaw [41] p. 368), the Green's function solution is

$$\begin{aligned} p(r, t; r_0, 0) = \frac{1}{8\pi r r_0}\frac{1}{\sqrt{\pi D t}}\Big(&exp[-(r - r_0)^2/4Dt] \\ &+ exp[-(r + r_0 - 2R)^2/4Dt] \\ &- 2B\sqrt{\pi D t}\, exp[B^2 Dt + B(r + r_0 - 2R)] \\ &* erfc(\frac{(r_0 - R)}{2\sqrt{Dt}} + B\sqrt{Dt})\Big) \end{aligned} \tag{3.21}$$

where $B = (1 + \frac{c}{4\pi R D})/R$. Note that the Green's function also depends on c through B. The propensity function is

$$a(t)dt = p(R, t; r_0, 0)\, c\, dt \tag{3.22}$$

and the time to next reaction, t_{wait}, can be determined from the integrated propensity by evaluating

$$\frac{B}{4\pi R^2 r_0}\left[erfc\big[\frac{B(r_0-R)}{2\sqrt{\tau}}\big]- \Big(exp(Br_0-BR+\tau)\,erfc\big[\frac{Br_0-BR+2\tau}{2\sqrt{\tau}}\big]\Big)-1\right]_0^{\tau-max} - ln(1/r_k)=0 \quad (3.23)$$

with $r_k \sim uniform[0,1]$ and $\tau = tDB^2$. The waiting time is inferred as $t_{wait} = \tau/DB^2$. As in the point particle context, when molecule B of the molecular pair has been diffusing for a time t_{offset} when we are sampling reactions for molecule A, the integrated propensity up through t_{offset} must first be subtracted from the L.H.S. of Eq. 3.23. Alternately, Eq. 3.21 can be numerically integrated.

Validation of the Wait Time Sampling Procedure

Noyes' theory is formulated in terms of the probability two molecules will re-collide (and potentially rebind) following a nonreactive encounter, therefore comparing the theoretical and simulated rebinding time probability densities is a useful test of our Gillespie inspired wait time sampling procedure based. Following Chew et al. ([49] Figure 2), we consider the activation limited and diffusion influenced cases. These are distinguished by $c/\nu_{AB} < 1$ and $c/\nu_{AB} \geq 1$, respectively. The parameter c is the boundary value parameter appearing in the finite particle propensity function and is related to the collision frequency ν_{AB} between A and B molecules in a hypothetical nonreactive system as:

$$c = \frac{P(reaction \mid encounter)\nu_{AB}}{g(r=R)\rho_B},$$

where $g(r)$ is the particle pair correlation function in a liquid phase and ρ_B is the relative density of B molecules. See Section III. and Eq.21 of [276]. Figure 3-2 shows, for three values of c/ν_{AB}, the theoretical density and the results of our simulations. We computed the integrated reaction propensity at 50,000,000 time points linearly spaced in the range [1e-8,1]. on the order of

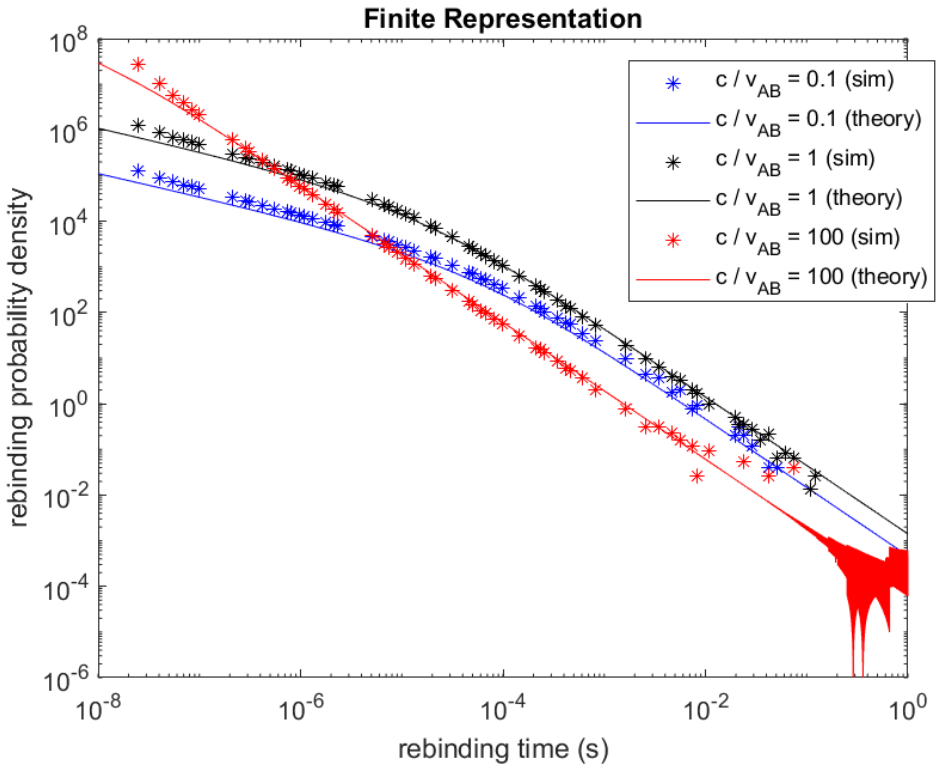


Figure 3-2: Rebind time probability density from Noyes' theory. We compare the theoretical curves in the finite particle representation with values computed from simulations at $c/\nu_{AB} = 0.1$, 1, and 100. Deviations from the theory at larger rebind times are due to the fact that more samples are required to characterize the probability densities than were drawn in our analysis. Simulation parameters: $D_A = 1\mu m^2 s^{-1}$, $D_A = 0\mu m^2 s^{-1}$, $r_0 = 0.01001\mu m$.

$\sim 100,000,000$ wait time samples were drawn for each of the three ratios and then aggregated into bins of width $w_{bin} = 2e-7s$. Simulated probability densities for a subset of bins were computed as

$$pdf(bin) = \frac{N_{bin}}{N_{total}\ w_{bin}},$$

where N_{bin} is the number of samples in the bin and N_{total} is the total number of samples. As the total number of samples grows, the simulated values approach the theoretical density.

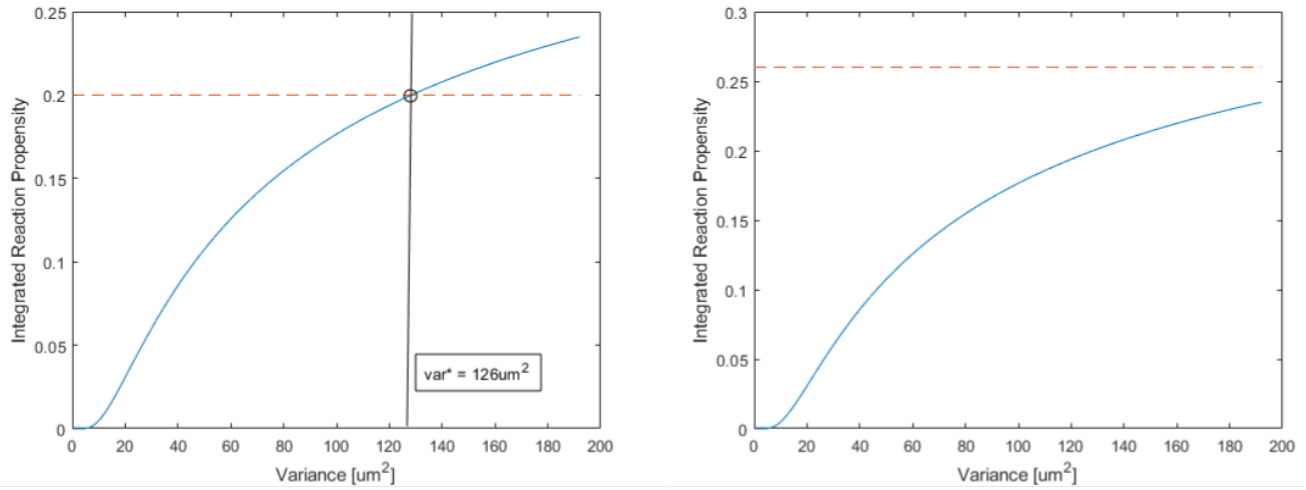


Figure 3-3: Examples of successful and unsuccessful sampling of a biomolecular reaction waiting time. In both subfigures, the solid curve is the integrated reaction propensity associated with two reactants described by Gaussians with means separated by $8\mu m$. The left subfigure shows the successful sampling of a bimolecular reaction waiting time as there is a variance value (and thus, a waiting time) at which the integrated reaction propensity equals the exponentially distributed random number, 0.2. In the right subfigure, the exponentially distributed random number is 0.26, and so there is not sufficient integrated propensity for a reaction to occur. The propensity curve corresponds to an intrinsic rate constant $6 * 10^7 s^{-1}$, encounter radius squared $R_{enc}^2 = 0.01^2 \mu m$, and diffusion coefficients $D_a = D_b = 1$.

Algorithm 2 Sampling Bimolecular Wait Times - Point Particle Representation

1: (Pre-simulation) Define vector of variance values, $\mathbf{v} = [0, V_{max}]$

2: (Pre-simulation) Define the curve $IntF(\mathbf{v} \mid d, R_{enc}^2, c)$ as the cumulative sum of reaction propensity values along the points $\mathbf{v}$. $\{IntF(\mathbf{v} \mid d, R_{enc}^2, c)\}$ is then the set of integrated propensity curves at increasing d, computed once, before the simulation begins. If desired, further sets of curves can be precomputed for alternate values of R_{enc} and intrinsic rate c.

3: (At run time) For reactant pair k = (A,B), select the appropriate curve, $IntF(\mathbf{v} \mid d_k, R_{enc}^2, c)$

4: Evaluate $IntF(v_{t_{offset}})$, the integrated propensity to be discounted, at the variance value corresponding to t_{offset}, i.e., $6D_b t_{offset}$.

5: Set $v^* \leftarrow argmin_v\ IntF(\mathbf{v}) \geq ln(1/r_k) + IntF(v_{t_{offset}})$

6: If v^* exists, t_{wait} is the solution to $v^* = 6D_a t_{wait} + 6D_b(t_{wait} + t_{offset})$

7: Else, no reaction is sampled. Update particle positions.

3.5 Sampling Bimolecular Reaction Locations

Again we make use of the labels A and B for the specific molecules undergoing the next association reaction. At this time, the spatial region available for the reaction consists of the intersection of the diffusion spheres bounding their independent Gaussian probability distributions. In order to correctly sample from this region, henceforth called the *overlap volume* (OV), we first introduce the concept of equiprobable rings.

3.5.1 Equiprobability Rings

The line AB connecting the initial known positions of A and B defines an axis of symmetry in the sense that within the OV there exist rings centered on this axis, whose points are equidistant from A and equidistant from B. The rings are therefore sets of equiprobability points from which molecule positions might be sampled. Each ring is uniquely defined by two numbers: the magnitude, r_A, of any vector from the initial position of A to a point on the ring, and the CCW angle, θ_A, between the vector and the line AB. After sampling (r_A, θ_A), we choose the reaction location uniformly at random from on the ring.

The joint probability density describing (r_A, θ_A) can be factored as $p(r_A|t)$ and the conditional probability $p(\theta_A|r_A, t)$, which suggests a sequential sampling procedure. First determine r_A and then use it to determine θ_A.

3.5.2 Diffusion Sphere Overlap Volume

While the OV grows continuously due to diffusion, for the purpose of location sampling at a given time we have found it useful to classify it into one of 5 distinct cases. These cases are not inherently meaningful, they are simply convenient ways of describing the evolution of the OV as well as the integration regions involved in sampling r_A and θ_A. For example, if the OV is identical to the diffusion sphere of A (as in cases 3 and 5), θ_A may take on any value in $[0, 2\pi]$, however if the OV has an irregular shape, certain angles may be prohibited. Figure 3-4 visualizes the two trajectories possible for the OV. The first trajectory applies when $D_B > 4D_A$ and passes through cases 1, 2, 3 and 5. The second trajectory applies when $D_A < D_B < 4D_A$ and passes through cases 1, 2, 4 and 5. Given the current system time t, the waiting time until the next reaction of

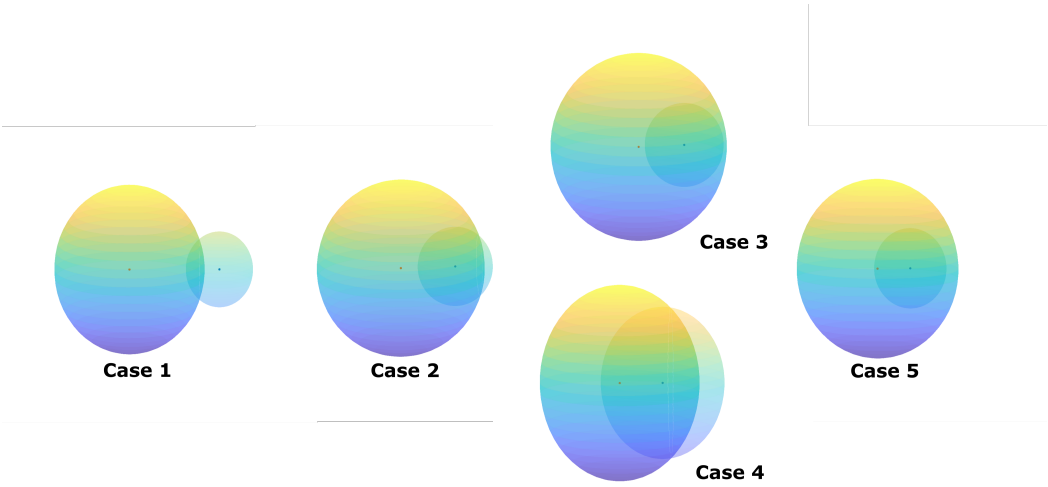


Figure 3-4: Cases of potential overlap of diffusion spheres in the process of sampling waiting time to a biomolecular reaction. Shown are the diffusion sphere intersections at increasing time points. It is assumed here that $D_B > D_A$. Case 1: The OV contains neither μ_A nor μ_B. Case 2: The OV contains μ_A only, and is not identical to either diffusion sphere. Case 3: The OV contains μ_A only, and is identical to the diffusion sphere of A. Case 4: The OV contains μ_A and μ_B, but is not identical to either diffusion sphere. Case 5: The OV contains μ_A and μ_B, and is identical to the diffusion sphere of A.

A and B, t_{wait}, and the system time at which the state B was last updated, we can infer $t_{A-elapsed}$ and $t_{B-elapsed}$, the durations during which each had been diffusing before the reaction, which includes the waiting time to the reaction. Using $t_{A-elapsed}$ and $t_{B-elapsed}$ to define the diffusion spheres at the moment the molecules react, we can infer the OV case.

Case 2 begins when the radius of the faster diffusing particle (here, B) is equal to d, the distance between the Gaussian means of A and B. This radius can be computed as $R_B(t) = n_{sigma}\sqrt{6D_B t}$, where n_{sigma} is the number of standard deviations bounded by the sphere. See Fig. 3-5 for an illustration of the integration variables in Case 2. The starting time is given by

$$t_{start-2} = \frac{d^2}{6D_B n_{sigma}^2} \tag{3.24}$$

Starting times for cases 3-5 are calculated as follows:

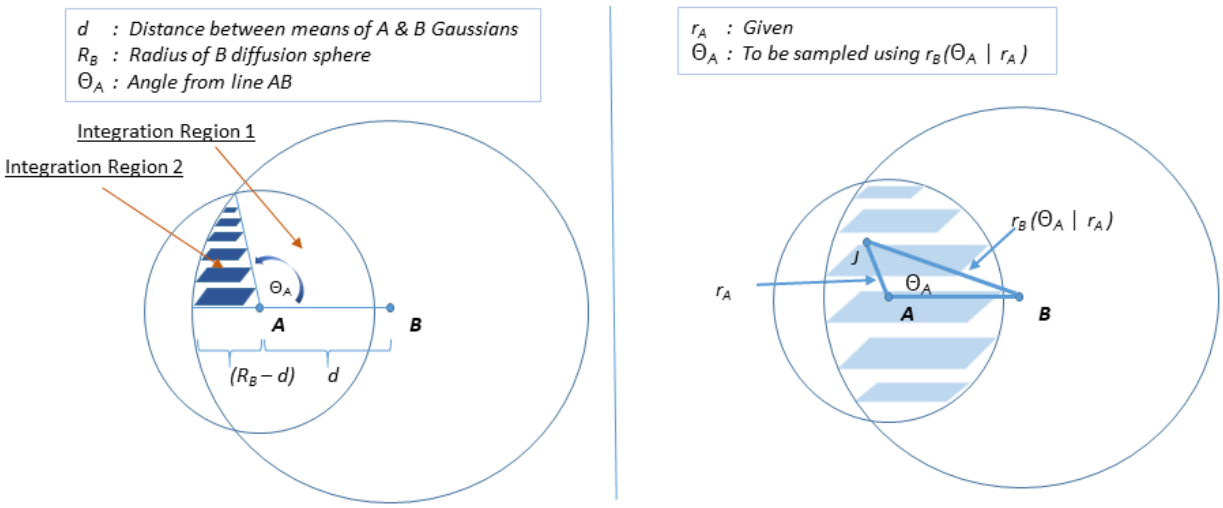


Figure 3-5: (**Left**) Visualizing the regions of integration for $w(r_A)$ in Case 2. (**Right**) Visualizing θ_A, r_A, and $r_B(\theta_A)$ in Case 2. The equiprobability ring passes through point J, perpendicular to the plane of the page.

Path 1	Case 3 Start	Case 5 Start
$D_B > 4D_A$	$d > R_A(t)$ $R_A(t) + d = R_B(t)$ $t_{start-3} = t_\gamma$	$d = R_A(t)$ $R_A(t) + d < R_B(t)$ $t_{start-5} = t_{\sim\gamma}$
Path 2	Case 4 Start	Case 5 Start
$D_A < D_B < 4D_A$	$d = R_A(t)$ $R_A(t) + d > R_B(t)$ $t_{start-4} = t_{\sim\gamma}$	$d < R_A(t)$ $R_A(t) + d = R_B(t)$ $t_{start-5} = t_\gamma$

where

$$t_\gamma = \frac{1}{6n_{sigma}^2(D_A - D_B)^2}(2D_A^2 n_{sigma}^2 \gamma + 2D_B^2 n_{sigma}^2 \gamma - 4D_A D_B n_{sigma}^2 \gamma + D_A d^2 + D_B d^2), \quad (3.25)$$

$t_{\sim\gamma} = \frac{d^2}{6D_A n_{sigma}^2}$, and $\gamma = \sqrt{\frac{D_A D_B d^4}{n_{sigma}^4 (D_A - D_B)^4}}$.

Case 1

Sampling r_A

In order to sample r_A correctly, we re-weight the probability density in the OV, i.e., compute a posterior probability. Define $h_{ring}(\theta_A)$ as the radius of the ring whose points are at distance r_A and for which the top most point defines a line with A at angle θ_A. The circumference of this ring is $2\pi h_{ring}(\theta_A)$. Integrating this circumference over the available θ_A range allows us to determine the size of the set of points at distance r_A.

$$p_{reweighted}(r_A, t) = w(r_A) * p(r_A, t) \tag{3.26}$$

with

$$w(r_A) = \frac{[TotalProbability - at - r_A]}{\int_{OV} dr\Big([TotalProbability - at - r] * p(r,t)\Big)},$$

$$\int p_{reweighted}(r_A, t) dr_A = \int w(r_A) p(r_A, t) = 1, \tag{3.27}$$

and

$$p(r,t) = \frac{1}{\sqrt{12\pi D_A t}} exp(-r^2/12D_A t) \tag{3.28}$$

$$\begin{aligned} w(r_A) &= \frac{\int_0^{\theta_{max}(r_A)} d\theta_A 2\pi h_{ring}(\theta_A)}{\int_{r_{lb}}^{r_{ub}} dr\Big[\big(\int_0^{\theta_{max}(r)} d\theta(r) 2\pi r \sin(\theta)\big) p(r,t)\Big]} \\ &= \frac{r_A\big(cos(\theta_{max}(r_A)) - cos(0)\big)}{\int_{r_{lb}}^{r_{ub}} dr\Big[r\big(cos(\theta_{max}(r)) - cos(0)\big) p(r,t)\Big]} \\ &= \frac{r_A\left(\frac{r_A^2 + d^2 - R_B^2}{2 r_A d} - 1\right)}{\Big[term1 + term2\Big]} \end{aligned} \tag{3.29}$$

$$term1 \quad = \quad \frac{1}{4d}(d^2 \; + \; 6D_A t \; - \; R_B^2)\big[erf\big(r_{ub}/\sqrt{12D_A t}\big) \; - \; erf\big(r_{lb}/\sqrt{12D_A t}\big)\big]$$

$$term2 \quad = \quad \frac{1}{\sqrt{12\pi D_A t}} 6D_A t\big[(r_{lb} \; - \; 2d)exp(-r_{lb}^2/12D_A t) \; - \; (r_{ub} \; - \; 2d)exp(-r_{ub}^2/12D_A t)\big]$$

The upper limit of integration, θ_{A-max}, is calculated by considering the triangle defined by the three points: A, B, I. The base (AB) length is d. The side BI has length R_B since I is the point at which (r_A, θ_A) intersects the OV, i.e., a point on the B diffusion sphere. The remaining side length is r_A. From the law of cosines, θ_{A-max} is calculated in terms of the side lengths.

$$\theta_{A-max}(r) = cos^{-1}\left(\frac{r^2 + d^2 - R_B^2}{2rd}\right) \tag{3.30}$$

The lower and upper bounds, r_{lb} and r_{ub}, on r_A defining the OV are $[(d - R_B), R_A]$.

Sampling $\theta_A|r_A, t$

The tuple (θ_A, r_A) uniquely defines a ring of equiprobability points within the OV from which a single reaction location can be chosen uniformly at random. Thus, the probability with which a given θ_A is sampled should be proportional to the size of the corresponding ring.

Consider the triangle defined by the points A, B, J where J is a point in the OV at (θ_A, r_A). The length of side BJ is $r_B(\theta_A)$ and can be computed with the Law of Cosines. The height of this triangle, h_{ring}, is again the radius of the ring passing through point J.

$$p(\theta_A|r_A, t) = p(r_B(\theta_A)|t) * RingCircumference$$
$$p(\theta_A|r_A, t) = \frac{1}{\sqrt{12\pi D_B t}}\, exp\left(-\frac{r_B(\theta_A)^2}{12D_B t}\right) * 2\pi h_{ring} \tag{3.31}$$

$$r_B^2(\theta_A) = r_A^2 + d^2 - 2r_A d\, cos(\theta_A) \tag{3.32}$$

$$h_{ring} = r_A\, sin(\theta_A) \tag{3.33}$$

$$\theta_A \in [0, \theta_{A-max}]$$

Case 2

Sampling r_A

$$w(r_A) = \int_0^{\theta_{A-max}(r_A)} d\theta_A 2\pi r_A sin(\theta_A) \; * \\ \Big(\int_0^{R_B-d} dr \big[\int_0^{\theta_{max}(r)} d\theta(r) 2\pi r \sin(\theta) \big] * p(r,t) \\ + \int_{R_B-d}^{R_A} dr \big[\int_0^{\theta_{max}(r)} d\theta(r) 2\pi r \sin(\theta) \big] * p(r,t) \Big)^{-1} \quad (3.34)$$

$$\theta_{A-max}(r) = cos^{-1}\Big(\frac{max[r^2, (R_B-d)^2] + d^2 - R_B^2}{2d \; max[r, (R_B-d)]} \Big) \quad (3.35)$$

Figure 3-5 provides a visual description of the relevant Case 2 variables. Variables for the other cases are defined similarly. For any r_A less than or equal to $(R_B - d)$, the full angular range of region 2 is available, i.e., $\theta \in (0, \pi)$. As r_A increases from $(R_B - d)$ to R_A, the available positions within region 2 decrease to 0. We capture this dependence with the angle integration limits, $(0, \theta_{A-max})$, where $\theta_{A-max} = \pi$ for $r_A \leq (R_B - d)$. The logic behind the form of $w(r_A)$ is analogous to case 1, however.

Sampling $\theta_A | r_A, t$

Sampling here is analogous to case 1, with updates to the available angle ranges for a given r_A.

$$p(\theta_A | r_A, t) = \frac{1}{\sqrt{12\pi D_B t}} \; exp\left(-\frac{r_B(\theta_A)^2}{12 D_B t} \right) * 2\pi h_{ring} \quad (3.36)$$

With $r_B^2(\theta_A) = r_A^2 + d^2 - 2 r_A d \; cos(\theta_A)$, $h_{ring} = r_A \; sin(\theta_A)$, and $\theta_A \in [0, \theta_{A-max}]$.

Case 3

Sampling r_A

In this case, the full range in r_A ($\in [0, R_A]$) is available. Therefore, no re-weighting of probabilities is needed.

$$p(r_A|t) = \frac{1}{\sqrt{12\pi D_A t}}\, exp\left(-\frac{r_A^2}{12D_A t}\right) \tag{3.37}$$

Sampling $\theta_A|r_A, t$

Sampling here is analogous to case 1, but with the full range of angles available.

$$p(\theta_A|r_A, t) = \frac{1}{\sqrt{12\pi D_B t}}\, exp\left(-\frac{r_B(\theta_A)^2}{12D_B t}\right) * 2\pi h_{ring} \tag{3.38}$$

With $r_B^2(\theta_A) = r_A^2 + d^2 - 2r_A d\, cos(\theta_A)$, $h_{ring} = r_A\, sin(\theta_A)$, and $\theta_A \in [0, \pi]$.

Case 4

Sampling r_A

Sampling here is analogous to case 2.

$$\begin{aligned} w(r_A) = \int_0^{\theta_{A-max}(r_A)} & d\theta_A 2\pi r_A sin(\theta_A)\ * \\ & \Big(\int_0^{R_B-d} dr \big[\int_0^{\theta_{max}(r)} d\theta(r) 2\pi r \sin(\theta) \big] * p(r,t) \\ & \quad + \int_{R_B-d}^{R_A} dr \big[\int_0^{\theta_{max}(r)} d\theta(r) 2\pi r \sin(\theta) \big] * p(r,t) \Big)^{-1} \end{aligned} \tag{3.39}$$

$$\theta_{A-max}(r) = cos^{-1}\Big(\frac{max[r^2, (R_B - d)^2] + d^2 - R_B^2}{2d\ max[r, (R_B - d)]}\Big) \tag{3.40}$$

Sampling $\theta_A|r_A, t$

Sampling here is also analgous to case 2.

$$p(\theta_A|r_A, t) = \frac{1}{\sqrt{12\pi D_B t}}\, exp\left(-\frac{r_B(\theta_A)^2}{12D_B t}\right) * 2\pi h_{ring} \tag{3.41}$$

With $r_B^2(\theta_A) = r_A^2 + d^2 - 2r_A d\, cos(\theta_A)$, $h_{ring} = r_A\, sin(\theta_A)$, and $\theta_A \in [0, \theta_{A-max}]$.

Case 5

Sampling r_A

In this case, the full range in r_A ($\in [0, R_A]$) is available. Therefore, no re-weighting of probabilities is needed.

$$p(r_A|t) = \frac{1}{\sqrt{12\pi D_A t}}\, exp(-\frac{r_A^2}{12D_A t}) \tag{3.42}$$

Sampling $\theta_A|r_A, t$

Sampling here is analogous to case 1, but with the full range of angles available.

$$p(\theta_A|r_A, t) = \frac{1}{\sqrt{12\pi D_B t}}\, exp\left(-\frac{r_B(\theta_A)^2}{12D_B t}\right) * 2\pi h_{ring} \tag{3.43}$$

With $r_B^2(\theta_A) = r_A^2 + d^2 - 2r_A d\, cos(\theta_A)$, $h_{ring} = r_A\, sin(\theta_A)$, and $\theta_A \in [0, \pi]$.

3.5.3 Determining Bimolecular Reaction Locations by Rejection Sampling

Because PDFs in each case may be complicated functions, we cannot always sample from them directly. Instead, we first draw a sample of our variable x (i.e., r_A or θ_A) uniformly from its feasible range. In order to determine whether this sample is accepted or rejected, we utilize an envelope function, $Q(x)$ whose probability density at all feasible points is at least as great as that of the PDF from which we want an observation. The sample x is accepted if $q(x)$ drawn uniformly from $[0, Q(x)]$ is less than $p(x)$.

One potential issue is that volume exclusion should prevent sampled locations from leading to particle overlap. We define two molecules to be overlapping if the distance between their Gaussian means is less than R, the minimum allowed separation, *and* neither molecule has been diffusing for longer than $R^2/6D$. In the event overlap is detected, a new location is sampled. This procedure for handling volume exclusion in location sampling is not optimized for efficiency and can significantly impact the run time as the particle density increases.

3.5.4 Simulation Boundaries

Figure 3-6 illustrates our method for ensuring all particles remain within the simulation volume. We treat this volume as a cube bounded by planes about which a particle may be reflected if its initially sampled position exceeds the plane. The algorithm samples an unconstrained reaction location and the displacements for both particles are noted. Next, assume the reaction location happens to be outside the simulation volume. Each molecule can be considered to have travelled along a linear path from its initial location to the reaction location, with one piece of the path within the simulation volume and one piece outside. Because the unconstrained spatial probability densities describe radial displacements from either particle's initially known location, application of reflective boundary conditions need only guarantee both particles' piecewise linear paths each sum to the noted displacements, and terminate within the simulation volume.

This procedure is strictly correct only if the wait time sampling, i.e., computing the integrated reaction propensities, is correct. The point (finite) particle reaction propensities described in this paper do not take into account the boundaries of the simulation volume. Error is therefore introduced in wait time sampling for molecules diffusing long enough to encounter a boundary. However, given a wait time t, reaction location sampling depends only on the possible net displacements of either particle after diffusing for t. We can therefore assume free diffusion to sample the location and then use our reflecting procedure if necessary.

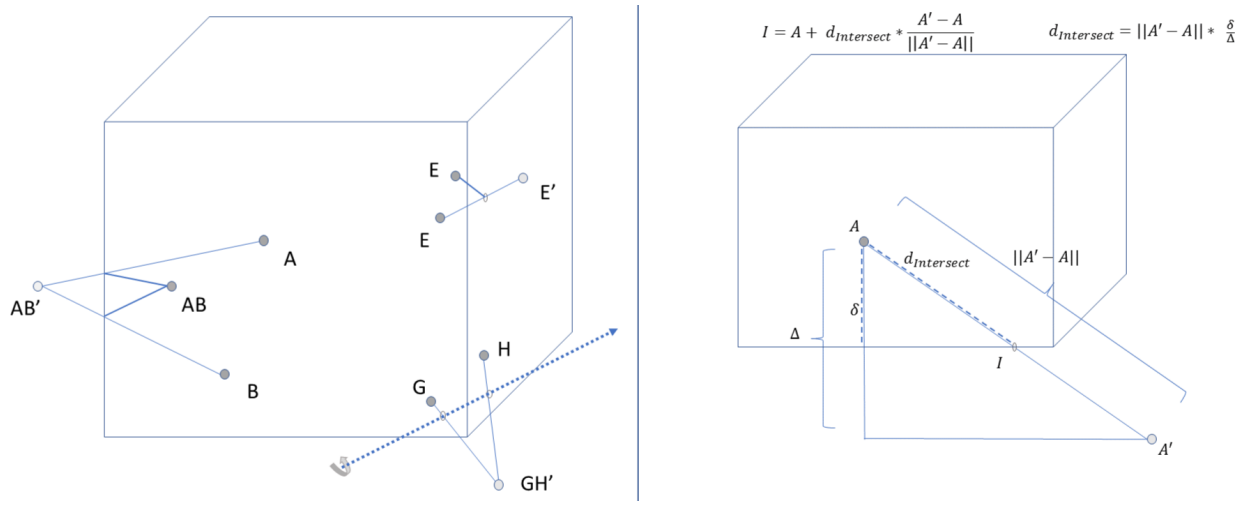


Figure 3-6: (**Left**) Shown are applications of the reflective boundary condition after a position-only-update event (e.g., E) or after a bimolecular reaction event (e.g., A&B, G&H). (*Single Reflection*) In the bimolecular case, we reflect about an axis defined by the two intersection points of the lines connecting the reactants with the product, and the boundary. This ensures that the distances traveled by both particles remains the same. When these lines exit the simulation box through the same face (e.g., A&B), the reflection axis is parallel to the face. When the lines exit though different faces (e.g., G&H), the axis must be computed and the reflection can be implemented with the Rodrigues rotation formula in the appropriate reference frame. (*Multiple Reflection*) Depending on the location of the reactants and the distances they travel, the post-reflection location may end up outside a different boundary, though to a lesser extent. We simply need to update the reactant positions to be the boundary intersection point(s) and reapply the reflection procedure. In principle, this procedure works for any simulation volume, including those with curved boundaries. (**Right**) For a cubic simulation volume, we determine through which face (and at what point) a reactant (A) first passed if it is found outside the simulation volume. In this case, the pre-reflection location A' exceeds the simulation volume along more than 1 dimension which means it is necessary to compute $d_{Intersect}$ for each 2d plane exceeded by A', and then compute the intersection point, I, for the face with minimum $d_{Intersect}$.

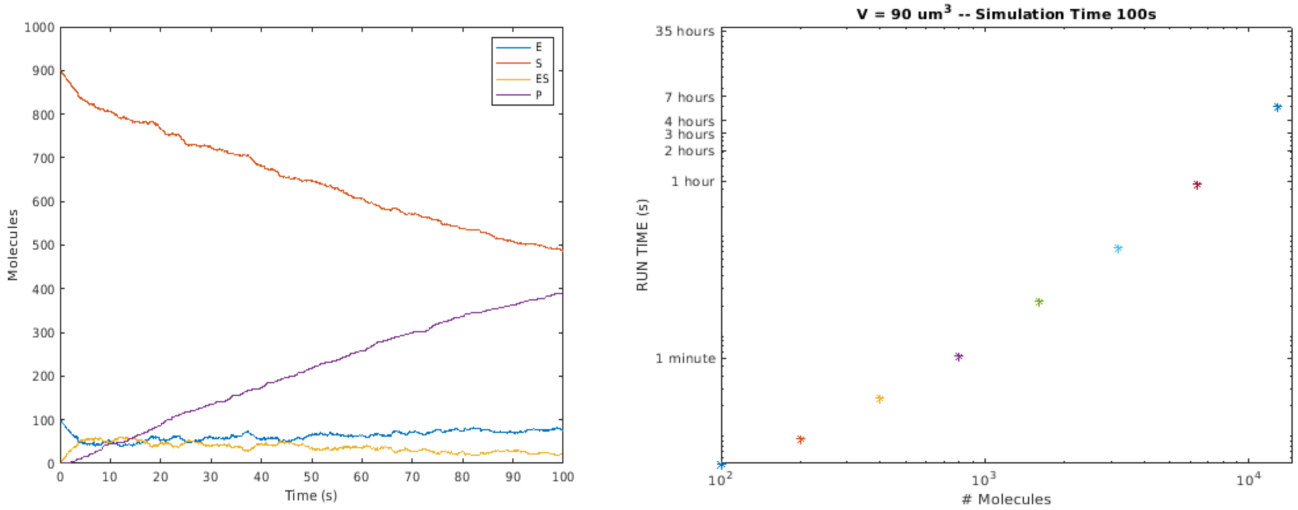


Figure 3-7: Point particle representation. (**Left**) Time evolution of 1000 molecules in the Michaelis-Menten model with DESSA-CS. Unimolecular rate constant $k_{uni} = 0.1s^{-1}$ (governing $ES \Rightarrow E+S$ and $ES \Rightarrow E+P$) and diffusion coefficient $D = 1\mu m^2 s^{-1}$ are taken from Figure 5 of Chew et al. [49]. In order to reproduce similar dynamics, we chose the intrinsic bimolecular rate constant $k_{bimol} = 2.5{*}10^7 s^{-1}$ (with $R^2_{enc} = 0.01^2 \mu m$). (**Right**) The run time for the model increases roughly linearly in log space with the number of molecules - [100,200,400,800,1600,3200,6400,12000,24000].

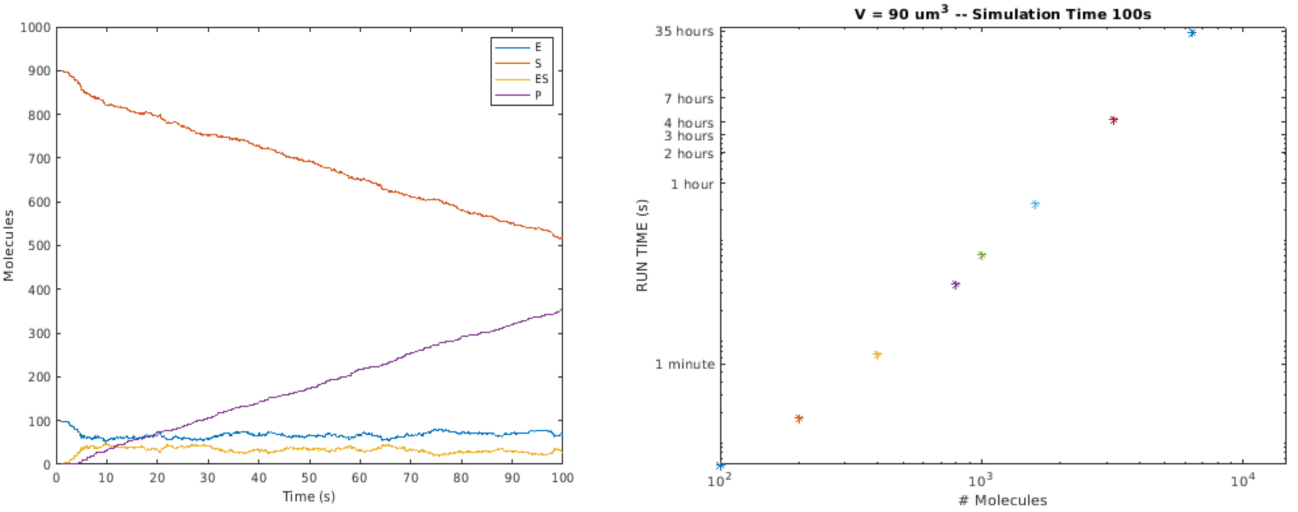


Figure 3-8: Finite particle representation. (**Left**) Time evolution of 1000 molecules in the Michaelis-Menten model with DESSA-CS. Unimolecular rate constant $k_{uni} = 0.1s^{-1}$ (governing $ES \Rightarrow E+S$ and $ES \Rightarrow E+P$), diffusion coefficient $D = 1\mu m^2 s^{-1}$, intrinsic bimolecular rate constant $k_{bimol} = 5 * 10^{-1} s^{-1}$, and $R_{enc} = 0.01\mu m$ are taken from Figure 5 of Chew et al. [49]. (**Right**) The run time for the model increases roughly linearly in log space with the number of molecules - [100,200,400,800,1600,3200,6400,12000].

1000 Molecules, 100s Simulation Time				
Local Workstation: Ubuntu 14.04 LTS, 128 GB memory, Intel Xeon E5-2630 2.40GHz				
[49] Workstation: Ubuntu 16.04 LTS, 48 GB memory, Intel Xeon X5680 3.33GHz				
Software	Run Time	Sim Parameters	Space / Time Steps	Workstation
DESSA-CS	100s	$r = 0nm$, $l_B = 0.1$	off-lattice / sampled	local
DESSA-CS	728s	$r = 10nm$, $l_B = 0.1$	off-lattice / sampled	local
eGFRD	10,561s	$r = 10nm$	off-lattice / variable	local
eGFRD	2,412s	$r = 1nm$	off-lattice / variable	[49]
eGFRD	3,246s	$r = 10nm$	off-lattice / variable	[49]
Smoldyn	20s	$\Delta t = 1ms$	off-lattice / fixed	[49]
Smoldyn	298s	$\Delta t = 67\mu s$	off-lattice / fixed	[49]
Spaciocyte MLM	13s	$\Delta t = 1ms$, $r = 38.73nm$	spatial lattice / fixed	[49]
Spaciocyte MLM	276s	$\Delta t = 67\mu s$, $r = 10nm$	spatial lattice / fixed	[49]

Table 3.1: Method Comparison on Updated Benchmark from Chew et al. [49]. Diffusion coefficients are $1\mu m^2 s^{-1}$. The max allowed diffusion time was 40s. The local eGFRD simulation was run using the open source simulation environment E-Cell version 4 [138].

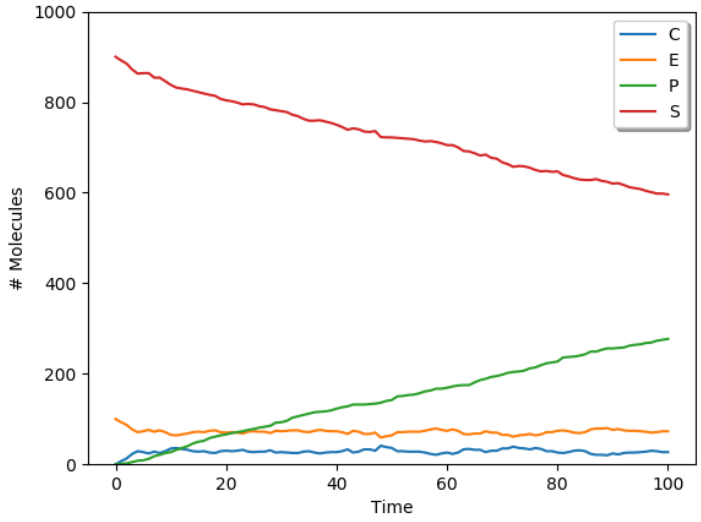


Figure 3-9: Time evolution of 1000 molecules in the Michaelis-Menten model with eGFRD in the E-Cell v4 environment. Unimolecular rate constant $k_{uni} = 0.1s^{-1}$ (governing $ES \Rightarrow E + S$ and $ES \Rightarrow E + P$), diffusion coefficient $D = 1\mu m^2 s^{-1}$, intrinsic bimolecular rate constant $k_{bimol} = 1 * 10^{-2} s^{-1}$, and particle radius $r = 0.01\mu m$ are taken from Figure 5 of Chew et al. [49]

3.6 Results

3.6.1 Application: Michaelis-Menten

We applied DESSA-CS to the well known Michaelis-Menten enzymatic reaction system within a $90\mu m^3$ volume. The original benchmark was developed by Andrews [6] and updated by Chew et al. [49] to account for the extreme run time demands of eGFRD. Figures 3-7 and 3-8 display our results for the updated benchmark, displaying the population dynamics for molecular species E, S, ES and P, which obey the binding rules $E + S \Leftrightarrow ES \Rightarrow P$. Figure 3-9 shows comparable results for eGFRD. Simulation run times for the point particle and finite particle representations respectively were 20 seconds and 90 seconds, roughly two orders of magnitude faster than eGFRD (see Figure 5 of Chew et al. [49] and Table 3.1).

In the point particle simulations, the data set (computed before run time) consisted of 3000 linearly spaced distances from R_{enc} to 2*d_{maxD}, where d_{maxD} corresponds to the mean square displacement due to diffusion at T_{maxD}, the max allowed diffusion time. At each distance, the integrated propensity was computed at 50,000 time points (i.e., variances). There were 40,000

linearly spaced time points from $1 * 10^{-6}$ to $1 * 10^{-2}$ where the curvature is often highest, and 10,000 linearly spaced time points from $1 * 10^{-2}$ to T_{maxD}. This 3000 by 50,000 data set was computed in 14.5 minutes in the Go language (golang). The propensity function integration error for a given distance value and time duration depends on the number of integration intervals - here 50,000 for the full duration. We used the trapezoid method, whose error at each time point can be upper bounded by $Err(\Delta t) = \frac{\Delta t^3}{12N^2} * K * c$, where K is the maximum magnitude of the second derivative of the CDF, c is the intrinsic reaction rate, and N is the number of integration intervals over the duration Δt. These are not likely to be tight upper bounds due to the presence of inflection points in the CDF graph. Our golang integration code, including a method to print error bounds and integrals to text files, can be found in the GitHub repository.

In the finite particle simulations, the data set consisted of 3000 integrated propensity curves at the same distance values, each of which was evaluated at 5500 time points. Computation of the integrated propensity data set required $\sim 1m$. Performing these numerical integrations is possible in Matlab but requires symbolic computation to evaluate the integrands. The result was that the same data set takes on the order of days to compute. In golang, the necessary numeric precision was achieved using its big math package which implements arbitrary-precision arithmetic.

In general, both the finite and point particle representations run more efficiently when each molecule is allowed to diffuse farther outside the boundaries before wait times are sampled. If we refer to cubic simulation volume as having dimension $(L\mu m)^3$ and the fraction of the cube length beyond which a particle may diffuse as l_B, then it is useful to analyze the behavior of simulations as these vary. The Michaelis-Menten Benchmark requires that $90\mu m^3 = L(1 + l_B)$. In Figure 3-10 we plot simulation trajectories at multiple values of l_B (i.e., $l_B = 0.03$, $l_B = 0.1$, and $l_B = 0.3$) for the finite representation and it is apparent that while the kinetics do not change significantly, the run time does. The respective run times are 2693s, 777s, and 157s. Figures 3-11 and 3-12 illustrate this more fully for both representations. As the relative distance allotted to the cube's length increases, so does the run time. This results from the fact that as l_B decreases, so does the maximum diffusion time of molecules near the boundary, leading to a much higher number of position updates compared with the roughly unchanging number of reaction events. Even though the kinetics do not change significantly, they do depend on l_B. The effective association rate is inversely proportional to l_B. Of course, if the wait time sampling procedure – specifically the

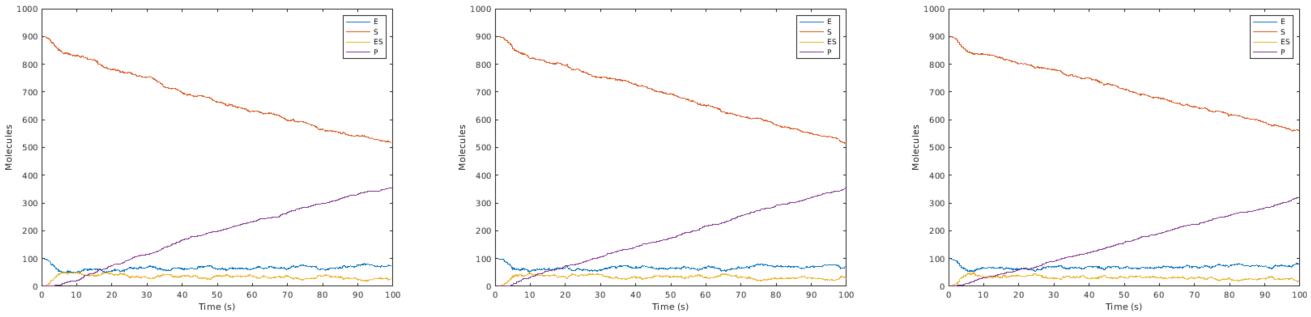


Figure 3-10: Varying the simulation volume's boundary parameter, finite particle representation. Time evolution of 1000 molecules in the Michaelis-Menten model with DESSA-CS. Unimolecular rate constant $k_{uni} = 0.1s^{-1}$ (governing $ES \Rightarrow E + S$ and $ES \Rightarrow E + P$), diffusion coefficient $D = 1\mu m^2 s^{-1}$, boundary condition parameter $c = 1.2 * 10^3 s^{-1}$, and $R_{enc} = 0.01\mu m$. From left to right: $l_B = 0.03$ (runtime = 2693s), $l_B = 0.1$ (runtime = 777s), and $l_B = 0.3$ (runtime = 157s).

calculation of integrated propensities – took into account the simulation walls, reaction locations could be determined without incurring error by sampling locations as if no boundary wall existed and then applying our reflection procedure to return out of box positions to the simulation volume. Future work will consider methods for updating our free space reaction propensities in this way.

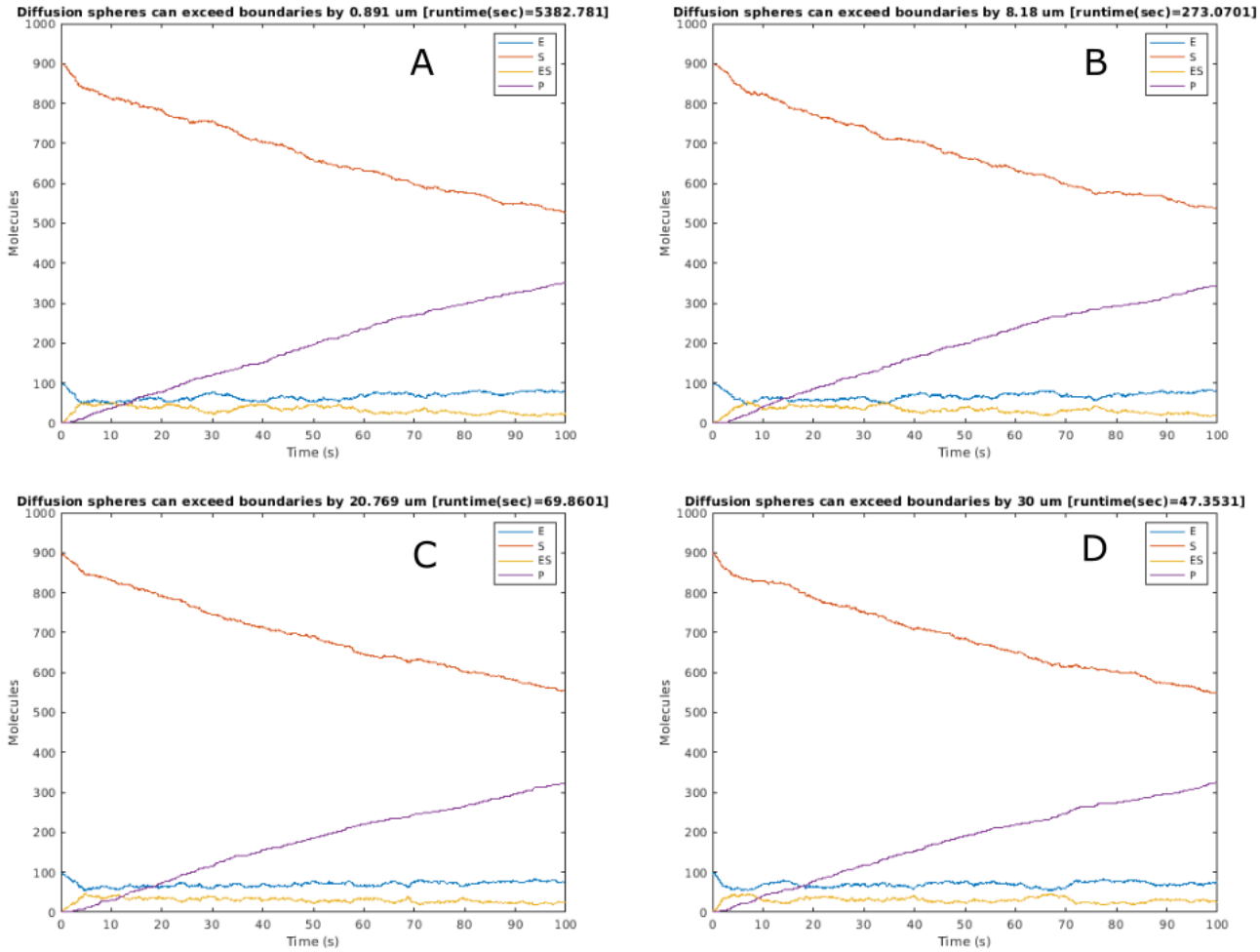


Figure 3-11: Finite Particle Representation. Varying the boundary parameter l_B in the Michaelis-Menten Benchmark test. (A) $l_B = 0.01$, run time 5382s. (B) $l_B = 0.1$, run time 273s. (C) $l_B = 0.3$, run time 69.8s. (D) $l_B = 0.5$, run time 47.3s. The kinetics depend weakly on l_B. Intrinsic reaction rate 5e-1s^{-1}

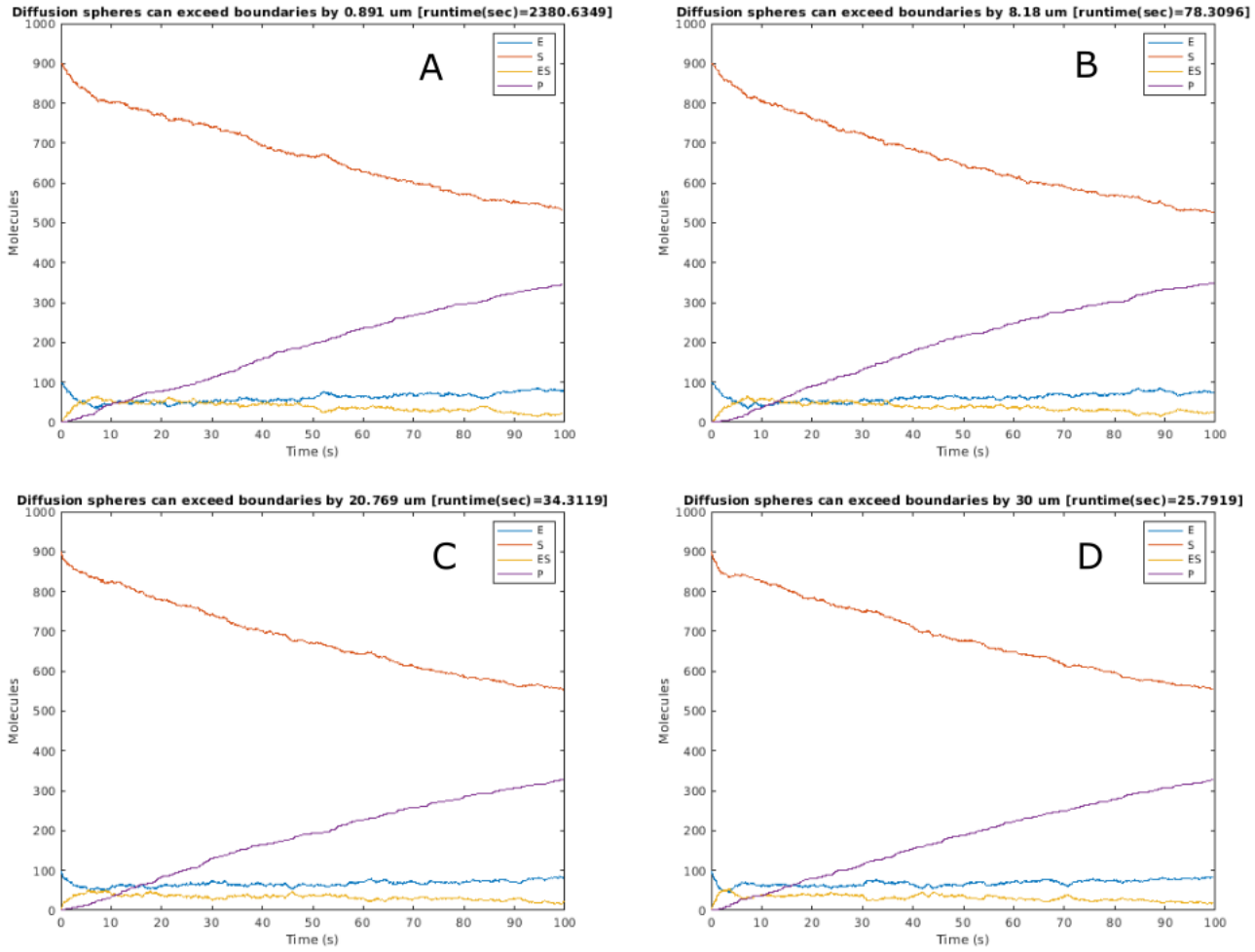


Figure 3-12: Point Particle Representation. Varying the boundary parameter l_B in the Michaelis-Menten Benchmark test. (A) $l_B = 0.01$, run time 2380s. (B) $l_B = 0.1$, run time 78.3s. (C) $l_B = 0.3$, run time 34.3s. (D) $l_B = 0.5$, run time 25.7s. The kinetics depend weakly on l_B. Intrinsic reaction rate 2.5e7s^{-1}.

Chapter 4

Immune Cell Signaling

4.1 Project Background

In previous work [215], Roybal and colleagues used fluorescence microscopy in conjunction with a computational image analysis pipeline to investigate the mechanism by which costimulation regulates actin dynamics in T cells. T cells are activated primarily by direct interaction with antigen presneting cells (APCs) through recognition of specific peptides on the APC surface. Essential to this recognition process is the parallel engagement of costimulatory receptors on the T cell, e.g., costimulatory receptor 28 (CD28), with their associated ligands on the APC, e.g., the B7 family ligands CD80 and CD86.

T cell activation stimulates a rapid and transient accumulation of actin in a region known as the immunological synapse at the interface with the APC. The precise mechanisms by which costimulation contributes to the regulation of T cell actin dynamics are unknown, however these dynamics have been shown to be critical for many aspects of T cell functioning, including spatiotemporal organization, APC coupling and transcription regulation [271, 187, 275, 153, 216]. Additionally, because actin regulation is a complex phenomenon dependent on the integrated interactions of numerous key regulators, each of whose functions has been established individually by genetic means, there is a need to focus on system function upon physiological perturbation rather than on individual protein function per se. For this reason, Roybal et al. [215] carried out flourescence experiments and computational analyses designed to monitor changes in the behavior of eight core actin regulators under two conditions: full stimulus and a costimulation-blocked

perturbation.

Building on this recent work, we investigate whether simple kinetic models can account for the spatially dependent concentration trajectories of actin and its core regulators observed in T cell-APC conjugates under each condition. The broader goal is to consider the role of spatial organization in signaling networks and to learn relationships between proteins without assuming a priori knowledge of the functions of the network derived from traditional experiments, e.g., genetic screens of single targets.

The kinetic model we utilize assumes actin and its regulators can each interact with T-cell receptors (TCR), but not with each other. More complex models with higher order interactions (e.g., proteins A and B must interact with TCR before protein C could) were tentatively explored, however we concluded that they offered too much flexibility to be reliably fit. While the model we describe is certainly biologically incomplete, conclusions drawn from it may still be valid and relevant.

4.2 Methods

Data Preprocessing

The subcellular localization of actin and seven primary actin regulators was determined by in vitro fluorescence imaging [215]. These are: Actin, ARP3, Cofilin, Coronin1A, CPalpha1, HS1, WASP and WAVE2, and they are referred to as proteins 1-8. The experimental procedure, including green fluorescent protein (GFP) tagged imaging, was performed sequentially for each actin regulator. Voxel intensities were normalized to the fraction of total fluorescence for each "sensor", i.e., for each GFP tagged protein type, and average concentrations of the sensors were earlier determined in [215]. Thus, we have access to absolute concentrations within each voxel. Following Ruan et al. [218], we began with voxel level concentration data (6628 voxels x 8 proteins x 12 time points) which was subsequently k-means clustered into spatial regions showing similar spatiotemporal patterns. Various numbers of clusters were originally tried with the result that three gave the highest Calinski Harabasz criterion, a metric for evaluating cluster performance when no ground truth is known. We therefore clustered all proteins and time points with $k = 3$

to establish the regions later identified with the immunological synapse (region 1), the cytosol (region 2) and the nucleus (region 3). Fluorescence intensities were measured at twelve time points relative to synapse formation: -40, -20, 0, 20, 40, 60, 80, 100, 120, 180, 300, and 420s. All concentrations are expressed in μM.

ODE rule-based model

We implemented a BioNetGen [83] model based on the following reaction rules and rates.

$$p^i_{synapse} \longleftrightarrow p^i_{cytosol} \quad (k^i_{SC},\ \ k^i_{CS})$$

$$p^i_{synapse} \longleftrightarrow p^i_{nucleus} \quad (k^i_{SN},\ \ k^i_{NS})$$

$$p^i_{nucleus} \longleftrightarrow p^i_{cytosol} \quad (k^i_{NC},\ \ k^i_{CN})$$

$$\mathbf{0} \longrightarrow TCR \quad (k_{activate},\quad t < 0)$$

$$TCR \longrightarrow \mathbf{0} \quad (k_{deactivate},\quad t > 0)$$

$$p^i_{synapse} + TCR \longleftrightarrow [p^i_{synapse}.TCR] \quad (k^i_{on},\ \ k^i_{off},\quad t < 0)$$

$$p^i_{synapse} + TCR \longleftrightarrow [p^i_{synapse}.TCR] \quad (f_{on} * k^i_{on},\ \ f_{off} * k^i_{off} * exp(-t * h^i_{decay}),\quad t > 0)$$

These ODEs describe an independent binding model in which the concentration trajectory of each protein does not depend on concentrations of the others. At the third time point ($t = 0$), the TCR activation ceases and deactivation begins. When bound to the TCR, proteins remain immobile and within the synapse region. Only after dissociation can a TCR molecule deactivate, thereby preventing further binding reactions. It is important to note, however, that binding may still occur in the synapse beyond $t = 0$ as long as the population of TCR remains above zero. Additionally, at the third time point the binding rates switch to their post synaptic formation values with the off rate allowed to decrease over time by means of the exponential decay factor. The decrease in the late time off-rate was usually minor or zero, but included to tune the rate of return to equilibrium concentrations following the spike in the synapse region.

Within RuleBender [112], a Matlab script and associated Cvode-mex implementation of the BioNetGen model were generated, allowing us to integrate the ODE solutions within a larger Matlab pipeline. Numerical integration was carried out by CVODE, a package within the Sundials suite of equation solvers [52].

The Objective Function

Note: The objective function design and parameter optimization methodology were both developed at an early stage of the work before we incorporated BioNetGen and Cvode to handle the numerical integrations. At that time, simulation of the model entirely within Matlab was orders of magnitude slower. The need to minimize the number of objective function evaluations led us to consider modifications to a traditional RMSD objective function. Additionally, we initially tried SNOBFIT as well as our novel optimization algorithm described in Chapter 2 without much success, leading us to develop the parameter optimization methodology described in the next section.

We treat the data fitting problem as an objective function (error) minimization. While the root mean square deviation (RMSD), evaluated element wise at each time point between model's output and the experimental data, is often the appropriate choice of error function, we found it useful to alter it in several ways. First, both data sets were scaled such that all values subsequent to the $t = -40$ value of a given curve represent percentage change relative to the $t = -40$ value. This helped to make the optimization equally sensitive to concentrations in each region. Second, we added a vector term to the error that relates concentrations at two time points as opposed to the single time point used with RMSD. It was often the case that two models with differently shaped curves produced roughly equal RMSDs. With the addition of the vector error, the curve whose shape better matched the experimental data was assigned a lower overall objective value. Summations are over all regions and time points simulated. The resulting objective function can be summarized as:

$$F(\mathbf{x}) = \Sigma \ RMSD(\mathbf{x}) + \Sigma||\mathbf{r} - \mathbf{s}(\mathbf{x})||$$

where

$$\mathbf{r} = [realData(t_n) - realData(t_{n-1})]/[t_n - t_{n-1}]$$

$$\mathbf{s} = [simData(t_n) - simData(t_{n-1})]/[t_n - t_{n-1}]$$

Parameter Optimization Methodology

Our ODE model consists of 13 parameters: the region transfer rates ($k^i_{SC}, k^i_{CS}, k^i_{SN}, k^i_{NS}, k^i_{NC}, k^i_{CN}$), the TCR activation rates ($k_{activate}, k_{deactivate}$), the early time binding rates (k^i_{on}, k^i_{off}), the late time factors (f_{on}, f_{off}) which when multiplied with the early rates give the late time binding rates, and the late time off rate exponential decay parameter h^i_{decay}. Due to the simplifying independent binding assumption, we were able to optimize each sensor's (i.e., each protein's) parameters separately. After manual trial and error, approximate ranges were determined for each parameter and used to construct a large grid from which to start objective function minimization.

Our optimization strategy is a heuristic grid search. It begins by identifying promising regions from a large initial grid and, based on these regions, constructs finer grids over more limited parameter ranges. The initial grid is computed using Matlab's *ngrid* function with a set of grid vectors specifying the range of values for each parameter as input. In this way, all combinations of parameter values are used. The initial grid's objective values are sorted from lowest to highest and then divided into a number of groups.

	Six Region Transfer Rates	TCR On Rate	TCR Off Rate	On Rate	Off Rate	Late On Factor	Late Off Factor	Late Off Decay
Grid Vector Min	0.001	0.0904	0.45	0.05	0.01	0.001	0.05	0
Grid Vector Max	0.2	0.6	0.45	2	1	1	1	0
Grid Vector Length	5	4	1	5	5	5	5	1

Table 4.1: Initial Grid of size [39,062,500,13]. Each of the 13 grid vectors is linearly spaced from its minimim to maximum values. These parameter ranges were discovered from manual trial and error.

The optimization algorithm begins by training a decision tree classifier using the grid points as observations and their associated group rankings as class labels. In this way, we learn the relationships between parameters that distinguish points with lower and higher objective values.

For example, group 1 (containing points with the lowest objective values) may be described by a number of paths in the decision tree, each leading to leaf nodes with class label 1. These paths are logical conjunctions, e.g., $param1 <= a \,\&\, param3 > b \,\&\, param6 >= c \,\&\, ...$ for numerical constants $a, b, c, etc.$ The method we used to divide the sorted objective values into groups is k-means clustering. Due to memory limitations in Matlab on our machine, only 20,000 points are used: the top 10,000 points and 10,000 logarithmically scaled points from those remaining. The number of clusters can be chosen manually. Our default was $k = 30$ and the top 10 clusters merged into a single cluster. The decision tree together with input points belonging to cluster 1 are then used to generate a "local grid" constructed from grid vectors for each parameter. These grid vectors' lower and upper bounds ($''lb,ub''$) are first set to the minimum and maximum parameter values seen in the cluster, respectively. Next, each is extended by an amount equal to the standard deviation of the corresponding parameter evaluated over all points in the cluster:

$$lb_{param\ j} = min_{Cluster1}(param\ j) - |std_{Cluster1}(param\ j)|$$

$$ub_{param\ j} = max_{Cluster1}(param\ j) + |std_{Cluster1}(param\ j)|$$

If a lower bound becomes negative, it is set to zero. For any parameter appearing in the DT path's logical conjunction, the appropriate bound is updated to that given in the path. In this way, the algorithm retains information about what parameter ranges lead to low objective values. The length of each grid vector determines the resolution of the grid. Generally, higher resolution is desirable but must be balanced by memory and run time constraints as the total number of points grows quickly. We would like higher resolution along parameters whose objective values vary on shorter length scales and less resolution along parameters whose objectives vary more slowly. We infer the relative differences in length scales by training a Gaussian process (GP) model of the objective function on the top 10,000 points from the initial large grid. The kernel function used is the rational quadratic with automatic relevance determination, i.e., a separate length scale hyperparameter for each dimension j of the input. Each learned kernel hyperparameter, $l_k(j)$, is sensitive to the number of training points and so are not used directly as the grid vector spacings. However, their rankings are fairly stable to changes in training set size, e.g., it may always be the case that $l_k(3) > l_k(4)$ suggesting the model is less sensitive to changes in $param3$

than it is to changes in $param4$. The length of each region transfer rate's grid vector (i.e., resolution f_j) is set according to the formula: $f_j = ceil[max(\mathbf{l_k})/l_k(j)]$. These values are then individually raised or lowered as needed. For example, if the resulting grid vector is too long ($[\prod_{j=1}^{6} f_j] > UpperThreshold$), each f_j is lowered by one starting with the highest value. Similarly, if the resulting grid vector's length is too low, it is raised to the lower threshold, e.g., 3. The grid resolutions for early time binding rates are defined similarly: $f_j = ceil[log(max(\mathbf{l_k})/l_k(j))] + 2$; as are the late time factors: $f_j = ceil[log(max(\mathbf{l_k})/l_k(j))] + 2$. Again, as with the six region transfer rate resolutions, the four binding rate resolutions are individually raised to a lower threshold (e.g., 2) or lowered by one in decreasing order if $[\prod_{j=9}^{12} f_j] > UpperThreshold$.

In addition to the local grid, we also generate an "exploratory grid" constructed from grid vectors based on all decision tree paths in the top N clusters. In our simulation experiments, points from the local grid tend to improve on already good fits from the initial large grid. However, for certain proteins (e.g., Actin, which requires larger TCR activation rates), only points from the exploratory grid significantly improve on the best point from the initial large grid.

4.3 Results

In all population trajectory figures, the simulation time on the x-axis starts at 0, with all subsequent time points offset by 40s compared with the measured fluorescence intensities described in the Data Processing section. The real data is shown in black and our simulations are shown in red. The 13 parameters and objective function value are displayed directly above the corresponding plots. We begin in Figures 4-1 - 4-8 with the results of optimizing a single model in which each of the 13 parameters is shared across all proteins. The sum of errors across all proteins was approximately minimized. If this simpler, less flexible model were capable of explaining the data there might be little reason to expect a more flexible model to be necessary.

However, it is clear that while certain of the proteins' simulated trajectories agree well with the experimental data, many do not. While a truly exhaustive search was not possible, We conducted simulations in larger search spaces and with finer grids with the no qualitative differences in the outcome. That is, different parameter sets can be found allowing some other combination of proteins (/regions) to be fit well, but with many other fit poorly.

We next moved to the independent binding model where the parameter sets are specific to each protein. Figures 4-9 - 4-16 show the best fits for these parameter optimizations.

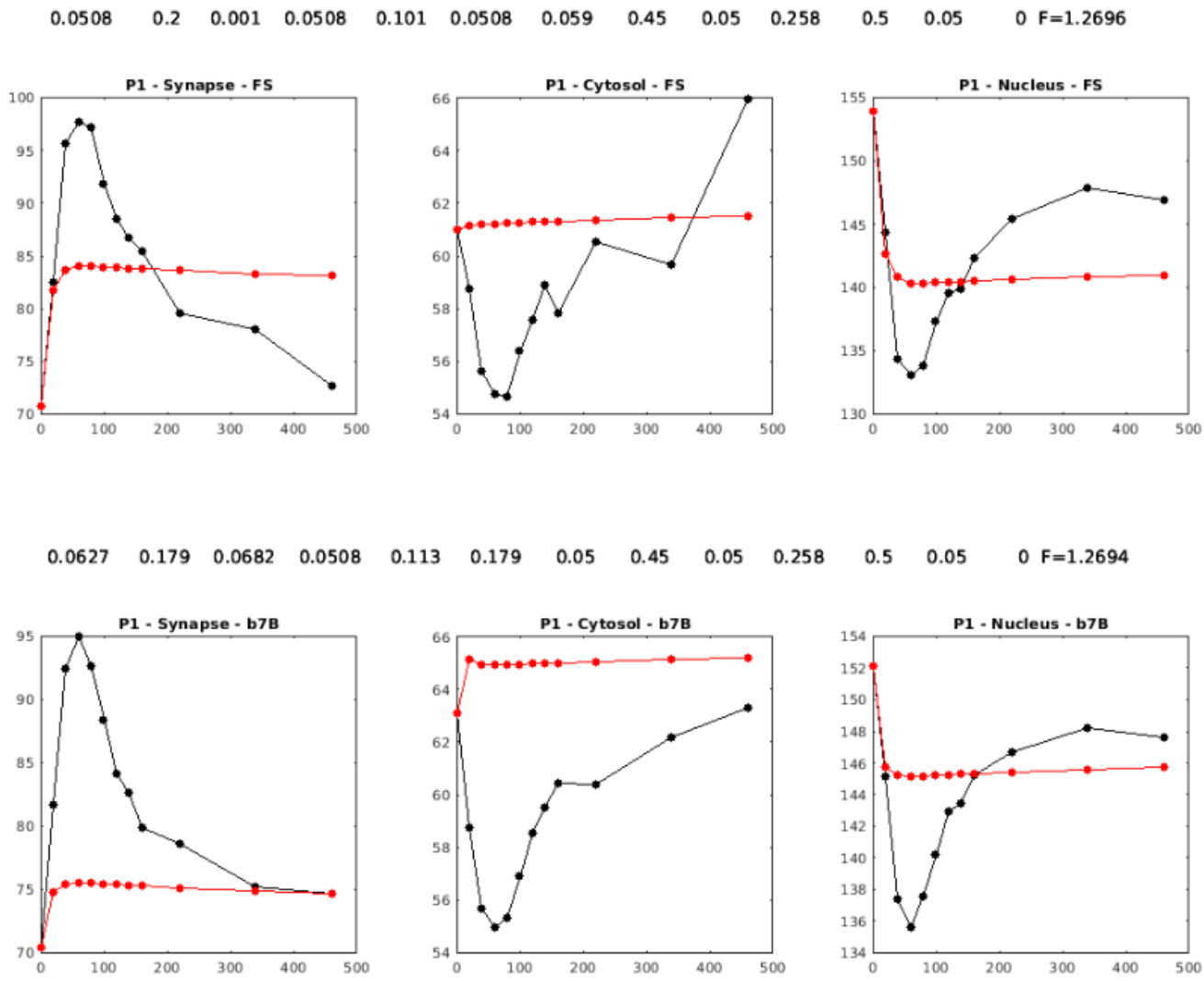


Figure 4-1: Protein 1, Actin. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

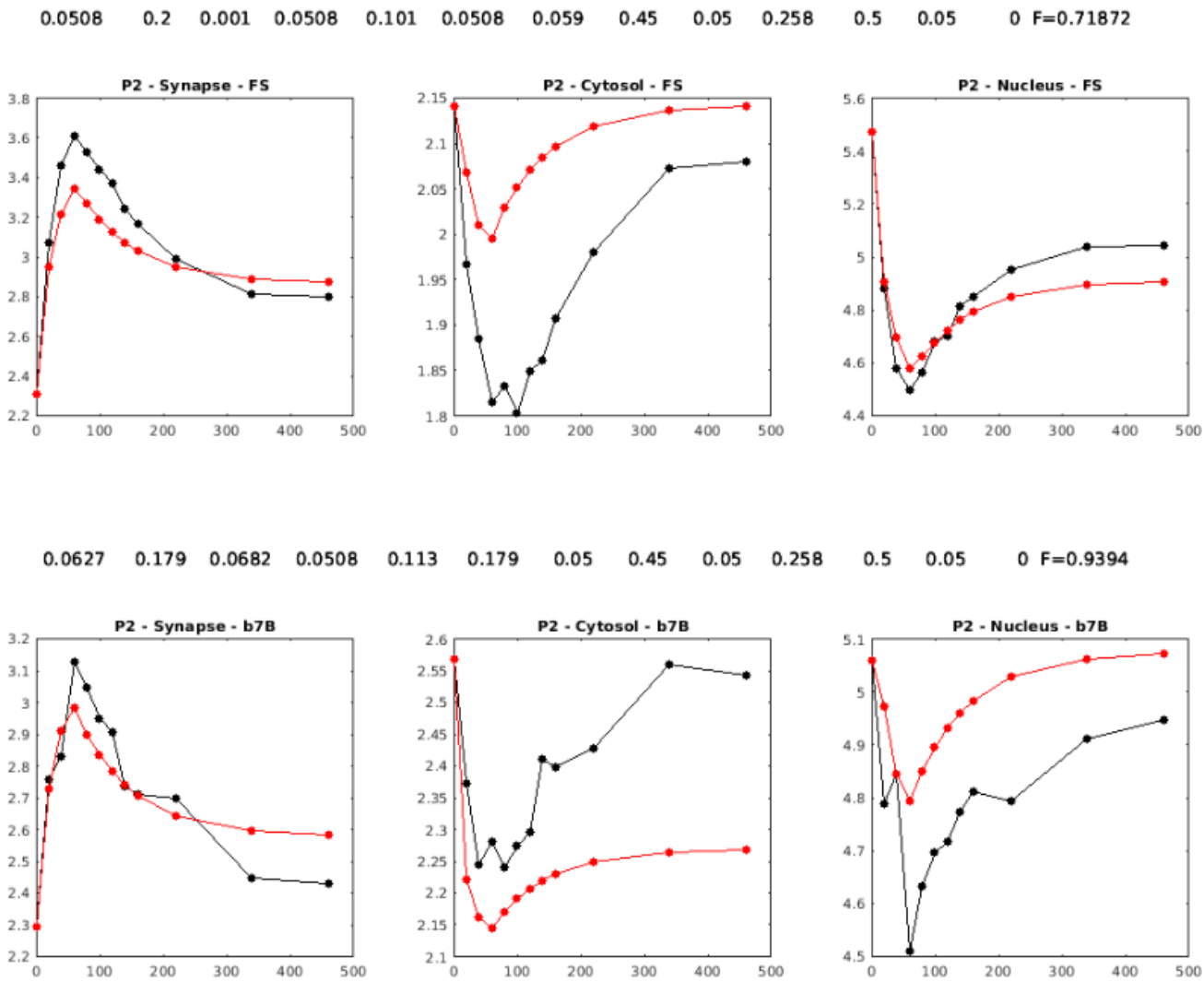


Figure 4-2: Protein 2, ARP3. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

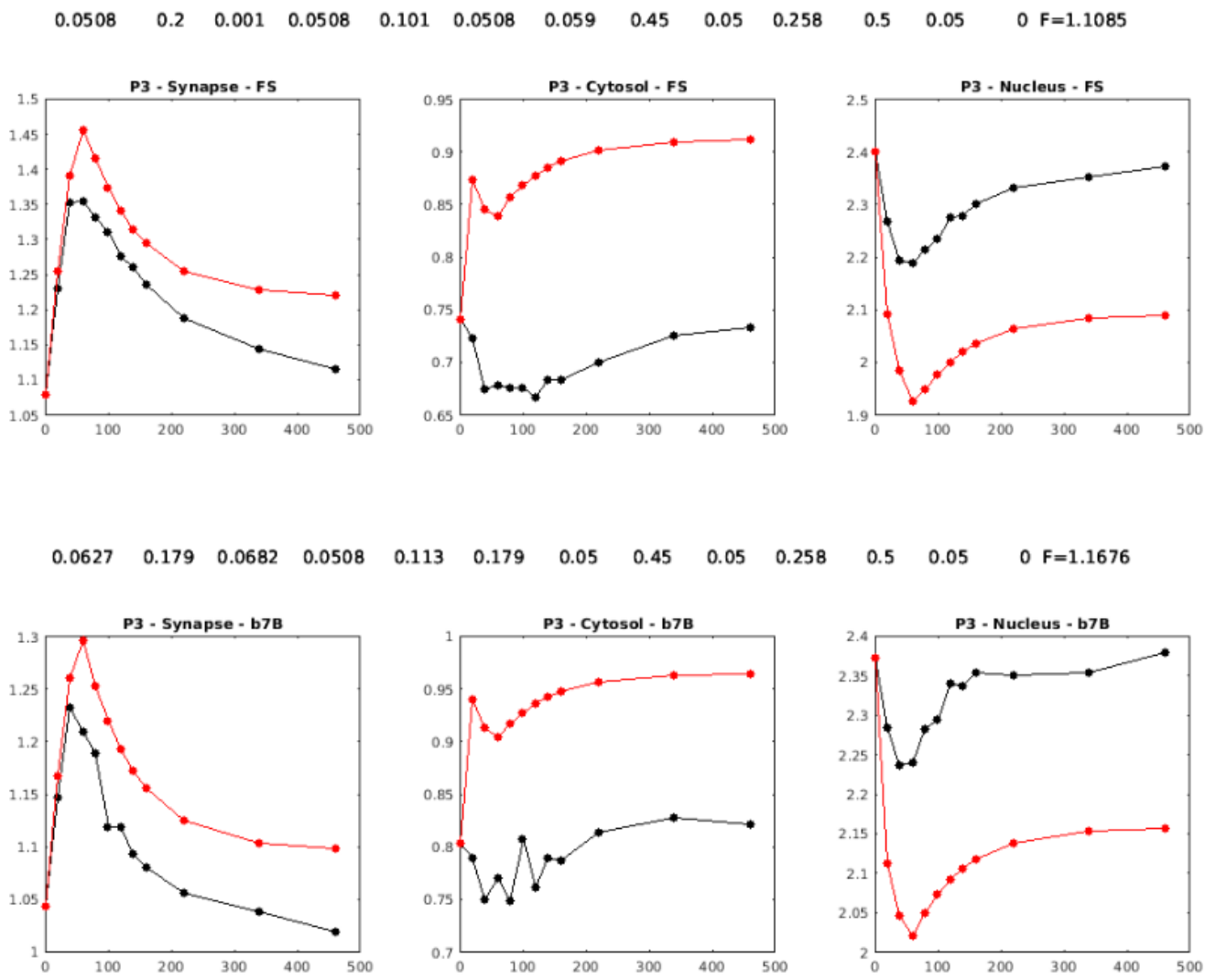


Figure 4-3: Protein 3, Cofilin. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

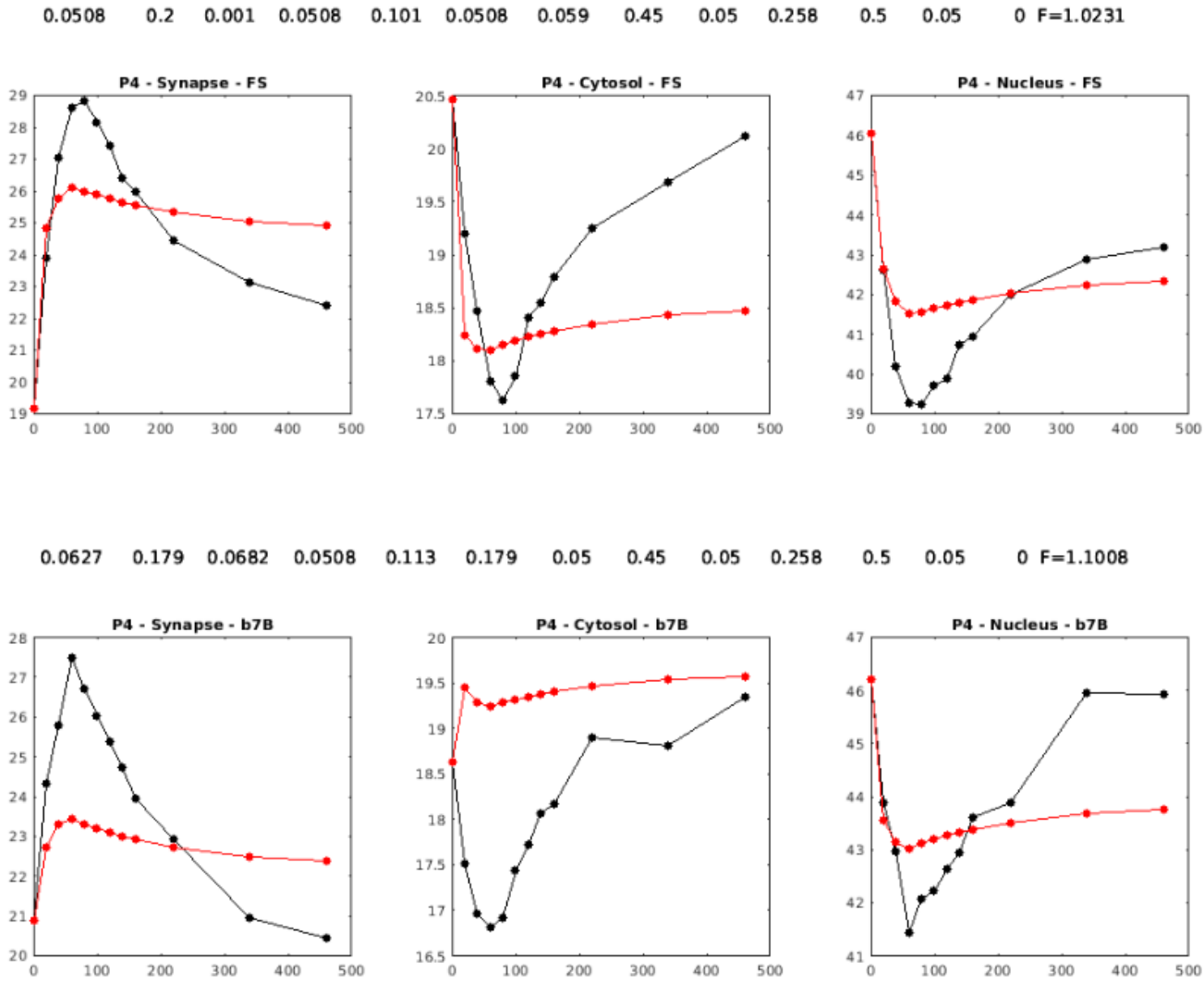


Figure 4-4: Protein 4, Coronin1A. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

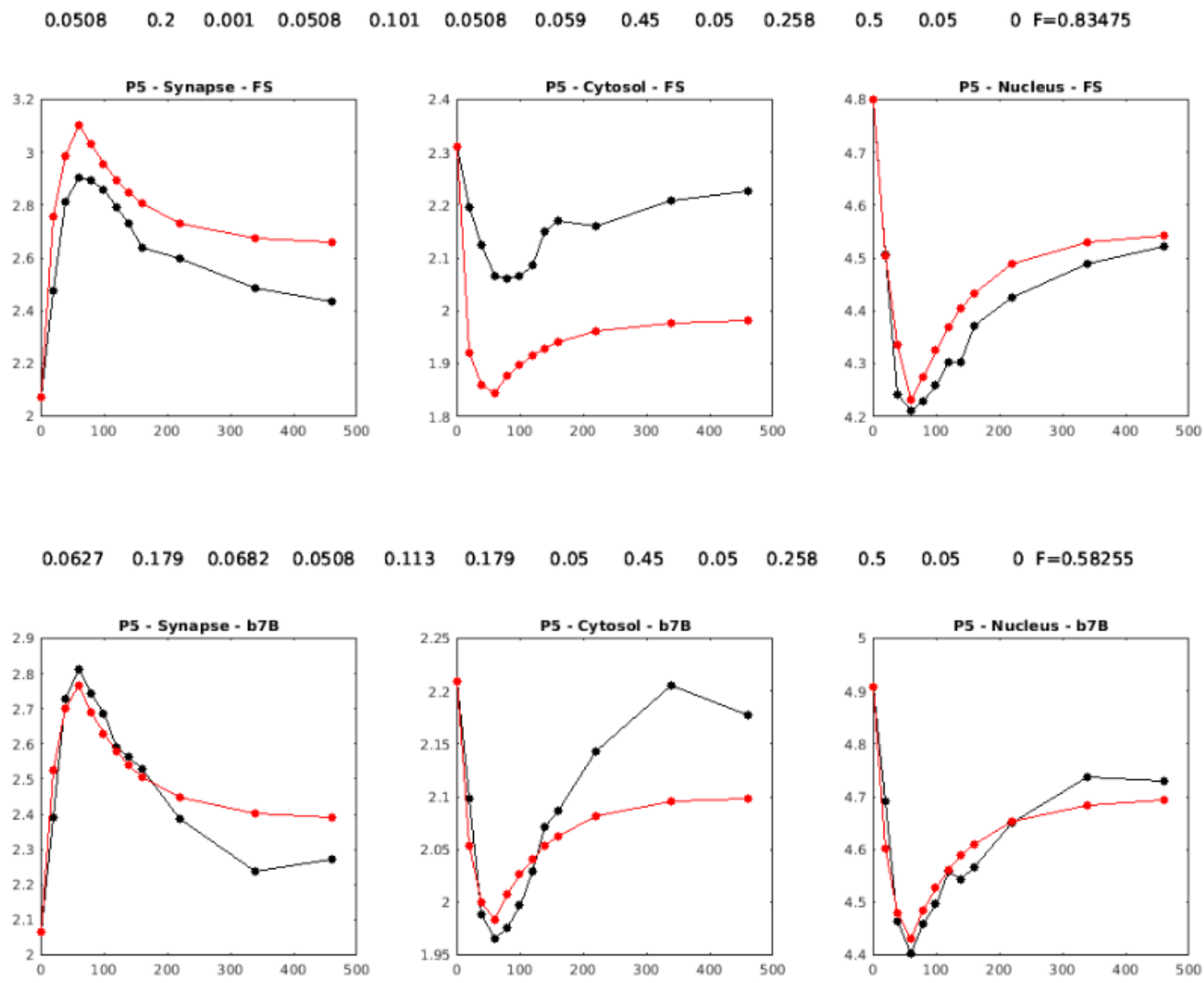


Figure 4-5: Protein 5, CPalpha1. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

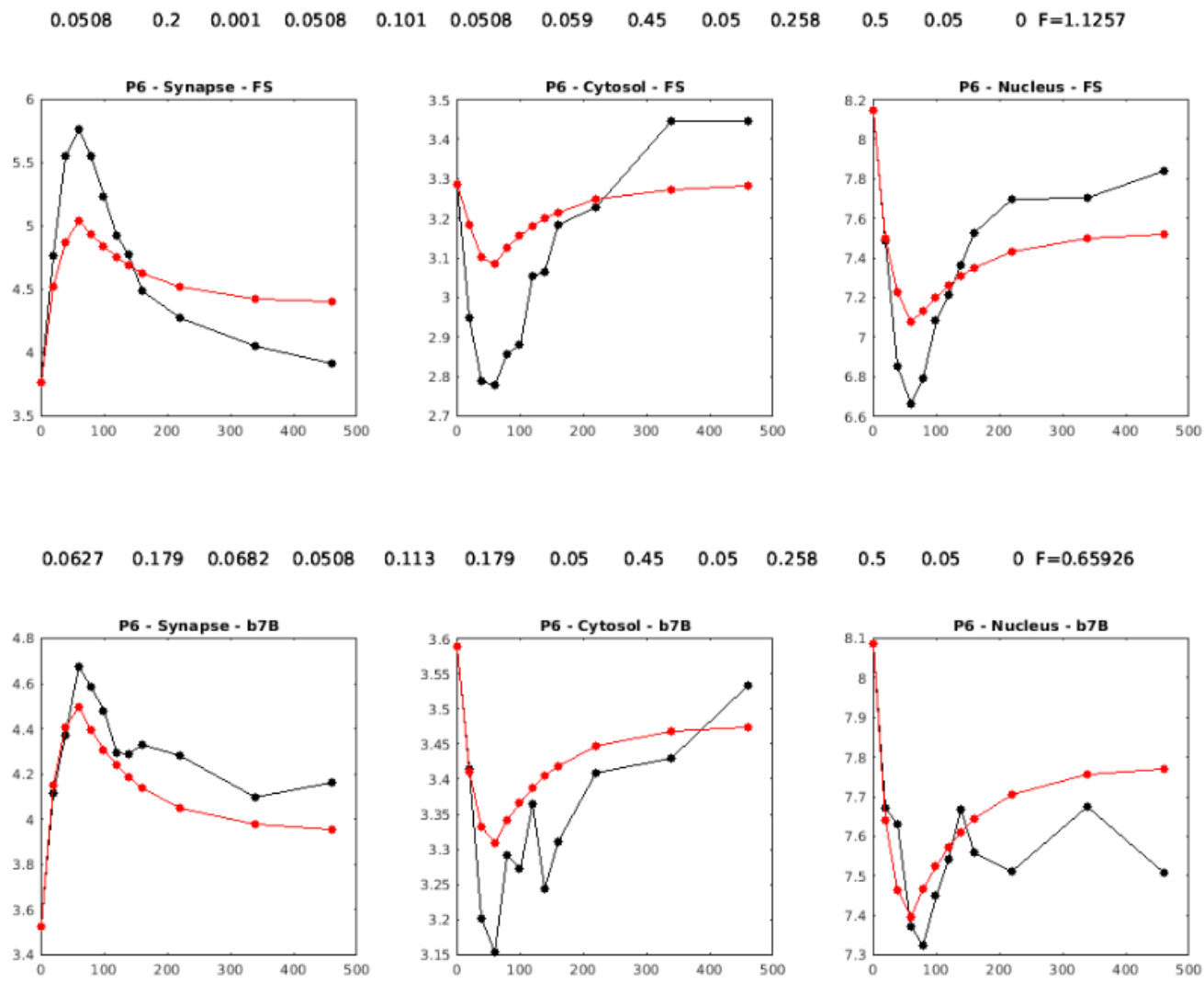


Figure 4-6: Protein 6, HS1. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

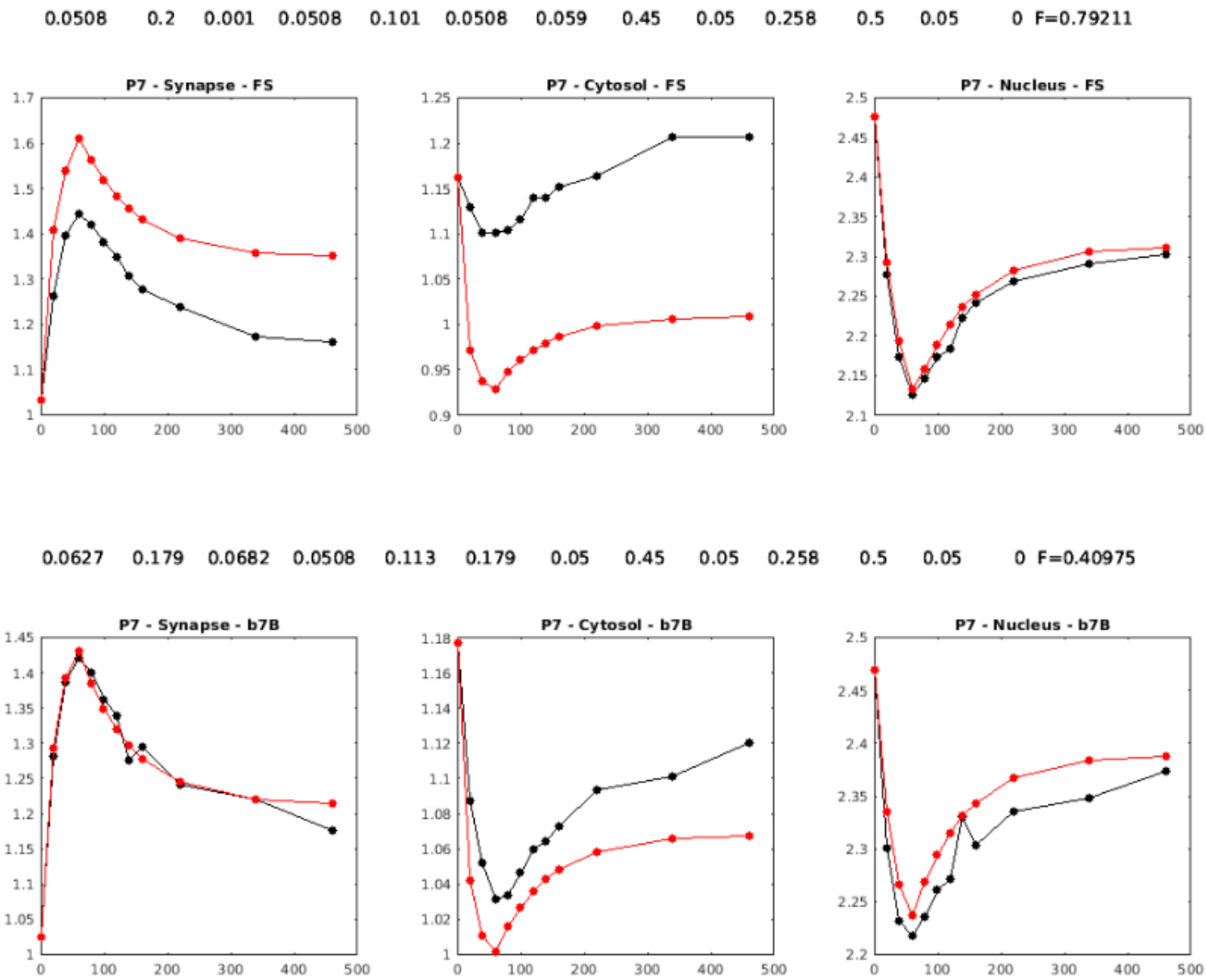


Figure 4-7: Protein 7, WASP. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

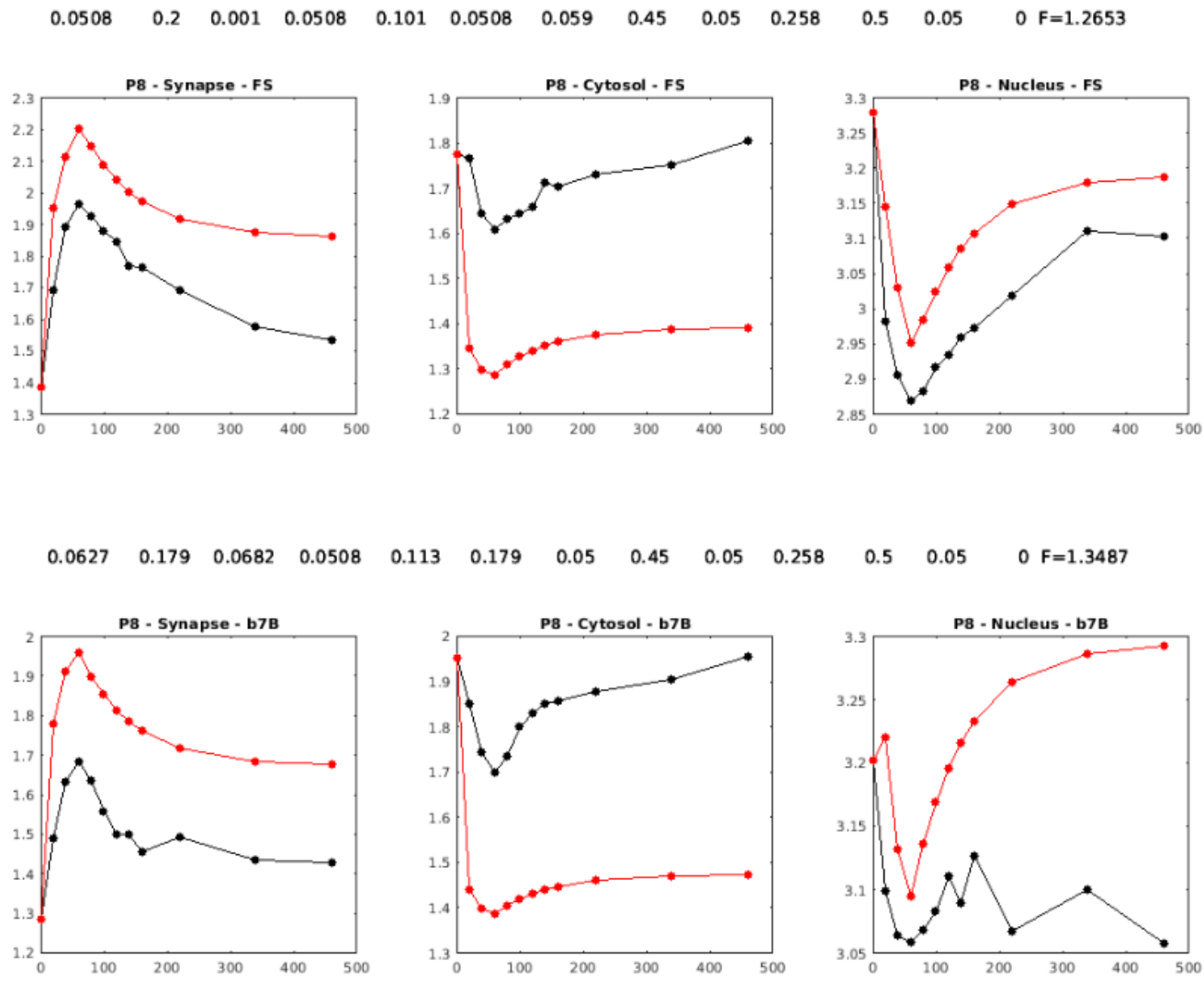


Figure 4-8: Protein 8, WAVE2. The parameter sets were shared among all proteins during optimization. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

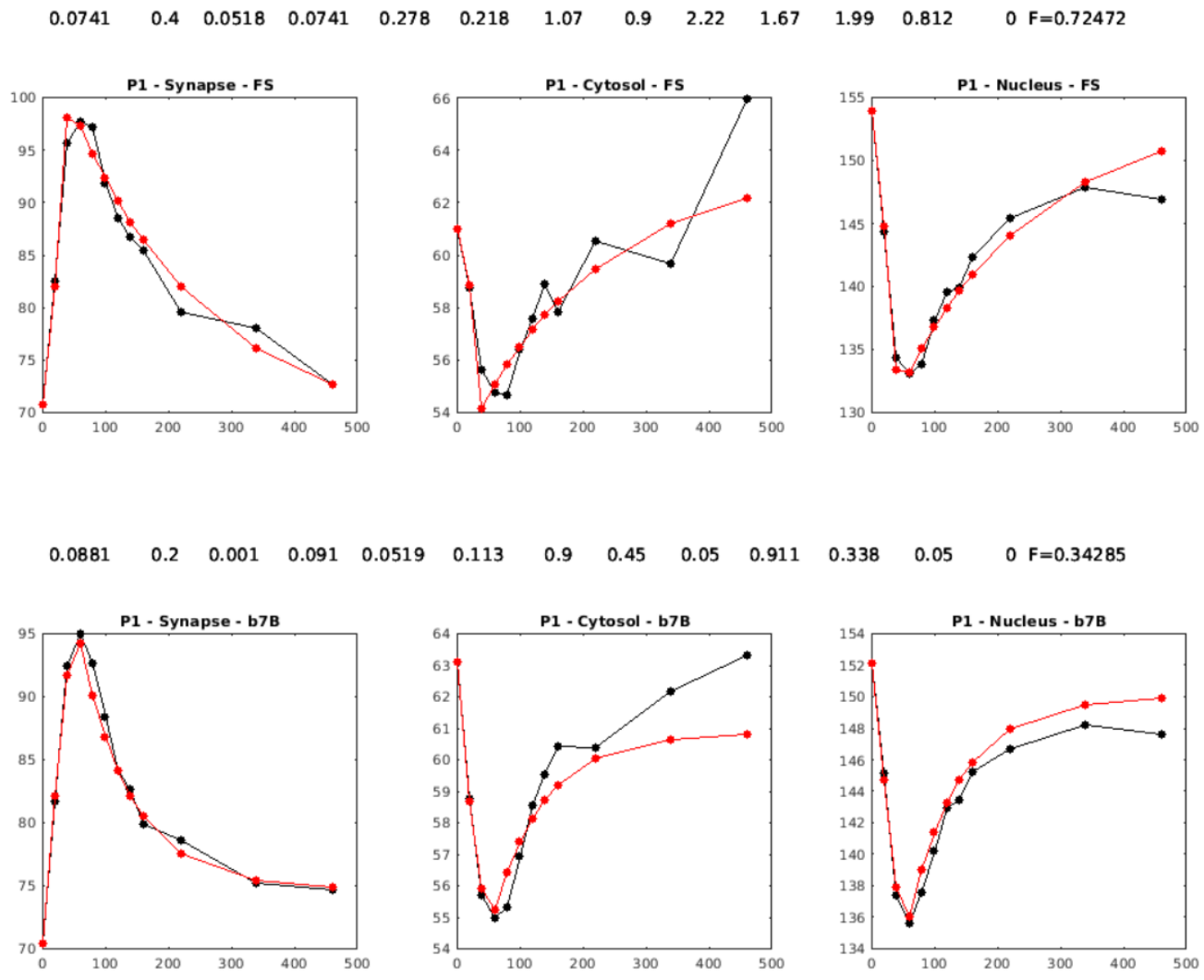


Figure 4-9: Protein 1, Actin. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}, k_{deactivate}$,$k_{on}, k_{off}$,$f_{on}, f_{off}$,$h_{decay}$), and the optimal objective value are displayed for each condition.

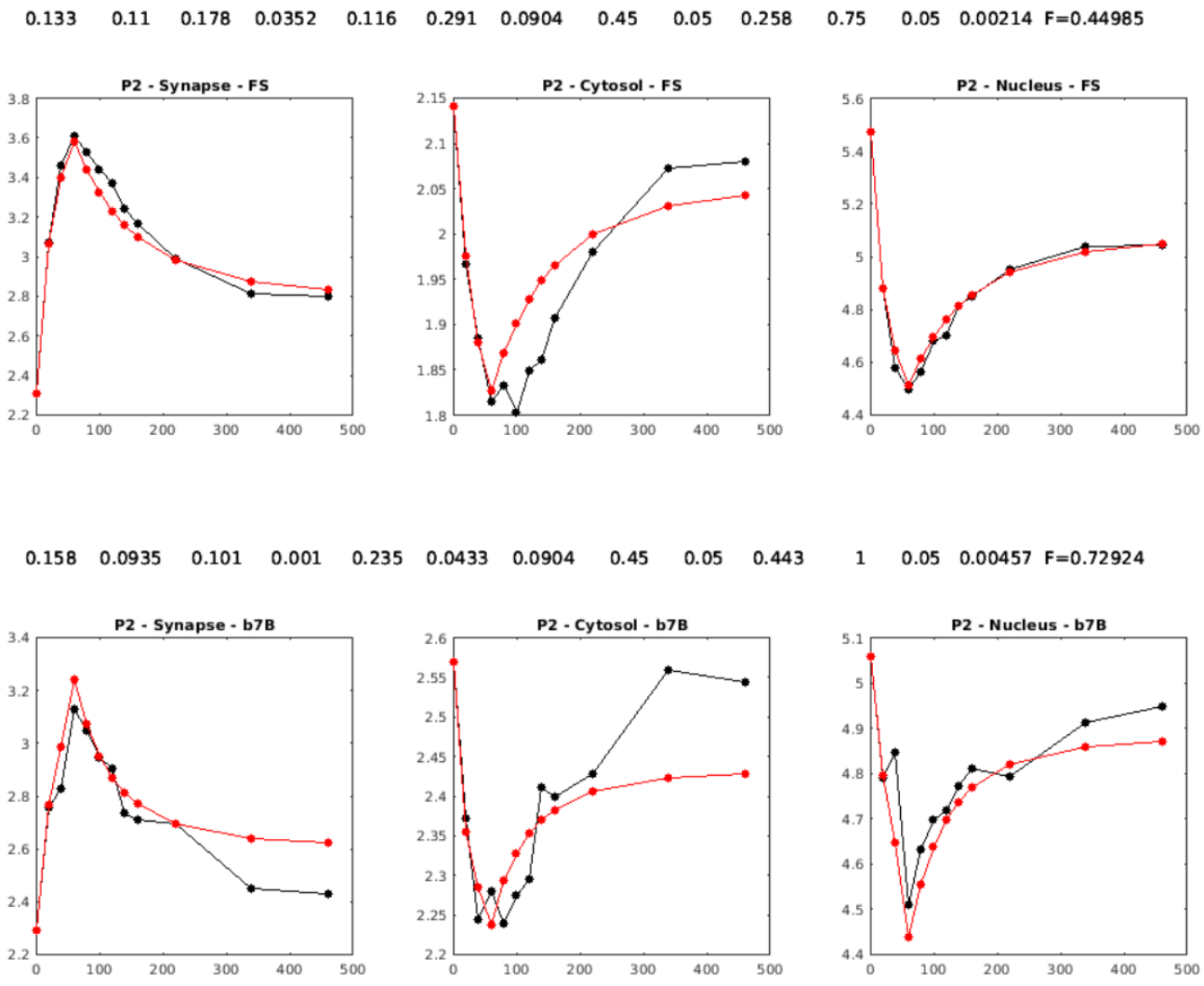


Figure 4-10: Protein 2, ARP3. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

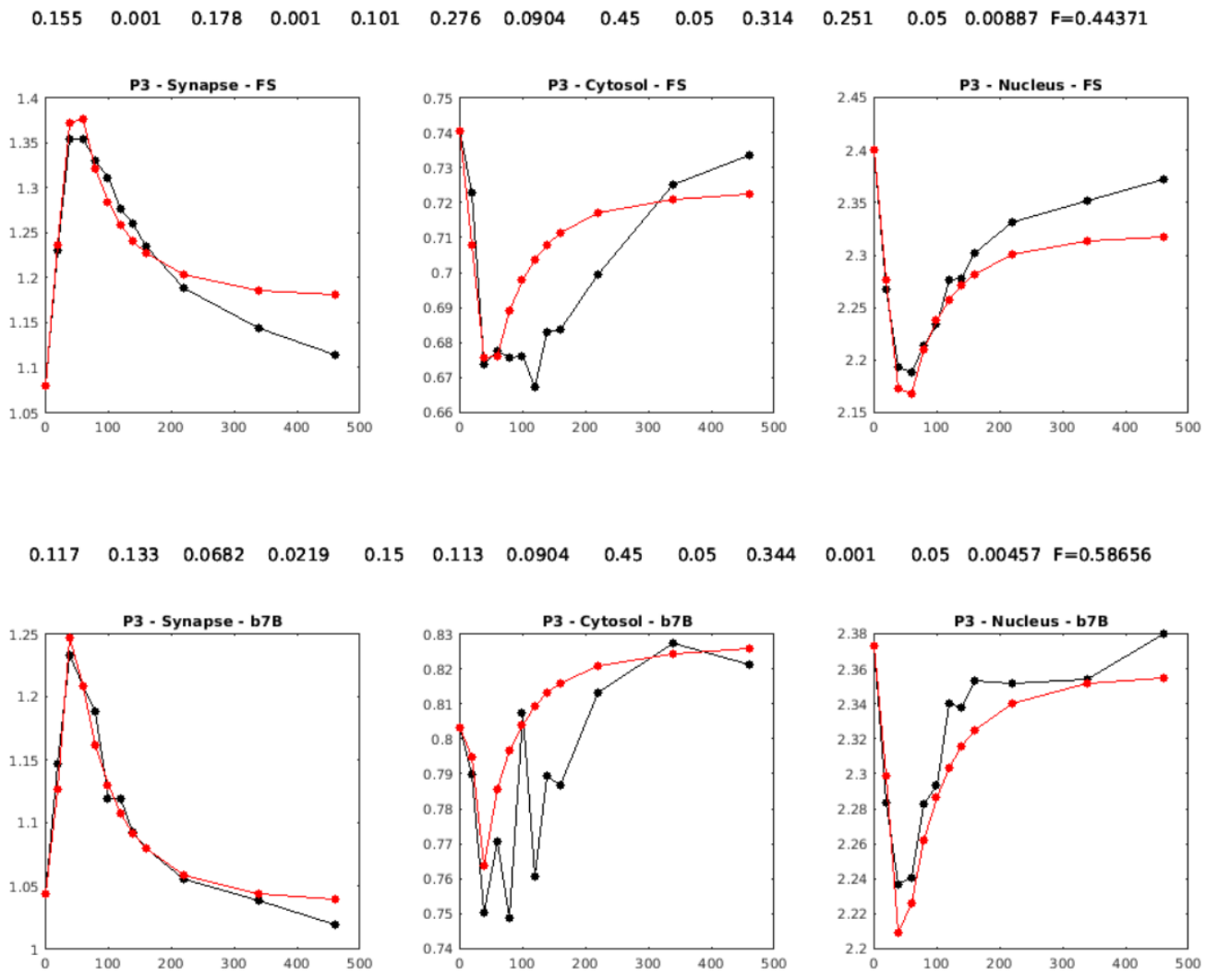


Figure 4-11: Protein 3, Cofilin. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

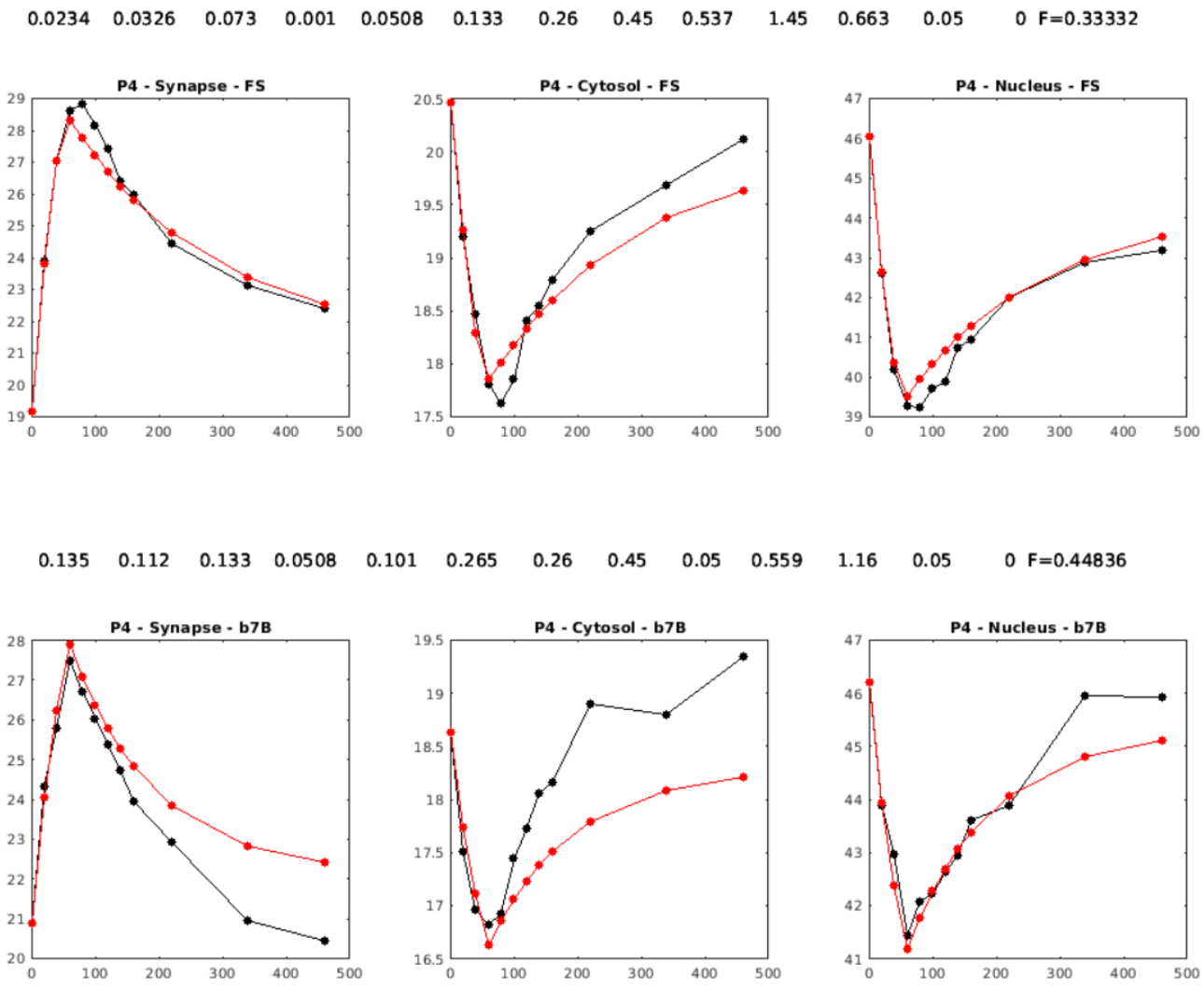


Figure 4-12: Protein 4, Coronin1A. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}, k_{deactivate}$,$k_{on}, k_{off}$,$f_{on}, f_{off}$,$h_{decay}$), and the optimal objective value are displayed for each condition.

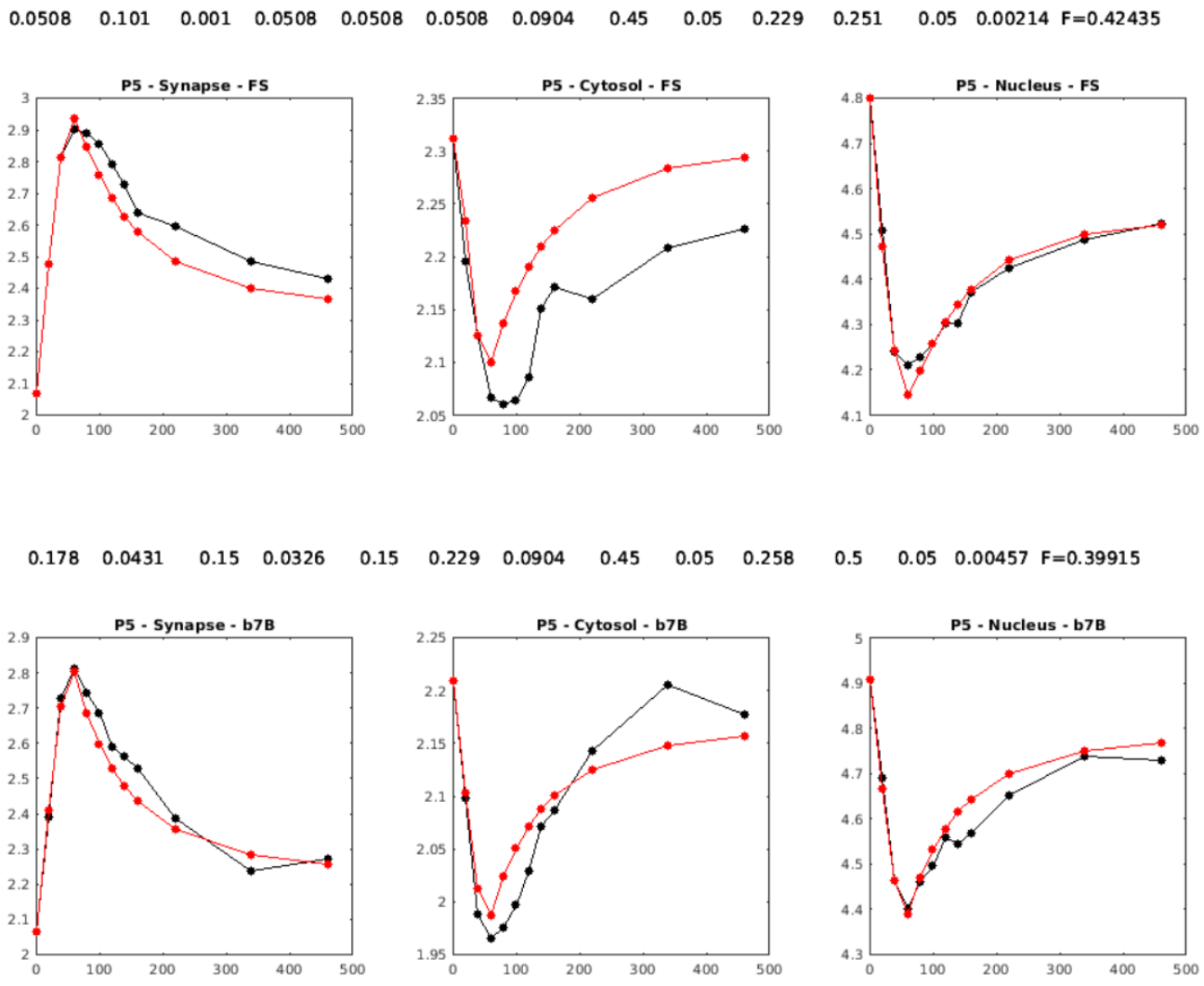


Figure 4-13: Protein 5, CPalpha1. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

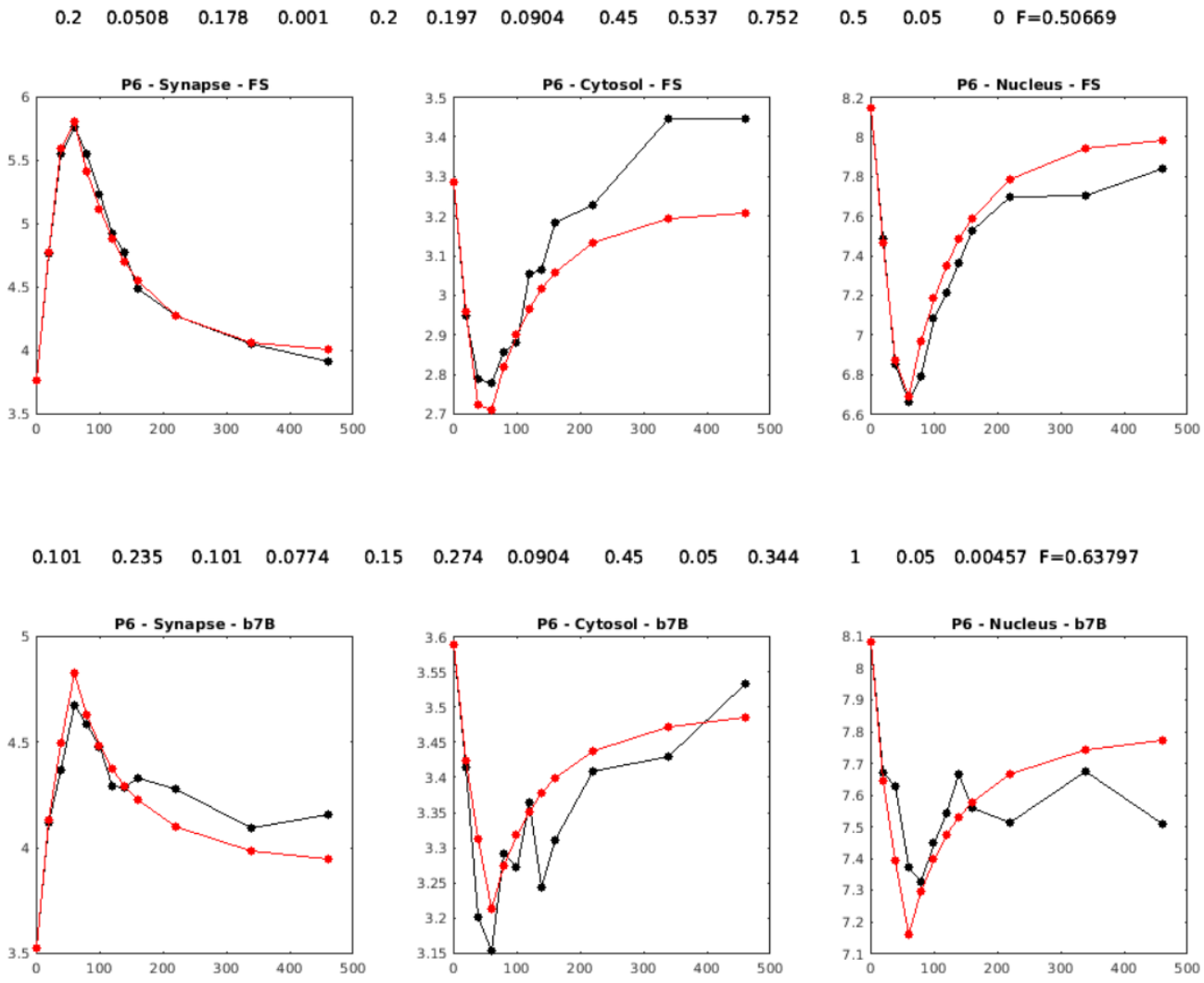


Figure 4-14: Protein 6, HS1. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

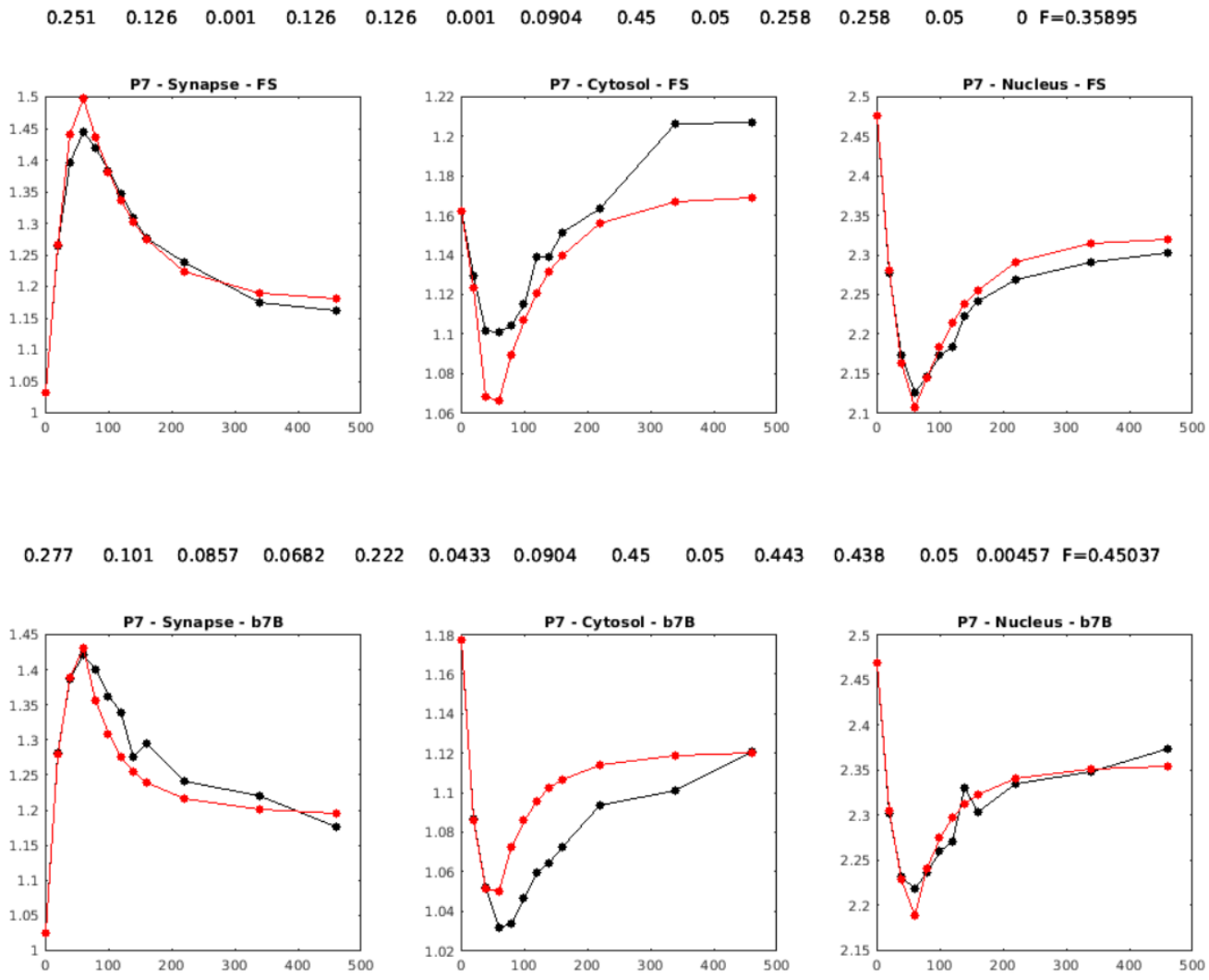


Figure 4-15: Protein 7, WASP. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, $(k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}, k_{activate}, k_{deactivate}, k_{on}, k_{off}, f_{on}, f_{off}, h_{decay})$, and the optimal objective value are displayed for each condition.

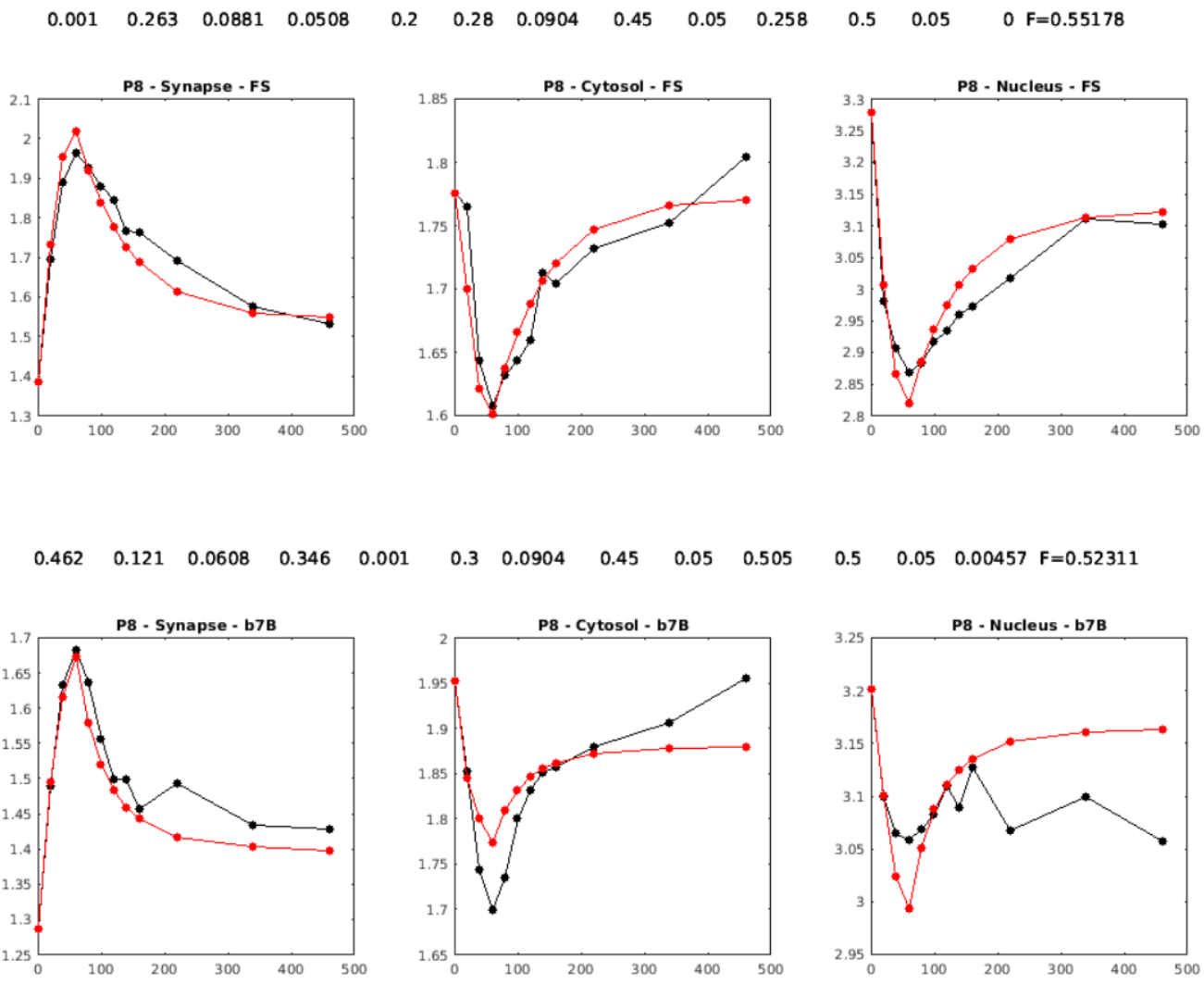


Figure 4-16: Protein 8, WAVE2. The parameter set was optimized independently from all other proteins. The top and bottom rows correspond to the Full Stimulus and B7 Blockade conditions, respectively. Concentrations are expressed in μM. The optimal parameter set, ($k_{SC}, k_{CS}, k_{SN}, k_{NS}, k_{NC}, k_{CN}$,$k_{activate}$, $k_{deactivate}$,k_{on}, k_{off},f_{on}, f_{off},h_{decay}), and the optimal objective value are displayed for each condition.

4.4 Discussion

Our kinetic model allows each protein's parameters to vary independently. The basic strategy was to start with "small" parameters and only increase them if necessary to fit the data. See Table 4.1 for a description of the initial grid. The region transfer rates began in the range [0.001,0.2] and the algorithm was able to fit all proteins without any exceeding 0.3. The TCR activation rate ($k_{activate}$, parameter 7 of 13) began in the range [0.0904, 0.6]. The models for Actin (protein 1) and Coronin1A (protein 4) could not be fit without the TCR activation rate reaching a certain minimuim threshold. Actin required roughly an order of magnitude larger rate (0.9) and Coronin1A a somewhat smaller increase (0.26) over the default value for all other proteins of 0.09. See Figure 4-17. Our hypothesis for Actin is that, because it can aggregate (i.e. self-interactions which are not an explicit part of our model), TCR binding is acting as a surrogate. A biological interpretation for Coronin1A is still being considered.

With caveats, the model is not sensitive to changes in transfer rate parameters if their ratios remain constant. In Figure 4-18, we plot for protein 1 how the objective value changes as we vary the transfer rates while maintaining a constant ratio among them. Each subfigure represents a set of points in which all six transfer rates k^i are first scaled by $\frac{1}{k^1}$ (to ensure a constant ratio) and then multiplied by a magnitude factor. This magnitude factor is displayed on the x-axis. The original point on which each data set is based is displayed in green. Below a magnitude of roughly 10^{-2}, the model becomes increasingly sensitive. Figure 4-19 displays a log scaling on the x-axis.

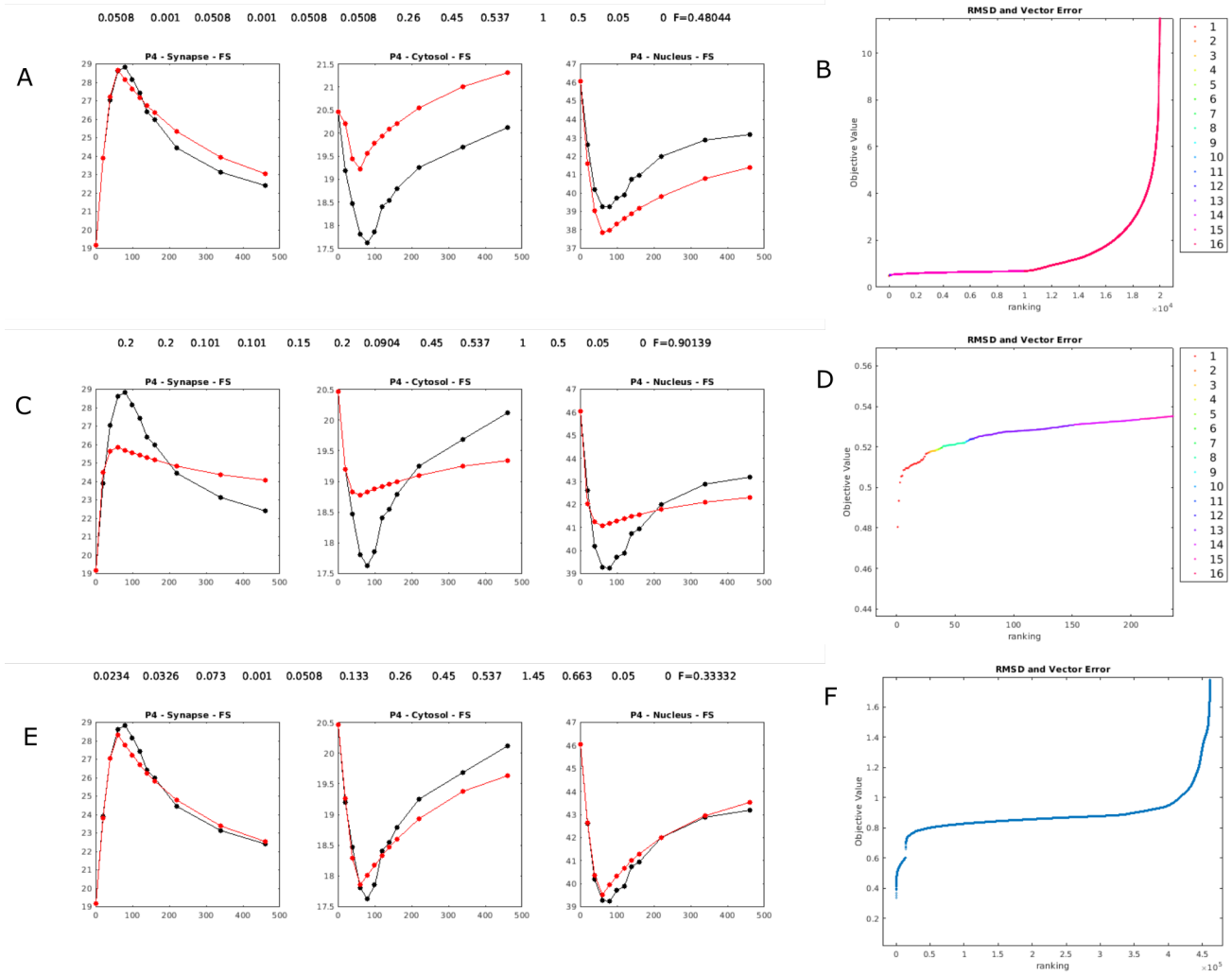


Figure 4-17: Protein 4, Coronin1A. Concentrations are expressed in μM. (A) The lowest objective point in the initial grid. (B) The best 10,000 objective values plus 10,000 logarithmically spaced points from the remaining (sorted) 39,052,500 points in the grid. This plot shows the full range of objective values seen in the initial grid. (C) The best point in the initial grid with a "low" TCR activation rate of 0.0904. This low rate is compatible with good fits for most other proteins, but not Coronin1A (either condition) nor Actin (either condition). (D) Zoomed in plot showing the top $\sim$ 300 points from (B) and clusters 1 to 15. (E) The best point after a round of optimization. (F) Low objective points from the "local grid" and the "exploratory grid", respectively.

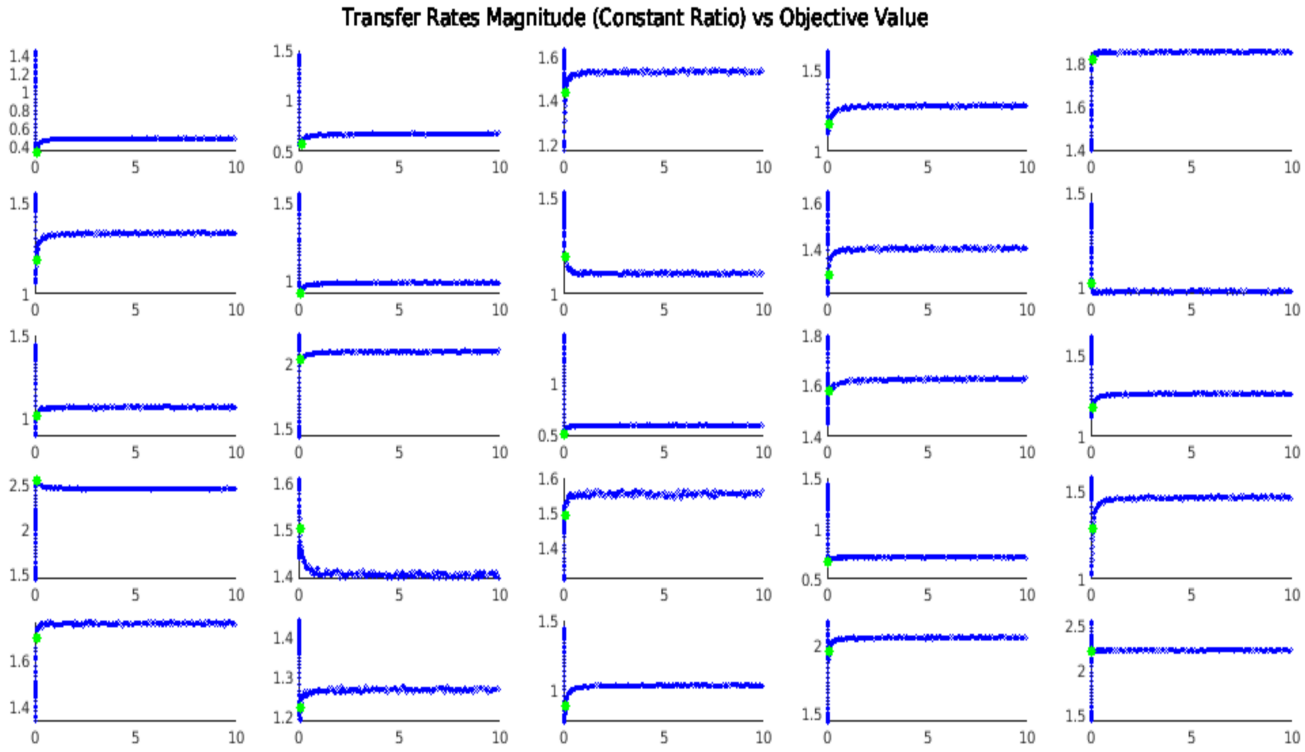


Figure 4-18: Sensitivity analysis for Protein 1, b7B condition. Best scoring point from optimization (top left subplot) and 24 points sampled from within 0.2 logs surrounding the best point (remaining subplots). For each subplot, shown in green is the objective value. The blue points show how the objective value changes as the transfer rates change such that their magnitudes increase by a common factor (given on the x-axis) but their ratios remain the same. Horizontal lines indicate no change in the objective as the transfer rates change, so long as their ratios remain the same.

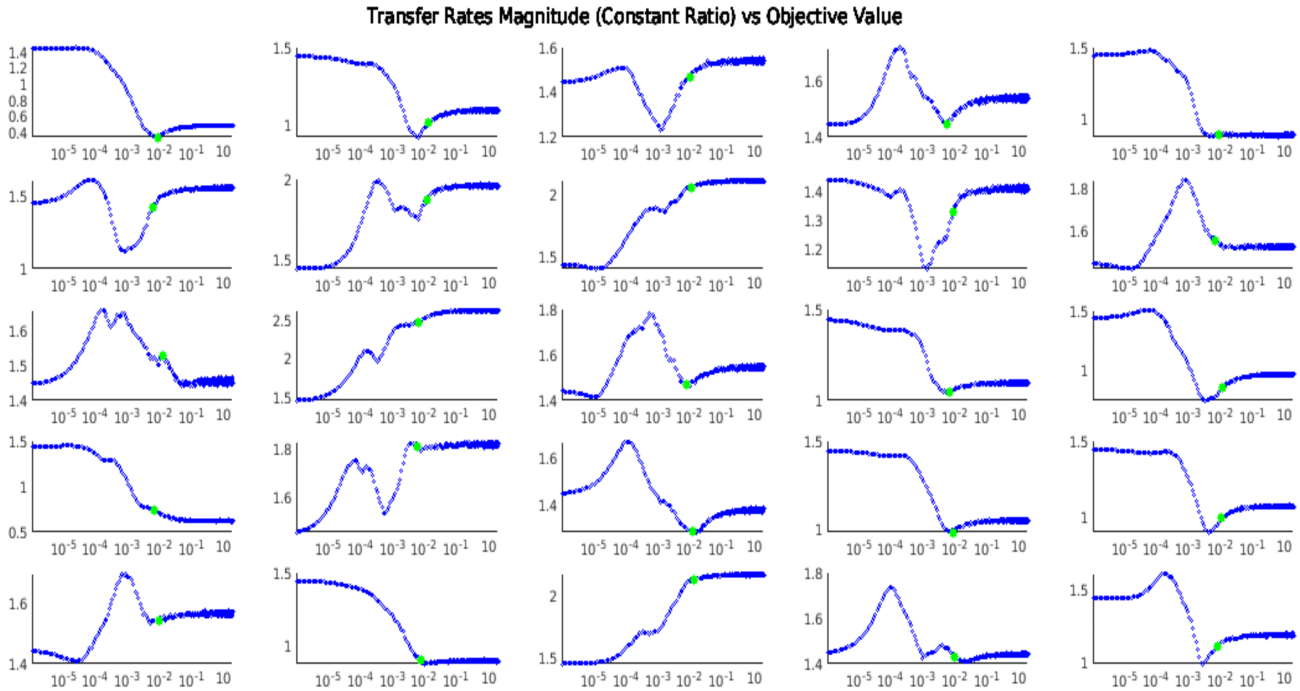


Figure 4-19: Sensitivity analysis for Protein 1, b7B condition. X-axis is log scaled. This figure is otherwise equivalent to 4-18. This figure indicates that when the transfer rates are very small, the statement that *only transfer rate ratios matter* is false. The actual magnitudes are important as these rates approach zero.

Sensitivity Analysis

Our parameter optimization assumes the experimental data, and therefore objective function evaluations, are noise free. Our sensitivity analysis therefore begins with the objective function itself. In future work, we will focus on other model observables, e.g., the widths of the synapse spikes and associated cytosol/nucleus inverse spikes. Because the objective function is nonconvex, we expect linear methods to have highest applicability in relatively small neighborhoods of the parameter space. The primary goal is to summarize how the ability of the differential equation model to explain the experimental data changes as the parameters change. Our general procedure is as follows. First, we construct a latin hypercube grid in log space surrounding the best point found using the optimization pipeline. The number of samples, N_s depends on search space size (n_{logs}). $N_s = 500 * (10^{3+n_{logs}})$. Thus for n_{logs} from 10^{-3} to 3, we have N_s from 5e5 to 5e8. Each of these points is then translated to linear space and their objective functions evaluated. For each data set, i.e., for increasing search space size, we determine the correlation between the true objective function and a reconstruction generated as a linear combination of the linear regression gradient with the input points: $F(\mathbf{x}) = a * x_1 + b * x_2 + ... + m * x_{13}$ where $[a, b, ..., m]^T = Grad(F)$. In this way we illustrate the extent to which the objective function varies linearly around the optimal point. See Figures 4-20 and 4-21 for the Full Stimulus and b7 Blockade conditions, respectively. In each figure, we plot the search space size first on a linear scale and then on a logarithmic scale. It is interesting to note the splitting behavior for certain proteins below roughly $\sim 10^{-2}$ logs search space size. One possible explanation is that, instead of a single 12d hyperplane capturing variation in the data in the neighborhood of the optimal point, there is a lower dimensional manifold capturing variation and many other directions characterized by high noise. Focusing on the upper points in the split, the explanation seems to be that the objective surface is locally linear nearby the optimal point. As we leave the neighborhood of the optimal point (roughly $\sim 10^{-1}$) and large scale trends emerge, we see the correlation slowly rise.

We also investigated the correlation of the objective function with its projection onto each PCA axis. No clear inference is drawn from this analyis but we show the result for protein 1 (Actin, Figure 4-22).

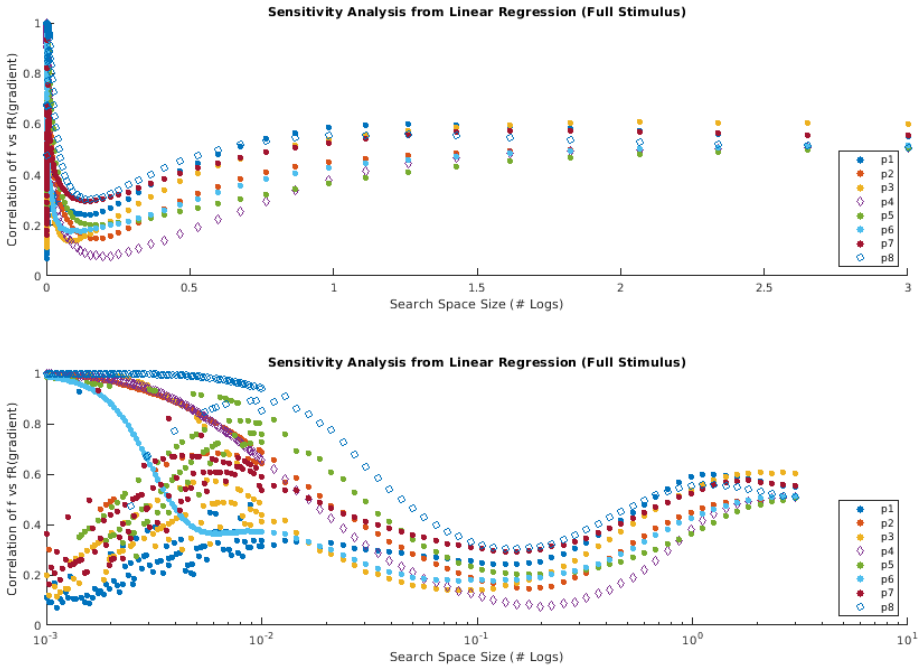


Figure 4-20: Sensitivity analysis for all proteins in the Full Stimulus condition. (y-axis) A measure of linearity of the objective function space vs (x-axis) the size of the search space in terms of the number of logs above and below each parameter in the best point found after optimization.

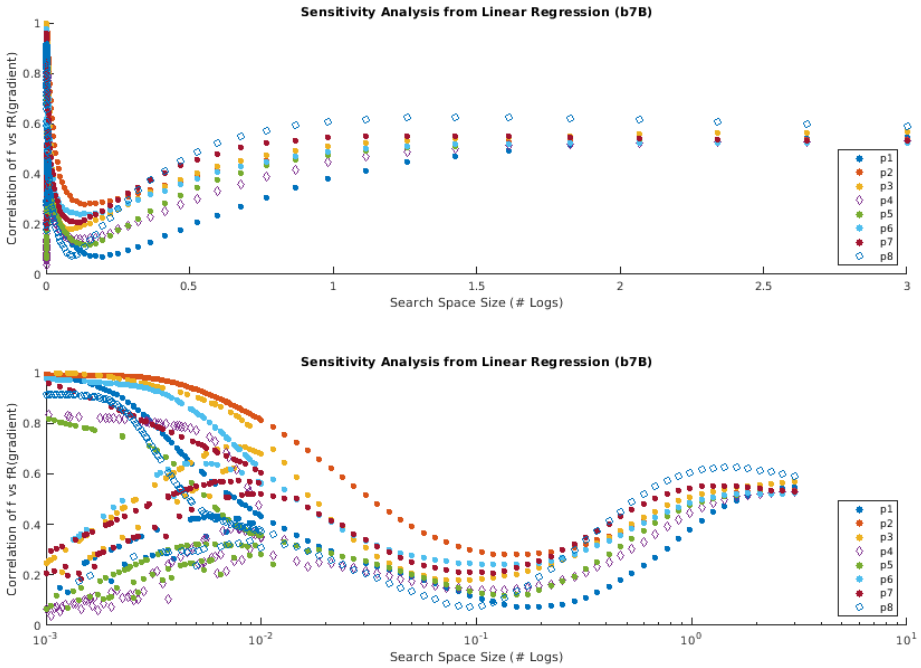


Figure 4-21: Sensitivity analysis for all proteins in the b7B Condition.

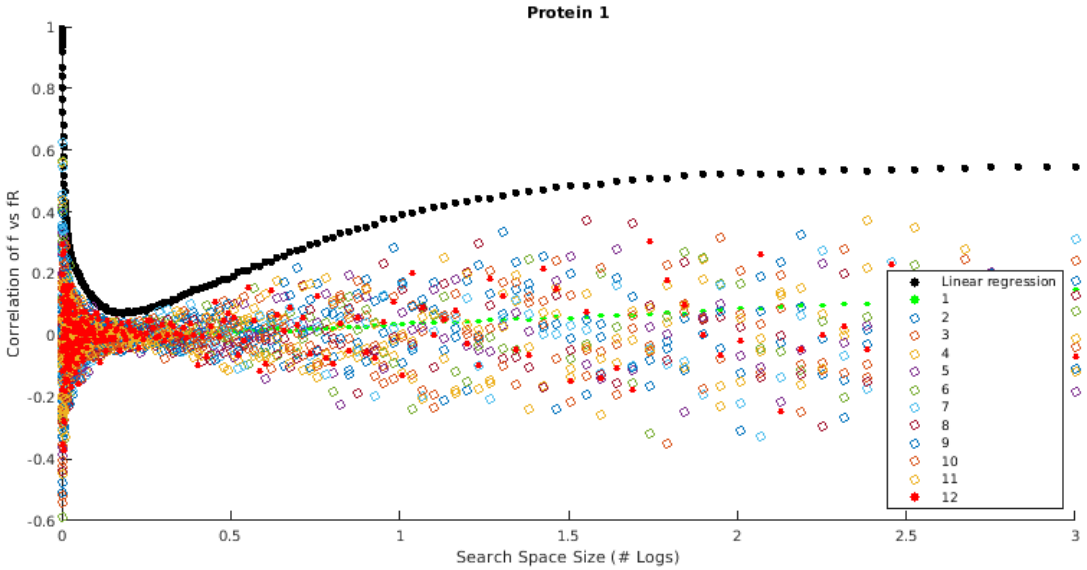


Figure 4-22: Sensitivity analysis for Protein 1, b7B condition. For increasing search space size, i.e., maximum distance along each dimension in log space from the best scoring parameter vector, the correlation is plotted between the true objective value and a number of reconstructions. Each reconstruction is generated as a linear combination of parameter vectors and a corresponding vector (e.g., linear regression gradient or PCA component). PCA was performed without first centering the data. The plot shows that the objective space is well approximated by a linear model only very close to the optimal point. The increasing correlation beginning around 0.25 logs on the x-axis is due to the increasing objective value outside the neighborhood of the optimal point.

Chapter 5

Conclusions

In this thesis we have worked to develop computational methods for better understanding aspects of molecular self-assembly, general reaction-diffusion chemistry and immune cell signaling.

We have developed a novel method for efficient Bayesian parameter inference from rule-based models of molecular self-assembly and demonstrated it for fitting stochastic and deterministic models of viral self-assembly to simulated SAXS or SANS data. Our results show that for stochastic systems of low to moderate dimension, treating the objective function as being separately generated by multiple Gaussian processes can be an effective way to discover its structure. When placed within a Bayesian optimization framework, this translates to efficiently discovering locally optimal regions of the parameter space.

We next presented a novel event-based method for simulating reaction diffusion systems in continuous space and in the presence of planar or curved boundaries. As in the Gillespie algorithm and related methods, we sample bimolecular reaction waiting times by utilizing propensity functions. However, with the introduction of 3d space, the reaction propensities now depend explicitly on the time reactants diffuse, allowing them to encounter one another. The result is that we integrate the propensity function of each reactant pair in order to determine whether (and when) a reaction is possible in a given duration. While the method is inspired by ideas from GFRD and eGFRD, our method for sampling reaction locations given the waiting time is, to our knowledge, novel relative to other spatial simulation methods. For point particles, we rely on two assumptions: (1) that reactions must happen in the region both reactants' diffusion spheres overlap and (2) that the probability distributions characterizing the possible distances either

reactant has traveled are Gaussian and independent. This implies there are rings of equiprobable points at constant distance from the Gaussian means of the reactants and suggests a method to sample such a ring: first, sample a distance, r_A from one reactant, and then sample the angle with respect to the axis connecting the means given r_A. Each ring is uniquely determined by this distance and angle, and the reaction location can then be selected uniformly at random from on the ring. In the case of molecules with finite size, Green's functions governing radial separation must be used for wait time sampling, however the same ring sampling procedure applies to reaction locations.

We compared our method with its most relevant competitor, eGFRD, on the modified Michaelis-Menten benchmark model described in Chew et al.[49]. The dynamics displayed by eGFRD, Spatiocyte (implementing a microscopic lattice method), and Smoldyn are quantitatively similar. DESSA-CS shows a substantial improvement over the run time of eGFRD, achieving run times generally comparable to the discrete-time alternatives Smoldyn and Spaciocyte MLM.

In general, both the finite and point particle representations run more efficiently when each molecule is allowed to diffuse farther outside the boundaries before wait times are sampled. In the case of a cubic volume in which the sum of the cube length and maximal distance a particle may diffuse beyond the boundary remains constant, the overall reaction rate changes slowly as the diffusion distance increases. In future work we will precisely characterize the error introduced with changing these lengths.

Finally, we investigated T-cell-Actin dynamics in the presence of antigen presenting cells through the lens of a simple kinetic binding model. We were able to fit the model and learned that Actin and the regulator Coronin1A show qualitatively different dynamics than the remaining six proteins. This project is an on-going collaboration and will continue to develop more biologically realistic kinetic models, perhaps incorporating a priori knowledge gained from genetic assays, and possibly using simulations with more spatial resolution. One important future direction will involve taking into account the sources of error noise present in spatiotemporal concentration data, both at the voxel level and at the level of the inferred cell regions (synapse, cytosol and nucleus). Currently, we treat these regions as noise free averages over subsets of voxels and therefore the objective function evaluations are also noise free.

5.1 Future Directions and Some Speculation

Future work for the self-assembly inference pipeline in the short term should prioritize obtaining real world time resolved data. On the computational side, assuming spatial simulations are available, progress would require revisiting the assumptions made when translating assembly simulations into *in silico* experimental data. Currently we extrapolate the single subunit scattering, obtained from the program CRYSOL, to the total system scattering by making a dilute solute assumption. That extrapolation depends on properly computing structure factors which depend on the spatial distributions of the subunits. In the long term, progress will involve the development of more flexible computational models capable of representing more than the simple binding kinetics of rigid structures. Capsid oligomers likely undergo deformations and there are important interactions with scaffolding proteins and genetic material.

DESSA-CS as presented leaves several avenues for extension and improvement in future work. It is able to achieve its comparatively high run time efficiency by exploiting the fact that, under certain assumptions, wait time sampling can be described by a deterministic part applicable in many circumstances, and a stochastic part specific to each reactant pair. We can therefore perform much of the expensive deterministic computations once, independently of each simulation run. Those assumptions include isotropic diffusion as the primary method of transport, and that reactions between distinct pairs of molecules are described by time-inhomogenous Poisson processes with mean parameter equal to the integrated propensity (this implies exponentially distributed waiting times). These are reasonable assumptions, yet both may be relaxed in future work. Numerically integrating these reaction propensities when considering new reactions at every step of the simulation can lead to the same computations being performed thousands or millions of times. Only the sampling of the exponentially distributed random numbers must be performed for all potential bimolecular reactions. The overall accuracy of the method is dependent on the resolution of the pre-computed integrated propensity curves.

In the longer term, simulation based exploration of self-assembly in more realistic *in vivo* conditions will require methods able to efficiently handle more complex dynamics including rotational diffusion, molecular crowding and anomylous diffusion, as well as flexible intermediate structures capable of structural changes post-aggregation (e.g., relaxation into lower energy

conformations). Maintaining efficiency will be a priority because important reaction pathway features of self-assembly typically span many time scales.

Another long term goal must be to invent methods that can benefit from the recent progress in machine learning, and deep learning in particular. As far as we are aware, no simulation methods make use of artificial intelligence in any significant respect, instead they are based purely on mechanistic physical models. The field may be approaching the limit of potential improvements in this kind of method design. The current paradigm is based on a strategy of starting with idealized assumptions and slowly adding particular kinds of complexity. For example, the Gillespie algorithm assumes molecules diffuse like particles in an ideal gas. The system is dilute and well mixed allowing the algorithm to ignore spatial effects and focus on the evolution of species level populations. eGFRD (and all spatial methods excepting molecular dynamics) assumes molecules obey a diffusion equation, an assumption only valid for sufficiently large time durations, yet its sampling is based on solving the diffusion equation within protective domains possibly not much larger than a molecular radius. The microscopic lattice method [49] assumes space is discrete and derives reaction probabilities for molecules occupying the same voxel from the same diffuse equation framework. Our work on DESSA-CS was yet another attempt to reframe this problem and while it shows promise in terms of efficiency, it is not an exact method according to those same assumptions. It may be that in order to make significant gains over the current methods in terms of the complexity of the dynamics that can be simulated accurately and efficiently, the starting point must be statistical observations rather than simplifying assumptions. Maybe one could train an algorithm to predict integrated reaction propensities as a function of molecular features and the local environment. Maybe one could learn local rules applicable not only to subunits, but to larger intermediate structures and use them for prediction in real time during simulation. Maybe one could maintain the assumptions used to construct the Gillespie algorithm, which samples trajectories from the Chemical Master Equation, but instead use ML methods to directly evolve the probabilities obeying this equation.

Bibliography

[1] Barak Akabayov, Sabine R Akabayov, Seung-Joo Lee, Gerhard Wagner, and Charles C Richardson. Impact of macromolecular crowding on dna replication. Nature communications, 4:1615, 2013.

[2] Mitchel Alioscha-Perez, Carine Benadiba, Katty Goossens, Sandor Kasas, Giovanni Dietler, Ronnie Willaert, and Hichem Sahli. A robust actin filaments image analysis framework. PLoS Comput Biol, 12(8):e1005063, 2016.

[3] Patrick Amar and Loïc Paulevé. Hsim: A hybrid stochastic simulation system for systems biology. Electronic Notes in Theoretical Computer Science, 313:3–21, 2015.

[4] David F Anderson. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. The Journal of chemical physics, 127(21):214107, 2007.

[5] David F Anderson. Incorporating postleap checks in tau-leaping. The Journal of chemical physics, 128(5):054103, 2008.

[6] Steven S Andrews, Nathan J Addy, Roger Brent, and Adam P Arkin. Detailed simulations of cell biology with smoldyn 2.1. PLoS Comput Biol, 6(3):e1000705, 2010.

[7] Steven S Andrews and Dennis Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. Physical biology, 1(3):137, 2004.

[8] Mihaela M Apetri, Rolf Harkes, Vinod Subramaniam, Gerard W Canters, Thomas Schmidt, and Thijs J Aartsma. Direct observation of α-synuclein amyloid aggregates in endocytic vesicles of neuroblastoma cells - licenced under cc by 4.0. PloS one, 11(4):e0153020, 2016.

[9] Nora Ausmees, Jeffrey R Kuhn, and Christine Jacobs-Wagner. The bacterial cytoskeleton: an intermediate filament-like function in cell shape. Cell, 115(6):705–713, 2003.

[10] Stephan Jan Bachmann, Marius Petitzon, and Bortolo Matteo Mognetti. Bond formation kinetics affects self-assembly directed by ligand–receptor interactions. Soft matter, 12(47):9585–9592, 2016.

[11] Yushi Bai, Quan Luo, and Junqiu Liu. Protein self-assembly via supramolecular strategies. Chemical Society Reviews, 45(10):2756–2767, 2016.

[12] JL Bailey, PJ Critser, C Whittington, JL Kuske, Mervin C Yoder, and SL Voytik-Harbin. Collagen oligomers modulate physical and biological properties of three-dimensional self-assembled matrices. Biopolymers, 95(2):77–93, 2011.

[13] Florence Baras and M Malek Mansour. Reaction-diffusion master equation: A comparison with microscopic simulations. Physical Review E, 54(6):6139, 1996.

[14] Johanna E Baschek, Heinrich CR Klein, and Ulrich S Schwarz. Stochastic dynamics of virus capsid formation: direct versus hierarchical self-assembly. BMC biophysics, 5(1):22, 2012.

[15] Daniel A Beard and Tamar Schlick. Computational modeling predicts the structure and dynamics of chromatin fiber. Structure, 9(2):105–114, 2001.

[16] Bonnie Berger, Peter W Shor, Lisa Tucker-Kellogg, and Jonathan King. Local rule-based theory of virus shell assembly. Proceedings of the National Academy of Sciences, 91(16):7732–7736, 1994.

[17] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb):281–305, 2012.

[18] Noah S Bieler, Tuomas PJ Knowles, Daan Frenkel, and Robert Vácha. Connecting macroscopic observables and microscopic assembly events in amyloid formation using coarse grained simulations. PLoS Comput Biol, 8(10):e1002692, 2012.

[19] Mirko Bischofberger, Ioan Iacovache, Daniel Boss, Felix Naef, F Gisou van der Goot, and Nacho Molina. Revealing assembly of a pore-forming complex using single-cell kinetic analysis and modeling. Biophysical journal, 110(7):1574–1581, 2016.

[20] Arne Bittig and Adelinde Uhrmacher. Ml-space: Hybrid spatial gillespie and particle simulation of multi-level rule-based models in cell biology. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016.

[21] Laurent Blanchoin, Kurt J Amann, Henry N Higgs, Jean-Baptiste Marchand, Donald A Kaiser, and Thomas D Pollard. Direct observation of dendritic actin filament networks nucleated by arp2/3 complex and wasp/scar proteins. Nature, 404(6781):1007–1011, 2000.

[22] 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.

[23] Michael L Blinov, James R Faeder, Byron Goldstein, and William S Hlavacek. A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. Biosystems, 83(2):136–151, 2006.

[24] Marvin A Boettcher, Heinrich CR Klein, and Ulrich S Schwarz. Role of dynamic capsomere supply for viral capsid self-assembly. Physical biology, 12(1):016014, 2015.

[25] Aarash Bordbar, Jonathan M Monk, Zachary A King, and Bernhard O Palsson. Constraint-based models predict metabolic and associated cellular functions. Nature Reviews Genetics, 15(2):107–120, 2014.

[26] Geuorgui K Bourov and Aniket Bhattacharya. The role of geometric constraints in amphiphilic self-assembly: A brownian dynamics study. The Journal of chemical physics, 119(17):9219–9225, 2003.

[27] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599, 2010.

[28] Rebecca J Burgess and Zhiguo Zhang. Histone chaperones in nucleosome assembly and human disease. Nature structural & molecular biology, 20(1):14–22, 2013.

[29] RE Buxbaum and SR Heidemann. A thermodynamic model for force integration and microtubule assembly during axonal elongation. Journal of theoretical biology, 134(3):379–390, 1988.

[30] By Awapuhi Lee (Own work) [CC BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0)], via Wikimedia Commons. 1x9j, 2015.

[31] By Biao He1, Quanshui Fan1, Fanli Yang, Tingsong Hu, Wei Qiu, Ye Feng, Zuosheng Li, Yingying Li, Fuqiang Zhang, Huancheng Guo, Xiaohuan Zou, and Changchun Tu [Public domain], via Wikimedia Commons. Hepatitis virus in long-fingered bats.

[32] By Doc. RNDr. Josef Reischig, CSc. (Author's archive) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons. Chromatin fibers (261 19) mitosis; xenopus egg.

[33] By Nephron (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons. Cerebral amyloid angiopathy -2b- amyloid beta - very high mag.

[34] Ruben D Cadena-Nava, Mauricio Comas-Garcia, Rees F Garmann, ALN Rao, Charles M Knobler, and William M Gelbart. Self-assembly of viral capsid protein and rna molecules of different sizes: requirement for a specific high protein/rna mass ratio. Journal of virology, 86(6):3318–3326, 2012.

[35] Marco Cammarata, Matteo Levantino, Friedrich Schotte, Philip A Anfinrud, Friederike Ewald, Jungkweon Choi, Antonio Cupane, Michael Wulff, and Hyotcherl Ihee. Tracking the structural dynamics of proteins in solution using time-resolved wide-angle x-ray scattering. Nature methods, 5(10):881–886, 2008.

[36] Mary A Canady, Hiro Tsuruta, and John E Johnson. Analysis of rapid, large-scale protein quaternary structural changes: time-resolved x-ray solution scattering of nudaurelia capensis ω virus (nωv) maturation. Journal of molecular biology, 311(4):803–814, 2001.

[37] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. The Journal of chemical physics, 122(1):014116, 2005.

[38] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Efficient step size selection for the tau-leaping simulation method. The Journal of chemical physics, 124(4):044109, 2006.

[39] Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. The journal of chemical physics, 121(9):4059–4067, 2004.

[40] Chris Williams Carl Edward Rasmussen, Hannes Nickisch. Gpml.

[41] Horatio Scott Carslaw and John Conrad Jaeger. Conduction of heat in solids. Number BOOK. Clarendon press, 1992.

[42] Greg L Casini, David Graham, David Heine, Robert L Garcea, and David T Wu. In vitro papillomavirus capsid assembly analyzed by light scattering. Virology, 325(2):320–327, 2004.

[43] Brian T Castle and David J Odde. Brownian dynamics of subunit addition-loss kinetics and thermodynamics in linear polymer self-assembly. Biophysical journal, 105(11):2528–2540, 2013.

[44] Pavel Castro-Villarreal, Alejandro Villada-Balbuena, José Miguel Méndez-Alcaraz, Ramón Castañeda-Priego, and Sendic Estrada-Jiménez. A brownian dynamics algorithm for colloids in curved manifolds. The Journal of chemical physics, 140(21):214115, 2014.

[45] Maura Cescatti, Daniela Saverioni, Sabina Capellari, Fabrizio Tagliavini, Tetsuyuki Kitamoto, James Ironside, Armin Giese, and Piero Parchi. Analysis of conformational stability of abnormal prion protein aggregates across the spectrum of creutzfeldt-jakob disease prions. Journal of virology, pages JVI–00144, 2016.

[46] Barnali N Chaudhuri. Emerging applications of small angle solution scattering in structural biology. Protein Science, 24(3):267–276, 2015.

[47] Chao Chen, C Cheng Kao, and Bogdan Dragnea. Self-assembly of brome mosaic virus capsids: Insights from shorter time-scale experiments†. The Journal of Physical Chemistry A, 112(39):9405–9412, 2008.

[48] Maelenn Chevreuil, Didier Law-Hine, Jingzhi Chen, Stéphane Bressanelli, Sophie Combet, Doru Constantin, Jéril Degrouard, Johannes Möller, Mehdi Zeghal, and Guillaume Tresset. Nonequilibrium self-assembly dynamics of icosahedral viral capsids packaging genome or polyelectrolyte. Nature communications, 9(1):3071, 2018.

[49] Wei-Xiang Chew, Kazunari Kaizu, Masaki Watabe, Sithi V Muniandy, Koichi Takahashi, and Satya NV Arjunan. Reaction-diffusion kinetics on lattice at the microscopic scale. Physical Review E, 98(3):032418, 2018.

[50] I-Chun Chou and Eberhard O Voit. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Mathematical biosciences, 219(2):57–83, 2009.

[51] Nathan Clement, Muhibur Rasheed, and Chandrajit Lal Bajaj. Viral capsid assembly: A quantified uncertainty approach. Journal of Computational Biology, 25(1):51–71, 2018.

[52] Scott D Cohen, Alan C Hindmarsh, and Paul F Dubois. Cvode, a stiff/nonstiff ode solver in c. Computers in physics, 10(2):138–143, 1996.

[53] Andrew G Cole. Modulators of hbv capsid assembly as an approach to treating hepatitis b virus infection. Current Opinion in Pharmacology, 30:131–137, 2016.

[54] Frank C Collins and George E Kimball. Diffusion-controlled reaction rates. Journal of colloid science, 4(4):425–437, 1949.

[55] Joshua Colvin, Michael I Monine, Ryan N Gutenkunst, William S Hlavacek, Daniel D Von Hoff, and Richard G Posner. Rulemonkey: software for stochastic simulation of rule-based models. BMC bioinformatics, 11(1):1, 2010.

[56] Le Cong, F Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D Hsu, Xuebing Wu, Wenyan Jiang, Luciano A Marraffini, et al. Multiplex genome engineering using crispr/cas systems. Science, 339(6121):819–823, 2013.

[57] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. Introduction to derivative-free optimization. SIAM, Philadelphia, 2009.

[58] Daniel J Cosgrove. Re-constructing our models of cellulose and primary cell wall assembly. Current opinion in plant biology, 22:122–131, 2014.

[59] Ann E Cowan, Ion I Moraru, James C Schaff, Boris M Slepchenko, and Leslie M Loew. Spatial modeling of cell signaling networks. Methods in cell biology, 110:195, 2012.

[60] Marta del Álamo, Germán Rivas, and Mauricio G Mateu. Effect of macromolecular crowding agents on human immunodeficiency virus type 1 capsid protein assembly in vitro. Journal of virology, 79(22):14271–14281, 2005.

[61] Wouter K Den Otter, Marten R Renes, and WJ Briels. Self-assembly of three-legged patchy particles into polyhedral cages. Journal of Physics: Condensed Matter, 22(10):104103, 2010.

[62] David W Denning. Minimizing fungal disease deaths will allow the unaids target of reducing annual aids deaths below 500 000 by 2020 to be realized. Phil. Trans. R. Soc. B, 371(1709):20150468, 2016.

[63] Arshad Desai and Timothy J Mitchison. Microtubule polymerization dynamics. Annual review of cell and developmental biology, 13(1):83–117, 1997.

[64] Bob Diamond, David Krahl, Anthony Nastasi, and Peter Tag. Extendsim advanced techology: integrated simulation database. In Proceedings of the Winter Simulation Conference, pages 32–39. Winter Simulation Conference, 2010.

[65] J Fernando Diaz, Jose M Andreu, Greg Diakun, Elizabeth Towns-Andrews, and Joan Bordas. Structural intermediates in the assembly of taxoid-induced microtubules and gdp-tubulin double rings: time-resolved x-ray scattering. Biophysical journal, 70(5):2408–2420, 1996.

[66] Aleksandar Donev, Chiao-Yu Yang, and Changho Kim. Efficient reactive brownian dynamics. The Journal of chemical physics, 148(3):034103, 2018.

[67] Rory M Donovan, Andrew J Sedgewick, James R Faeder, and Daniel M Zuckerman. Efficient stochastic simulation of chemical kinetics networks using a weighted ensemble of trajectories. The Journal of chemical physics, 139(11):09B642_1, 2013.

[68] Andrew Duncan, Radek Erban, and Konstantinos Zygalakis. Hybrid framework for the simulation of stochastic chemical kinetics. arXiv preprint arXiv:1512.03988, 2015.

[69] David Duvenaud. Github - additive gaussian processes.

[70] Eric C Dykeman, Peter G Stockley, and Reidun Twarock. Building a viral capsid in the presence of genomic rna. Physical Review E, 87(2):022717, 2013.

[71] Eric C Dykeman, Peter G Stockley, and Reidun Twarock. Solving a levinthal's paradox for virus assembly identifies a unique antiviral strategy. Proceedings of the National Academy of Sciences, 111(14):5361–5366, 2014.

[72] Sharon Eden, Rajat Rohatgi, Alexandre V Podtelejnikov, Matthias Mann, and Marc W Kirschner. Mechanism of regulation of wave1-induced actin nucleation by rac1 and nck. Nature, 418(6899):790–793, 2002.

[73] Marc Edwards, Adam Zwolak, Dorothy A Schafer, David Sept, Roberto Dominguez, and John A Cooper. Capping protein regulators fine-tune actin assembly dynamics. Nature Reviews Molecular Cell Biology, 15(10):677–689, 2014.

[74] Oren M Elrad and Michael F Hagan. Mechanisms of size control and polymorphism in viral capsid assembly. Nano letters, 8(11):3850–3857, 2008.

[75] Oren M Elrad and Michael F Hagan. Encapsulation of a polymer by an icosahedral virus. Physical biology, 7(4):045003, 2010.

[76] Dan Endres, Masaki Miyahara, Paul Moisant, and Adam Zlotnick. A reaction landscape identifies the intermediates critical for self-assembly of virus capsids and other polyhedral structures. Protein science, 14(6):1518–1525, 2005.

[77] Åsa EY Engqvist-Goldstein and David G Drubin. Actin assembly and endocytosis: from yeast to mammals. Annual review of cell and developmental biology, 19(1):287–332, 2003.

[78] Åsa EY Engqvist-Goldstein, Robin A Warren, Michael M Kessels, James H Keen, John Heuser, and David G Drubin. The actin-binding protein hip1r associates with clathrin during early stages of endocytosis and promotes clathrin assembly in vitro. The Journal of cell biology, 154(6):1209–1224, 2001.

[79] Radek Erban. From molecular dynamics to brownian dynamics. In Proc. R. Soc. A, volume 470(2167), page 20140036. The Royal Society, 2014.

[80] Radek Erban, Mark B Flegg, and Garegin A Papoian. Multiscale stochastic reaction–diffusion modeling: application to actin dynamics in filopodia. Bulletin of mathematical biology, 76(4):799–818, 2014.

[81] Wesley J Errington, M Qasim Khan, Stephanie A Bueler, John L Rubinstein, Avijit Chakrabartty, and Gilbert G Privé. Adaptor protein self-assembly drives the control of a cullin-ring ubiquitin ligase. Structure, 20(7):1141–1153, 2012.

[82] James R Faeder, Michael L Blinov, Byron Goldstein, and William S Hlavacek. Rule-based modeling of biochemical networks. Complexity, 10(4):22–41, 2005.

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

[84] Mohammad Fallahi-Sichani, JoAnne L Flynn, Jennifer J Linderman, and Denise E Kirschner. Differential risk of tuberculosis reactivation among anti-tnf therapies is due to drug binding kinetics and permeability. The Journal of Immunology, 188(7):3169–3178, 2012.

[85] Joseph Fass, Chi Pak, James Bamburg, and Alex Mogilner. Stochastic simulation of actin dynamics reveals the role of annealing and fragmentation. Journal of theoretical biology, 252(1):173–183, 2008.

[86] Benjamin Franz, Mark B Flegg, S Jonathan Chapman, and Radek Erban. Multiscale reaction-diffusion algorithms: Pde-assisted brownian dynamics. SIAM Journal on Applied Mathematics, 73(3):1224–1247, 2013.

[87] John M Frazier, Yaroslav Chushak, and Brent Foy. Stochastic simulation and analysis of biomolecular reaction networks. BMC systems biology, 3(1):1, 2009.

[88] Akira Funahashi, Yukiko Matsuoka, Akiya Jouraku, Mineo Morohashi, Norihiro Kikuchi, and Hiroaki Kitano. Celldesigner 3.5: a versatile modeling tool for biochemical networks. Proceedings of the IEEE, 96(8):1254–1265, 2008.

[89] Kevin Gardner and Vann Bennett. Modulation of spectrin–actin assembly by erythrocyte adducin. Nature, 328(6128):359–362, 1987.

[90] Melissa K Gardner, Alan J Hunt, Holly V Goodson, and David J Odde. Microtubule assembly dynamics: new insights at the nanoscale. Current opinion in cell biology, 20(1):64–70, 2008.

[91] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry, 81(25):2340–2361, 1977.

[92] Daniel T Gillespie. A rigorous derivation of the chemical master equation. Physica A: Statistical Mechanics and its Applications, 188(1-3):404–425, 1992.

[93] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. The Journal of Chemical Physics, 115(4):1716–1733, 2001.

[94] Daniel T Gillespie. Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem., 58:35–55, 2007.

[95] Daniel T Gillespie. A diffusional bimolecular propensity function. The Journal of chemical physics, 131(16):164109, 2009.

[96] Daniel T Gillespie, Andreas Hellander, and Linda R Petzold. Perspective: Stochastic algorithms for chemical kinetics. The Journal of chemical physics, 138(17):05B201_1, 2013.

[97] Daniel T Gillespie, Effrosyni Seitaridou, and Carol A Gillespie. The small-voxel tracking algorithm for simulating chemical reactions among diffusing molecules. The Journal of chemical physics, 141(23):12B649_1, 2014.

[98] Martin Ginkel, Andreas Kremling, Torsten Nutsch, Robert Rehner, and Ernst Dieter Gilles. Modular modeling of cellular systems with promot/diva. Bioinformatics, 19(9):1169–1176, 2003.

[99] Nail M Gizzatkulov, Igor I Goryanin, Eugeny A Metelkin, Ekaterina A Mogilevskaya, Kirill V Peskov, and Oleg V Demin. Dbsolve optimum: a software package for kinetic modeling which allows dynamic visualization of simulation results. BMC systems biology, 4(1):109, 2010.

[100] Otto Glatter and Otto Kratky. Small angle X-ray scattering. Academic press, London, 1982.

[101] Andrew Golightly and Darren J Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. Interface focus, page rsfs20110047, 2011.

[102] Rita Graceffa, R Paul Nobrega, Raul A Barrea, Sagar V Kathuria, Srinivas Chakravarthy, Osman Bilsel, and Thomas C Irving. Sub-millisecond time-resolved saxs using a continuous-flow mixer and x-ray microbeam. Journal of synchrotron radiation, 20(6):820–825, 2013.

[103] Gerd Gruenert, Bashar Ibrahim, Thorsten Lenser, Maiko Lohel, Thomas Hinze, and Peter Dittrich. Rule-based spatial modeling with diffusing, geometrically constrained molecules. BMC bioinformatics, 11(1):1, 2010.

[104] Gerd Grünert and Peter Dittrich. Using the srsim software for spatial and rule-based modeling of combinatorially complex biochemical reaction systems. In International Conference on Membrane Computing, pages 240–256. Springer, 2010.

[105] Michael F Hagan. Modeling viral capsid assembly. Advances in chemical physics, 155:1, 2014.

[106] Michael F Hagan and David Chandler. Dynamic pathways for viral capsid assembly. Biophysical journal, 91(1):42–54, 2006.

[107] Michael F Hagan and Oren M Elrad. Understanding the concentration dependence of viral capsid assembly kinetics—the origin of the lag time and identifying the critical nucleus size. Biophysical journal, 98(6):1065–1074, 2010.

[108] Michael F Hagan, Oren M Elrad, and Robert L Jack. Mechanisms of kinetic trapping in self-assembly and phase transformation. The Journal of chemical physics, 135(10):104115, 2011.

[109] Petter Hammar, Prune Leroy, Anel Mahmutovic, Erik G Marklund, Otto G Berg, and Johan Elf. The lac repressor displays facilitated diffusion in living cells. Science, 336(6088):1595–1598, 2012.

[110] Ronald Hancock. A role for macromolecular crowding effects in the assembly and function of compartments in the nucleus. Journal of structural biology, 146(3):281–290, 2004.

[111] Maike MK Hansen, Lenny HH Meijer, Evan Spruijt, Roel JM Maas, Marta Ventosa Rosquelles, Joost Groen, Hans A Heus, and Wilhelm TS Huck. Macromolecular crowding creates heterogeneous environments of gene expression in picolitre droplets. Nature nanotechnology, 11(2):191–197, 2016.

[112] Leonard A Harris, Justin S Hogg, José-Juan Tapia, John AP Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P Sheehan, and James R Faeder. Bionetgen 2.2: advances in rule-based modeling. Bioinformatics, 32(21):3366–3368, 2016.

[113] Martin Hemberg, Sophia N Yaliraki, and Mauricio Barahona. Stochastic kinetics of viral capsid assembly based on detailed protein structures. Biophysical journal, 90(9):3029–3042, 2006.

[114] Yi-Wen Heng and Cheng-Gee Koh. Actin cytoskeleton dynamics and the cell division cycle. The international journal of biochemistry & cell biology, 42(10):1622–1633, 2010.

[115] William S Hlavacek, James R Faeder, Michael L Blinov, Alan S Perelson, and Byron Goldstein. The complexity of complexes in signal transduction. Biotechnology and bioengineering, 84(7):783–794, 2003.

[116] Neil A Holmes, John Walshaw, Richard M Leggett, Annabelle Thibessard, Kate A Dalton, Michael D Gillespie, Andrew M Hemmings, Bertolt Gust, and Gabriella H Kelemen. Coiled-coil protein scy is a key component of a multiprotein assembly controlling polarized growth in streptomyces. Proceedings of the National Academy of Sciences, 110(5):E397–E406, 2013.

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

[118] Michael Hucka, Frank T Bergmann, Stefan Hoops, Sarah M Keating, Sven Sahle, James C Schaff, Lucian P Smith, and Darren J Wilkinson. The systems biology markup language (sbml): language specification for level 3 version 1 core. J Integr Bioinforma, 12(2):266, 2015.

[119] Michael Hucka, ABBJ Finney, Benjamin J Bornstein, Sarah M Keating, Bruce E Shapiro, Joanne Matthews, Ben L Kovitz, Maria J Schilstra, Akira Funahashi, John C Doyle, et al. Evolving a lingua franca and associated software infrastructure for computational systems biology: the systems biology markup language (sbml) project. Systems biology, 1(1):41–53, 2004.

[120] Michael Hucka, Andrew Finney, Herbert M Sauro, Hamid Bolouri, John C Doyle, Hiroaki Kitano, Adam P Arkin, Benjamin J Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics, 19(4):524–531, 2003.

[121] Kathrin Hueging, Mandy Doepke, Gabrielle Vieyres, Dorothea Bankwitz, Anne Frentzen, Juliane Doerrbecker, Frauke Gumz, Sibylle Haid, Benno Wölk, Lars Kaderali, et al. Apolipoprotein e codetermines tissue tropism of hepatitis c virus and is crucial for viral cell-to-cell transmission by contributing to a postenvelopment step of assembly. Journal of virology, 88(3):1433–1446, 2014.

[122] Waltraud Huyer and Arnold Neumaier. Global optimization by multilevel coordinate search. Journal of Global Optimization, 14(4):331–355, 1999.

[123] Waltraud Huyer and Arnold Neumaier. Snobfit–stable noisy optimization by branch and fit. ACM Transactions on Mathematical Software (TOMS), 35(2):9, 2008.

[124] Ioana M Ilie, Wouter K den Otter, and Wim J Briels. Rotational brownian dynamics simulations of clathrin cage formation. The Journal of chemical physics, 141(6):065101, 2014.

[125] Samuel A Isaacson. The reaction-diffusion master equation as an asymptotic approximation of diffusion to a small target. SIAM Journal on Applied Mathematics, 70(1):77–111, 2009.

[126] Shintaro Iwasaki, Hiroshi M Sasaki, Yuriko Sakaguchi, Tsutomu Suzuki, Hisashi Tadakuma, and Yukihide Tomari. Defining fundamental steps in the assembly of the drosophila rnai enzyme complex. Nature, 521(7553):533–536, 2015.

[127] Farokh Jamalyaria, Rori Rohlfs, and Russell Schwartz. Queue-based method for efficient simulation of biological self-assembly systems. Journal of Computational Physics, 204(1):100–120, 2005.

[128] Neema Jamshidi and Bernhard Ø Palsson. Mass action stoichiometric simulation models: incorporating kinetics and regulation into stoichiometric models. Biophysical journal, 98(2):175–185, 2010.

[129] Paul A Janmey. Phosphoinositides and calcium as regulators of cellular actin assembly and disassembly. Annual Review of Physiology, 56(1):169–191, 1994.

[130] Malene Hillerup Jensen, Katrine Nørgaard Toft, Gabriel David, Svend Havelund, Javier Pérez, and Bente Vestergaard. Time-resolved saxs measurements facilitated by online hplc buffer exchange. Journal of synchrotron radiation, 17(6):769–773, 2010.

[131] Wenjuan Jiang, Juntao Luo, and Shikha Nangia. Multiscale approach to investigate self-assembly of telodendrimer based nanocarriers for anticancer drug delivery. Langmuir, 31(14):4270–4280, 2015.

[132] Jennifer M Johnson, Jinghua Tang, Yaw Nyame, Deborah Willits, Mark J Young, and Adam Zlotnick. Regulating self-assembly of spherical oligomers. Nano letters, 5(4):765–770, 2005.

[133] Donald R Jones. Direct global optimization algorithm. In Encyclopedia of optimization, pages 431–440. Springer, Boston, 2001.

[134] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. Journal of Optimization Theory and Applications, 79(1):157–181, 1993.

[135] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4):455–492, 1998.

[136] Gyungah Jun, Carla A Ibrahim-Verbaas, Maria Vronskaya, Jean-Charles Lambert, Jaeyoon Chung, Adam C Naj, Brian W Kunkle, Li-San Wang, Joshua C Bis, Céline Bellenguez, et al. A novel alzheimer disease locus located near the gene encoding tau protein. Molecular psychiatry, 2015.

[137] Niklas O Junker, Farzaneh Vaghefikia, Alyazan Albarghash, Henning Höfig, Daryan Kempe, Julia Walter, Julia Otten, Martina Pohl, Alexandros Katranidis, Simone Wiegand, et al. The impact of molecular crowding on translational mobility and conformational properties of biological macromolecules. The Journal of Physical Chemistry B, 2019.

[138] K. Kaizu, K. Nishida, Y. Sakamoto, S. Kato, T. Niina, N. Nishida, M. Koizumi, N. Aota, , and K. Takahashi. E-Cell version 4.

[139] Ziya Kalay. Kinetics of self-assembly via facilitated diffusion: Formation of the transcription complex. Physical Review E, 92(4):042716, 2015.

[140] Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I Glass, and Markus W Covert. A whole-cell computational model predicts phenotype from genotype. Cell, 150(2):389–401, 2012.

[141] Sagar V Kathuria, Liang Guo, Rita Graceffa, Raul Barrea, R Paul Nobrega, C Robert Matthews, Thomas C Irving, and Osman Bilsel. Minireview: Structural insights into early folding events using continuous-flow time-resolved small-angle x-ray scattering. Biopolymers, 95(8):550–558, 2011.

[142] Cihan Kaya, Mary H Cheng, Ethan R Block, Tom M Bartol, Terrence J Sejnowski, Alexander Sorkin, James R Faeder, and Ivet Bahar. Heterogeneities in axonal structure and transporter distribution lower dopamine reuptake efficiency. eNeuro, 5(1), 2018.

[143] Thomas Keef, Cristian Micheletti, and Reidun Twarock. Master equation approach to the assembly of viral capsids. Journal of theoretical biology, 242(3):713–721, 2006.

[144] Rex A Kerr, Thomas M Bartol, Boris Kaminsky, Markus Dittrich, Jen-Chien Jack Chang, Scott B Baden, Terrence J Sejnowski, and Joel R Stiles. Fast monte carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. SIAM journal on scientific computing, 30(6):3126–3149, 2008.

[145] Taeyoon Kim, Wonmuk Hwang, Hyungsuk Lee, and Roger D Kamm. Computational analysis of viscoelastic properties of crosslinked actin networks. PLoS Comput Biol, 5(7):e1000439, 2009.

[146] Young C Kim, Robert B Best, and Jeetain Mittal. Macromolecular crowding effects on protein–protein binding affinity and specificity. The Journal of chemical physics, 133(20):205101, 2010.

[147] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. European journal of operational research, 192(3):707–716, 2009.

[148] Stanislav Kler, Roi Asor, Chenglei Li, Avi Ginsburg, Daniel Harries, Ariella Oppenheim, Adam Zlotnick, and Uri Raviv. Rna encapsidation by sv40-derived nanoparticles follows a rapid two-state mechanism. Journal of the American Chemical Society, 134(21):8823–8830, 2012.

[149] Klaus Klumpp and Thibaut Crépin. Capsid proteins of enveloped viruses as antiviral drug targets. Current opinion in virology, 5:63–71, 2014.

[150] Tuomas PJ Knowles, Michele Vendruscolo, and Christopher M Dobson. The amyloid state and its association with protein misfolding diseases. Nature reviews Molecular cell biology, 15(6):384–396, 2014.

[151] Dieter Kressler, Ed Hurt, and Jochen Baβler. Driving ribosome assembly. Biochimica et Biophysica Acta (BBA)-Molecular Cell Research, 1803(6):673–683, 2010.

[152] M Senthil Kumar and Russell Schwartz. A parameter estimation technique for stochastic self-assembly systems and its application to human papillomavirus self-assembly. Physical biology, 7(4):045005, 2010.

[153] Sudha Kumari, David Depoil, Roberta Martinelli, Edward Judokusumo, Guillaume Carmona, Frank B Gertler, Lance C Kam, Christopher V Carman, Janis K Burkhardt, Darrell J Irvine, et al. Actin foci facilitate activation of the phospholipase c-γ in primary t lymphocytes via the wasp pathway. Elife, 4:e04953, 2015.

[154] Louie Lamorte, Steve Titolo, Christopher T Lemke, Nathalie Goudreau, Jean-François Mercier, Elizabeth Wardrop, Vaibhav B Shah, Uta K von Schwedler, Charles Langelier, Soma SR Banik, et al. Discovery of novel small-molecule hiv-1 replication inhibitors that stabilize capsid complexes. Antimicrobial agents and chemotherapy, 57(10):4622–4631, 2013.

[155] Ayala Lampel, Yaron Bram, Anat Ezer, Ronit Shaltiel-Kario, Jamil S Saad, Eran Bacharach, and Ehud Gazit. Targeting the early step of building block organization in viral capsid assembly. ACS chemical biology, 10(8):1785–1790, 2015.

[156] Noel D Lazo, Marianne A Grant, Margaret C Condron, Alan C Rigby, and David B Teplow. On the nucleation of amyloid β-protein monomer folding. Protein Science, 14(6):1581–1596, 2005.

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

[158] D Leckband and S Sivasankar. Forces controlling protein interactions: theory and experiment. Colloids and surfaces B: Biointerfaces, 14(1):83–97, 1999.

[159] Deborah Leckband and Jacob Israelachvili. Intermolecular forces in biology. Quarterly reviews of biophysics, 34(02):105–267, 2001.

[160] Kelly K Lee, Hiro Tsuruta, Roger W Hendrix, Robert L Duda, and John E Johnson. Cooperative reorganization of a 420 subunit virus capsid. Journal of molecular biology, 352(3):723–735, 2005.

[161] Nathan N Lee, Venkata R Chalamcharla, Francisca Reyes-Turcu, Sameet Mehta, Martin Zofall, Vanivilasini Balachandran, Jothy Dhakshnamoorthy, Nitika Taneja, Soichiro Yamanaka, Ming Zhou, et al. Mtr4-like protein coordinates nuclear rna processing for heterochromatin assembly and for telomere maintenance. Cell, 155(5):1061–1074, 2013.

[162] Gene-Wei Li, David Burkhardt, Carol Gross, and Jonathan S Weissman. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. Cell, 157(3):624–635, 2014.

[163] Shi-Hua Li and Xiao-Jiang Li. Huntingtin–protein interactions and the pathogenesis of huntington's disease. TRENDS in Genetics, 20(3):146–154, 2004.

[164] Yen Ting Lin, Song Feng, and William S Hlavacek. Scaling methods for accelerating kinetic monte carlo simulations of chemical reaction networks. The Journal of Chemical Physics, 150(24):244101, 2019.

[165] Larry Lok and Roger Brent. Automatic generation of cellular reaction networks with moleculizer 1.0. Nature biotechnology, 23(1):131–136, 2005.

[166] Elkin Lopez-Fontal, Anna Grochmal, Tom Foran, Lilia Milanesi, and Salvador Tomas. Ship in a bottle: confinement-promoted self-assembly. Chemical science, 9(7):1760–1768, 2018.

[167] Baoliang Ma, Jinbing Xie, Liangshu Wei, and Wei Li. Macromolecular crowding modulates the kinetics and morphology of amyloid self-assembly by β-lactoglobulin. International journal of biological macromolecules, 53:82–87, 2013.

[168] Taras Makhnevych and Walid A Houry. The role of hsp90 in protein complex assembly. Biochimica et Biophysica Acta (BBA)-Molecular Cell Research, 1823(3):674–682, 2012.

[169] Arakaparampil M Mathai and Serge B Provost. Quadratic forms in random variables: theory and applications. Dekker, 1992.

[170] Tsutomu Matsui, Hiro Tsuruta, and John E Johnson. Balanced electrostatic and structural forces guide the large conformational change associated with maturation of t= 4 virus. Biophysical journal, 98(7):1337–1343, 2010.

[171] Richard Matthews and Christos N Likos. Structures and pathways for clathrin self-assembly in the bulk and on membranes. Soft Matter, 9(24):5794–5806, 2013.

[172] Pedro Mendes. Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Computer applications in the biosciences: CABIOS, 9(5):563–571, 1993.

[173] Allen P Minton. Implications of macromolecular crowding for protein assembly. Current opinion in structural biology, 10(1):34–39, 2000.

[174] Navodit Misra, Daniel Lees, Tiequan Zhang, and Russell Schwartz. Pathway complexity of model virus capsid assembly systems. Computational and Mathematical Methods in Medicine, 9(3-4):277–293, 2008.

[175] Navodit Misra and Russell Schwartz. Efficient stochastic sampling of first-passage times with applications to self-assembly simulations. The Journal of chemical physics, 129(20):204109, 2008.

[176] S Mondragón-Rodríguez, G Perry, J Luna-Muñoz, MC Acevedo-Aquino, and S Williams. Phosphorylation of tau protein at sites ser396–404 is one of the earliest events in alzheimer's disease and down syndrome. Neuropathology and applied neurobiology, 40(2):121–135, 2014.

[177] Michael I Monine, Richard G Posner, Paul B Savage, James R Faeder, and William S Hlavacek. Modeling multivalent ligand-receptor interactions with steric constraints on configurations of cell-surface receptor aggregates. Biophysical journal, 98(1):48–56, 2010.

[178] Olivia L Mooren, Brian J Galletta, and John A Cooper. Roles for actin assembly in endocytosis. Biochemistry, 81(1):661, 2012.

[179] Alexander Yu Morozov, Robijn F Bruinsma, and Joseph Rudnick. Assembly of viruses and the pseudo-law of mass action. The Journal of chemical physics, 131(15):155101, 2009.

[180] Anke M Mulder, Craig Yoshioka, Andrea H Beck, Anne E Bunner, Ronald A Milligan, Clinton S Potter, Bridget Carragher, and James R Williamson. Visualizing ribosome biogenesis: parallel assembly pathways for the 30s subunit. Science, 330(6004):673–677, 2010.

[181] Michaela Müller-McNicoll and Karla M Neugebauer. How cells get the message: dynamic assembly and function of mrna-protein complexes. Nature Reviews Genetics, 14(4):275–287, 2013.

[182] Ambarish Nag, Michael I Monine, James R Faeder, and Byron Goldstein. Aggregation of membrane proteins by cytosolic cross-linkers: theory and simulation of the lat-grb2-sos1 system. Biophysical journal, 96(7):2604–2623, 2009.

[183] K Razi Naqvi, KJ Mork, and S Waldenstrom. Diffusion-controlled reaction kinetics. equivalence of the particle pair approach of noyes and the concentration gradient approach of collins and kimball. The Journal of Physical Chemistry, 84(11):1315–1319, 1980.

[184] K Razi Naqvi, S Waldenstrøm, and KJ Mork. Kinetics of diffusion-mediated bimolecular reactions. a new theoretical framework. The Journal of Physical Chemistry, 86(24):4750–4756, 1982.

[185] Hung D Nguyen, Vijay S Reddy, and Charles L Brooks. Deciphering the kinetic mechanism of spontaneous self-assembly of icosahedral capsids. Nano Letters, 7(2):338–344, 2007.

[186] Dan V Nicolau and Kevin Burrage. Stochastic simulation of chemical reactions in spatially complex media. Computers & Mathematics with Applications, 55(5):1007–1018, 2008.

[187] Jeffrey C Nolz, Martin E Fernandez-Zapico, and Daniel D Billadeau. Tcr/cd28-stimulated actin dynamics are required for nfat1-mediated transcription of c-rel leading to cd28 response element activation. The Journal of Immunology, 179(2):1104–1112, 2007.

[188] Richard M Noyes. Models relating molecular reactivity and diffusion in liquids. Journal of the American Chemical Society, 78(21):5486–5490, 1956.

[189] RM Noyes. Prog. react. kinet. 1961.

[190] Kei-ichi Okazaki, Takato Sato, and Mitsunori Takano. Temperature-enhanced association of proteins due to electrostatic interaction: A coarse-grained simulation of actin–myosin binding. Journal of the American Chemical Society, 134(21):8918–8925, 2012.

[191] Fumio Oosawa. Size distribution of protein polymers. Journal of theoretical biology, 27(1):69–86, 1970.

[192] Fumio Oosawa and Sugie Higashi. Statistical thermodynamics of polymerization and polymorphism of protein. Progress in theoretical biology, 1:79–164, 1967.

[193] Tomas Opplestrup, Vasily V Bulatov, George H Gilmer, Malvin H Kalos, and Babak Sadigh. First-passage monte carlo algorithm: diffusion without all the hops. Physical review letters, 97(23):230602, 2006.

[194] Mark S Palmer, Aidan J Dryden, J Trevor Hughes, and John Collinge. Homozygous prion protein genotype predisposes to sporadic creutzfeldt–jakob disease. Nature, 352(6333):340–342, 1991.

[195] Dominique Pantaloni and Marie-France Carlier. How profilin promotes actin filament assembly in the presence of thymosin $\beta 4$. Cell, 75(5):1007–1014, 1993.

[196] Johan Paulsson. Models of stochastic gene expression. Physics of life reviews, 2(2):157–175, 2005.

[197] Suzanne R Pfeffer and James E Rothman. Biosynthetic protein transport and sorting by the endoplasmic reticulum and golgi. Annual review of biochemistry, 56(1):829–852, 1987.

[198] John W Pham and Erik J Sontheimer. Molecular requirements for rna-induced silencing complex assembly in the drosophila rna interference pathway. Journal of Biological Chemistry, 280(47):39278–39283, 2005.

[199] André Plagens, Vanessa Tripp, Michael Daume, Kundan Sharma, Andreas Klingl, Ajla Hrle, Elena Conti, Henning Urlaub, and Lennart Randau. In vitro assembly and activity of an archaeal crispr-cas type ia cascade interference complex. Nucleic acids research, 42(8):5125–5138, 2014.

[200] Rudolf Podgornik, M Alphan Aksoyoglu, Selcuk Yasar, Daniel Svensek, and V Adrian Parsegian. Dna equation of state: In vitro vs in viro. The Journal of Physical Chemistry B, 120(26):6051–6060, 2016.

[201] Thomas D Pollard and John A Cooper. Actin and actin-binding proteins. a critical evaluation of mechanisms and functions. Annual review of biochemistry, 55(1):987–1035, 1986.

[202] Michael JD Powell. Uobyqa: unconstrained optimization by quadratic approximation. Mathematical Programming, 92(3):555–582, 2002.

[203] Peter E Prevelige Jr, Dennis Thomas, and Jonathan King. Nucleation and growth phases in the polymerization of coat and scaffolding subunits into icosahedral procapsid shells. Biophysical journal, 64(3):824, 1993.

[204] Martin Pring, Marie Evangelista, Charles Boone, Changsong Yang, and Sally H Zigmond. Mechanism of formin-induced nucleation of actin filaments. Biochemistry, 42(2):486–496, 2003.

[205] Vishwanath R Lingappa, Clarence R Hurt, and Edward Garvey. Capsid assembly as a point of intervention for novel anti-viral therapeutics. Current pharmaceutical biotechnology, 14(5):513–523, 2013.

[206] DC Rapaport, JE Johnson, and J Skolnick. Supramolecular self-assembly: molecular dynamics modeling of polyhedral shell formation. Computer physics communications, 121:231–235, 1999.

[207] Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning, volume 1. MIT press, Cambridge, 2006.

[208] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. The Journal of Chemical Physics, 119(24):12784–12794, 2003.

[209] Diana C Resasco, Fei Gao, Frank Morgan, Igor L Novak, James C Schaff, and Boris M Slepchenko. Virtual cell: computational tools for modeling in cell biology. Wiley Interdisciplinary Reviews: Systems Biology and Medicine, 4(2):129–140, 2012.

[210] Marion Ritzi and Rolf Knippers. Initiation of genome replication: assembly and disassembly of replication-competent chromatin. Gene, 245(1):13–20, 2000.

[211] Germán Rivas, Javier A Fernández, and Allen P Minton. Direct observation of the enhancement of noncooperative protein self-assembly by macromolecular crowding: indefinite linear self-association of bacterial cell division protein ftsz. Proceedings of the National Academy of Sciences, 98(6):3150–3155, 2001.

[212] Francisco Rivero, B Koppel, Barbara Peracino, Salvatore Bozzaro, Florian Siegert, Cornelis J Weijer, Michael Schleicher, Richard Albrecht, and Angelika A Noegel. The role of the cortical cytoskeleton: F-actin crosslinking proteins protect against osmotic stress, ensure cell size, cell shape and motility, and contribute to phagocytosis and development. Journal of Cell Science, 109(11):2679–2691, 1996.

[213] Stephen L Rogers and Vladimir I Gelfand. Membrane trafficking, organelle transport, and the cytoskeleton. Current opinion in cell biology, 12(1):57–62, 2000.

[214] Jennifer L Ross, M Yusuf Ali, and David M Warshaw. Cargo transport: molecular motors navigate a complex cytoskeleton. Current opinion in cell biology, 20(1):41–47, 2008.

[215] Kole T Roybal, Taráz E Buck, Xiongtao Ruan, Baek Hwan Cho, Danielle J Clark, Rachel Ambler, Helen M Tunbridge, Jianwei Zhang, Paul Verkade, Christoph Wülfing, et al. Computational spatiotemporal analysis identifies wave2 and cofilin as joint regulators of costimulation-mediated t cell actin dynamics. Science signaling, 9(424):rs3–rs3, 2016.

[216] Kole T Roybal, Emily M Mace, Danielle J Clark, Alan D Leard, Andrew Herman, Paul Verkade, Jordan S Orange, and Christoph Wülfing. Modest interference with actin dynamics in primary t cell activation by antigen presenting cells preferentially affects lamellal signaling. PloS one, 10(8):e0133231, 2015.

[217] Robin Roychaudhuri, Mingfeng Yang, Minako M Hoshi, and David B Teplow. Amyloid β-protein assembly and alzheimer disease. Journal of Biological Chemistry, 284(8):4749–4753, 2009.

[218] Xiongtao Ruan, Christoph Wülfing, and Robert F Murphy. Image-based spatiotemporal causality inference for protein signaling networks. Bioinformatics, 33(14):i217–i224, 2017.

[219] R Rubenstein, PC Gray, TJ Cleland, MS Piltch, WS Hlavacek, RM Roberts, J Ambrosiano, and J-I Kim. Dynamics of the nucleated polymerization model of prion replication. Biophysical chemistry, 125(2):360–367, 2007.

[220] Teresa Ruiz-Herrero and Michael F Hagan. Simulations show that virus assembly and budding are facilitated by membrane microdomains. Biophysical journal, 108(3):585–595, 2015.

[221] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. Statistical science, pages 409–423, 1989.

[222] Isabelle Sagot, Avital A Rodal, James Moseley, Bruce L Goode, and David Pellman. An actin nucleation mechanism mediated by bni1 and profilin. Nature cell biology, 4(8):626–631, 2002.

[223] Jayodita C Sanghvi, Sergi Regot, Silvia Carrasco, Jonathan R Karr, Miriam V Gutschow, Benjamin Bolival Jr, and Markus W Covert. Accelerated discovery via a whole-cell model. Nature methods, 10(12):1192–1195, 2013.

[224] Andela Saric, Yassmine C Chebaro, Tuomas PJ Knowles, and Daan Frenkel. Crucial role of nonspecific interactions in amyloid nucleation. Proceedings of the National Academy of Sciences, 111(50):17869–17874, 2014.

[225] Daisuke Sato, Hideaki Ohtomo, Yoshiteru Yamada, Takaaki Hikima, Atsushi Kurobe, Kazuo Fujiwara, and Masamichi Ikeguchi. Ferritin assembly revisited: a time-resolved small-angle x-ray scattering study. Biochemistry, 55(2):287–293, 2016.

[226] Marissa G Saunders and Gregory A Voth. Comparison between actin filament models: coarse-graining reveals essential differences. Structure, 20(4):641–653, 2012.

[227] Herbert M Sauro, Michael Hucka, Andrew Finney, Cameron Wellock, Hamid Bolouri, John Doyle, and Hiroaki Kitano. Next generation simulation tools: the systems biology workbench and biospice integration. Omics A Journal of Integrative Biology, 7(4):355–372, 2003.

[228] Dorothy A Schafer, Phillip B Jennings, and John A Cooper. Dynamics of capping protein and actin assembly in vitro: uncapping barbed ends by polyphosphoinositides. The Journal of Cell Biology, 135(1):169–179, 1996.

[229] James Schaff, Charles C Fink, Boris Slepchenko, John H Carson, and Leslie M Loew. A general computational framework for modeling cellular structure and function. Biophysical journal, 73(3):1135, 1997.

[230] James C Schaff and Leslie M Loew. The virtual cell. In Pacific Symposium on Biocomputing, volume 4, pages 228–239. Citeseer, 1999.

[231] James C Schaff, Dan Vasilescu, Ion I Moraru, Leslie M Loew, and Michael L Blinov. Rule-based modeling with virtual cell. Bioinformatics, page btw353, 2016.

[232] Jeremy D Schmit, Kingshuk Ghosh, and Ken Dill. What drives amyloid molecules to assemble into oligomers and fibrils? Biophysical journal, 100(2):450–458, 2011.

[233] S Schnell and TE Turner. Reaction kinetics in intracellular environments with macromolecular crowding: simulations and rate laws. Progress in biophysics and molecular biology, 85(2):235–260, 2004.

[234] Russell Schwartz, Peter W Shor, Peter E Prevelige, and Bonnie Berger. Local rules simulation of the kinetics of virus capsid self-assembly. Biophysical journal, 75(6):2626–2636, 1998.

[235] Dianne S Schwarz, György Hutvágner, Tingting Du, Zuoshang Xu, Neil Aronin, and Phillip D Zamore. Asymmetry in the assembly of the rnai enzyme complex. Cell, 115(2):199–208, 2003.

[236] Dennis J Selkoe. Cell biology of protein misfolding: the examples of alzheimer's and parkinson's diseases. Nature cell biology, 6(11):1054–1061, 2004.

[237] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1):148–175, 2016.

[238] Michael P Sheetz, Denise B Wayne, and Alan L Pearlman. Extension of filopodia by motor-dependent actin assembly. Cell motility and the cytoskeleton, 22(3):160–169, 1992.

[239] Jiong Shi, David B Friedman, and Christopher Aiken. Retrovirus restriction by trim5 proteins requires recognition of only a small fraction of viral capsid subunits. Journal of virology, 87(16):9271–9278, 2013.

[240] Jonghyeon Shin, Paul Jardine, and Vincent Noireaux. Genome replication, synthesis, and assembly of the bacteriophage t7 in a single cell-free reaction. ACS synthetic biology, 1(9):408–413, 2012.

[241] Sushmita Singh and Adam Zlotnick. Observed hysteresis of virus capsid disassembly is implicit in kinetic models of assembly. Journal of Biological Chemistry, 278(20):18249–18255, 2003.

[242] Morten Slyngborg and Peter Fojan. A computational study of the self-assembly of the rfffr peptide. Physical Chemistry Chemical Physics, 17(44):30023–30036, 2015.

[243] Gregory R Smith, Lu Xie, Byoungkoo Lee, and Russell Schwartz. Applying molecular crowding models to simulations of virus capsid assembly in vitro. Biophysical journal, 106(1):310–320, 2014.

[244] Gregory R Smith, Lu Xie, and Russell Schwartz. Modeling effects of rna on capsid assembly pathways via coarse-grained stochastic simulation. PloS one, 11(5):e0156547, 2016.

[245] Stephen Smith and Ramon Grima. Breakdown of the reaction-diffusion master equation with nonelementary rates. Physical Review E, 93(5):052135, 2016.

[246] 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.

[247] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems, pages 2951–2959, 2012.

[248] Thomas R Sokolowski, Joris Paijmans, Laurens Bossen, Thomas Miedema, Martijn Wehrens, Nils B Becker, Kazunari Kaizu, Koichi Takahashi, Marileen Dogterom, and Pieter Rein ten Wolde. egfrd in all dimensions. The Journal of chemical physics, 150(5):054108, 2019.

[249] Jeffrey A Speir, Brian Bothner, Chunxu Qu, Deborah A Willits, Mark J Young, and John E Johnson. Enhanced local symmetry interactions globally stabilize a mutant virus capsid that maintains infectivity and capsid dynamics. Journal of virology, 80(7):3582–3591, 2006.

[250] Justin M Spiriti and Daniel M Zuckerman. Tabulation as a high-resolution alternative to coarse-graining protein interactions: Initial application to virus capsid subunits. Biophysical Journal, 110(3):495a, 2016.

[251] Thomas Splettstoesser, Kenneth C Holmes, Frank Noé, and Jeremy C Smith. Structural modeling and molecular dynamics simulation of the actin filament. Proteins: Structure, Function, and Bioinformatics, 79(7):2033–2043, 2011.

[252] Christopher B Stanley, Tatiana Perevozchikova, and Valerie Berthelier. Structural formation of huntingtin exon 1 aggregates probed by small-angle neutron scattering. Biophysical journal, 100(10):2504–2512, 2011.

[253] Joel R Stiles, Thomas M Bartol, et al. Monte carlo methods for simulating realistic synaptic microphysiology using mcell. Computational neuroscience: realistic modeling for experimentalists, pages 87–127, 2001.

[254] JR Stiles, D Van Helden, TM Bartol, and MM SALPETER. Miniature endplate current rise times< 100 mu s from improved dual recordgs can be modified with passive acetylcholine diffusion from a synaptic ves icle. Proceedings of the National Academy of Sciences of the United States of America, 93(12):5747–5752, 1996.

[255] Audrius B Stundzia and Charles J Lumsden. Stochastic simulation of coupled reaction–diffusion processes. Journal of computational physics, 127(1):196–207, 1996.

[256] D Svergun, Claudio Barberato, and Michel HJ Koch. Crysol–a program to evaluate x-ray solution scattering of biological macromolecules from atomic coordinates. Journal of applied crystallography, 28(6):768–773, 1995.

[257] Blake Sweeney, Tiequan Zhang, and Russell Schwartz. Exploring the parameter space of complex self-assembly through virus capsid models. Biophysical journal, 94(3):772–783, 2008.

[258] Koichi Takahashi, N Ishikawa, Yasuhiro Sadamoto, Hiroykui Sasamoto, Seiji Ohta, Akira Shiozawa, Fumihiko Miyoshi, Yasuhiro Naito, Yoichi Nakayama, and Masaru Tomita. E-cell 2: multi-platform e-cell simulation system. Bioinformatics, 19(13):1727–1729, 2003.

[259] Gintautas Tamulaitis, Česlovas Venclovas, and Virginijus Siksnys. Type iii crispr-cas immunity: Major differences brushed aside. Trends in Microbiology, 2016.

[260] Zhenning Tan, Megan L Maguire, Daniel D Loeb, and Adam Zlotnick. Genetically altering the thermodynamics and kinetics of hepatitis b virus capsid assembly has profound effects on virus replication in cell culture. Journal of virology, 87(6):3208–3216, 2013.

[261] Anastasia F Thévenin, Tia J Kowal, John T Fong, Rachael M Kells, Charles G Fisher, and Matthias M Falk. Proteins and mechanisms regulating gap-junction assembly, internalization, and degradation. Physiology, 28(2):93–116, 2013.

[262] Ines Thiele, Neil Swainston, Ronan MT Fleming, Andreas Hoppe, Swagatika Sahoo, Maike K Aurich, Hulda Haraldsdottir, Monica L Mo, Ottar Rolfsson, Miranda D Stobbe, et al. A community-driven global reconstruction of human metabolism. Nature biotechnology, 31(5):419–425, 2013.

[263] Brandon R Thomas, Lily A Chylek, Joshua Colvin, Suman Sirimulla, Andrew HA Clayton, William S Hlavacek, and Richard G Posner. Bionetfit: a fitting tool compatible with bionetgen, nfsim, and distributed computing environments. Bioinformatics, page btv655, 2015.

[264] Marcus Thomas. DESSA-CS, July 2020.

[265] Marcus Thomas and Russell Schwartz. Quantitative computational models of molecular self-assembly in systems biology. Physical Biology, 14(3):035003, 2017.

[266] Marcus Thomas and Russell Schwartz. A method for efficient bayesian optimization of self-assembly systems from scattering data. BMC systems biology, 12(1):65, 2018.

[267] Marcus Thomas and Russell S Schwartz. A diffusion based embedding of the stochastic simulation algorithm in continuous space. Biophysical Journal, 118(3):302a, 2020.

[268] Larry S Tobacman and Edward D Korn. The kinetics of actin nucleation and polymerization. Journal of Biological Chemistry, 258(5):3207–3214, 1983.

[269] Eeva Toivari, Tiina Manninen, Amit K Nahata, Tuula O Jalonen, and Marja-Leena Linne. Effects of transmitters and amyloid-beta peptide on calcium signals in rat cortical astrocytes: Fura-2am measurements and stochastic model simulations. PloS one, 6(3):e17914, 2011.

[270] 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, 15(1):72–84, 1999.

[271] Irina Tskvitaria-Fuller, Andrew L Rozelle, Helen L Yin, and Christoph Wülfing. Regulation of sustained actin dynamics by the tcr and costimulation as a mechanism of receptor localization. The Journal of Immunology, 171(5):2287–2295, 2003.

[272] Roman Tuma, Hiro Tsuruta, Kenneth H French, and Peter E Prevelige. Detection of intermediates and kinetic control during assembly of bacteriophage p22 procapsid. Journal of molecular biology, 381(5):1395–1406, 2008.

[273] Thomas E Turner, Santiago Schnell, and Kevin Burrage. Stochastic approaches for modelling in vivo reactions. Computational biology and chemistry, 28(3):165–178, 2004.

[274] Jessica K Tyler, Christopher R Adams, Shaw-Ree Chen, Ryuji Kobayashi, Rohinton T Kamakaka, and James T Kadonaga. The rcaf complex mediates chromatin assembly during dna replication and repair. Nature, 402(6761):555–560, 1999.

[275] Salvatore Valitutti, Mark Dessing, Klaus Aktories, Harald Gallati, and Antonio Lanzavecchia. Sustained signaling leading to t cell activation results from prolonged t cell receptor occupancy. role of t cell actin cytoskeleton. The Journal of experimental medicine, 181(2):577–584, 1995.

[276] H Van Beijeren, W Dong, and L Bocquet. Diffusion-controlled reactions: A revisit of noyes' theory. The Journal of Chemical Physics, 114(14):6265–6275, 2001.

[277] Briana Van Treeck, David SW Protter, Tyler Matheny, Anthony Khong, Christopher D Link, and Roy Parker. Rna self-assembly contributes to stress granule formation and defining the stress granule transcriptome. Proceedings of the National Academy of Sciences, 115(11):2734–2739, 2018.

[278] Jeroen S van Zon and Pieter Rein Ten Wolde. Green's-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space. The Journal of chemical physics, 123(23):234910, 2005.

[279] Jeroen S van Zon and Pieter Rein Ten Wolde. Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics. Physical review letters, 94(12):128103, 2005.

[280] Caroline Vance, Boris Rogelj, Tibor Hortobágyi, Kurt J De Vos, Agnes Lumi Nishimura, Jemeen Sreedharan, Xun Hu, Bradley Smith, Deborah Ruddy, Paul Wright, et al. Mutations in fus, an rna processing protein, cause familial amyotrophic lateral sclerosis type 6. Science, 323(5918):1208–1211, 2009.

[281] Bente Vestergaard, Minna Groenning, Manfred Roessle, Jette S Kastrup, Marco Van De Weert, James M Flink, Sven Frokjaer, Michael Gajhede, and Dmitri I Svergun. A helical structural nucleus is the primary elongating unit of insulin amyloid fibrils. PLoS biology, 5(5):e134, 2007.

[282] Viola Vogel. Mechanotransduction involving multimodular proteins: converting force into biochemical signals. Annu. Rev. Biophys. Biomol. Struct., 35:459–488, 2006.

[283] Margaritis Voliotis, Philipp Thomas, Ramon Grima, and Clive G Bowsher. Stochastic simulation of biomolecular networks in dynamic environments. PLoS Comput Biol, 12(6):e1004923, 2016.

[284] M Von Smoluchowski. Mathematical theory of the kinetics of the coagulation of colloidal solutions. Z. Phys. Chem, 92:129–168, 1917.

[285] Da Wang, Michiel Hermes, Ramakrishna Kotni, Yaoting Wu, Nikos Tasios, Yang Liu, Bart De Nijs, Ernest B Van Der Wee, Christopher B Murray, Marjolein Dijkstra, et al. Interplay between spherical confinement and particle shape on the self-assembly of rounded cubes. Nature communications, 9(1):2228, 2018.

[286] Hui Wang, Sam Lacoche, Ling Huang, Bin Xue, and Senthil K Muthuswamy. Rotational motion during three-dimensional morphogenesis of mammary epithelial acini relates to

laminin matrix assembly. Proceedings of the National Academy of Sciences, 110(1):163–168, 2013.

[287] Ning Wang, James P Butler, Donald E Ingber, et al. Mechanotransduction across the cell surface and through the cytoskeleton. Science, 260(5111):1124–1127, 1993.

[288] Ning Wang, Jessica D Tytell, and Donald E Ingber. Mechanotransduction at a distance: mechanically coupling the extracellular matrix with the nucleus. Nature reviews Molecular cell biology, 10(1):75–82, 2009.

[289] Ya-Juan Wang, Dong Lu, Yi-Bin Xu, Wei-Qiang Xing, Xian-Kun Tong, Gui-Feng Wang, Chun-Lan Feng, Pei-Lan He, Li Yang, Wei Tang, et al. A novel pyridazinone derivative inhibits hepatitis b virus replication by inducing genome-free capsid formation. Antimicrobial agents and chemotherapy, 59(11):7061–7072, 2015.

[290] David J Warne, Ruth E Baker, and Matthew J Simpson. Simulation and inference algorithms for stochastic biochemical reaction networks: from basic concepts to state-of-the-art. Journal of the Royal Society Interface, 16(151):20180943, 2019.

[291] Murray D Weingarten, Arthur H Lockwood, Shu-Ying Hwo, and Marc W Kirschner. A protein factor essential for microtubule assembly. Proceedings of the National Academy of Sciences, 72(5):1858–1862, 1975.

[292] Grzegorz Wieczorek and Piotr Zielenkiewicz. Influence of macromolecular crowding on protein-protein association rates—a brownian dynamics study. Biophysical journal, 95(11):5030–5036, 2008.

[293] Wikimedia Commons / C.rose.kennedy. Julia-colonna active site structure1, 2010.

[294] Wikimedia Commons / Isabella Daidone. Beta hairpin, 2006.

[295] Darren J Wilkinson. Bayesian methods in bioinformatics and computational systems biology. Briefings in bioinformatics, 8(2):109–116, 2007.

[296] Anders Wimo, Linus Jönsson, John Bond, Martin Prince, Bengt Winblad, and Alzheimer Disease International. The worldwide economic impact of dementia 2010. Alzheimer's & Dementia, 9(1):1–11, 2013.

[297] Marc Wortmann. Dementia: a global health priority-highlights from an adi and world health organization report. Alzheimer's research & therapy, 4(5):1, 2012.

[298] Lu Xie, Gregory Smith, and Russell Schwartz. Applying derivative-free optimization to fit kinetic parameters of viral capsid self-assembly models from multi-source bulk in vitro data. Biophysical Journal, 108(2):470a–471a, 2015.

[299] Lu Xie, Gregory R Smith, Xian Feng, and Russell Schwartz. Surveying capsid assembly pathways through simulation-based data fitting. Biophysical journal, 103(7):1545–1554, 2012.

[300] Lu Xie, Gregory R Smith, and Russell Schwartz. Derivative-free optimization of rate parameters of capsid assembly models from bulk in vitro data. IEEE/ACM transactions on computational biology and bioinformatics, 14(4):844–855, 2017.

[301] Wen Xu, Adam M Smith, James R Faeder, and G Elisabeta Marai. Rulebender: a visual interface for rule-based modeling. Bioinformatics, 27(12):1721–1722, 2011.

[302] Jin Yang, Michael I Monine, James R Faeder, and William S Hlavacek. Kinetic monte carlo method for rule-based modeling of biochemical networks. Physical Review E, 78(3):031910, 2008.

[303] Liuqing Yang, Jozsef Gal, Jing Chen, and Haining Zhu. Self-assembled fus binds active chromatin and regulates gene transcription. Proceedings of the National Academy of Sciences, 111(50):17809–17814, 2014.

[304] Zhu Yang, Michael Reeves, Jun Ye, Phong Trang, Li Zhu, Jingxue Sheng, Yu Wang, Ke Zen, Jianguo Wu, and Fenyong Liu. Rnase p ribozymes inhibit the replication of human cytomegalovirus by targeting essential viral capsid proteins. Viruses, 7(7):3345–3360, 2015.

[305] Casey A Ydenberg, Benjamin A Smith, Dennis Breitsprecher, Jeff Gelles, and Bruce L Goode. Cease-fire at the leading edge: New perspectives on actin filament branching, debranching, and cross-linking. Cytoskeleton, 68(11):596–602, 2011.

[306] Roya Zandi, Paul van der Schoot, David Reguera, Willem Kegel, and Howard Reiss. Classical nucleation theory of virus capsids. Biophysical journal, 90(6):1939–1948, 2006.

[307] Tiequan Zhang, Rori Rohlfs, and Russell Schwartz. Implementation of a discrete event simulator for biological self-assembly systems. pages 2223–2231, 2005.

[308] Tiequan Zhang and Russell Schwartz. Simulation study of the contribution of oligomer/oligomer binding to capsid assembly kinetics. Biophysical journal, 90(1):57–64, 2006.

[309] Adam Zlotnick. To build a virus capsid: an equilibrium model of the self assembly of polyhedral protein complexes. Journal of molecular biology, 241(1):59–67, 1994.

[310] Adam Zlotnick. Theoretical aspects of virus capsid assembly. Journal of Molecular Recognition, 18(6):479–490, 2005.

[311] Adam Zlotnick, Ryan Aldrich, Jennifer M Johnson, Pablo Ceres, and Mark J Young. Mechanism of capsid assembly for an icosahedral plant virus. Virology, 277(2):450–456, 2000.

[312] Adam Zlotnick, Jennifer M Johnson, Paul W Wingfield, Stephen J Stahl, and Dan Endres. A theoretical model successfully identifies features of hepatitis b virus capsid assembly. Biochemistry, 38(44):14644–14652, 1999.