Inferring Viral Capsid Self-Assembly Pathway from Bulk Experiment Measurement via Parameter Fitting Methods

Lu Xie

CMU-CB-15-105

April 2015

Department of Computational Biology
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Dr. Russell Schwartz, Chair
Dr. James Faeder
Dr. Fred Homa
Dr. Nikolaos Sahinidis

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

This research was supported by U.S. National Institute of Health award No. 1R01AI076318 and the Department of Computational Biology, Carnegie Mellon University. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, donors or the U.S. Government.

Keywords: Self-Assembly, Coarse-Grained Model, Stochastic Simulation, Parameter-Fitting

Once a Schwartz Lab member,

Always one.

Abstract

Self-assembly is a common phenomenon in the macro-molecular environment inside the cell and is critical for many cellular functions. Viral capsid assembly has been studied as a key model for self-assembly systems by researchers from different fields. There nonetheless remains a substaintial gap between experimental observations and current models, as the direct measurement of the assembly dynamics is currently intractible. Simulation-based methods can help bridge the gap, but the validity of such methods relies on the accuracy of a variety of physical parameters needed to instantiate the models, which also currently cannot be aquired by direct measurement.

The work of this thesis is focused on developing a parameter-learning framework that can infer kinetic parameters of viral assembly models by fitting the models to indirect bulk experimental measurements. The underlying rationale is based on the assumption that the set of parameters that minimizes the difference between simulated and experimental results would be the most plausible candidate. The framework extends existing stochastic self-assembly simulation methods, viral capsid models, and a prior heuristic optimization method to a flexible architecture that is adaptive to multiple data sources and alternative optimization methods.

The thesis specifically explores prospects for greater efficiency and accuracy through the use of more advanced algorithms or data sources for simulation-based model fitting. The framework has been tested on three *in vitro* viral assembly systems: human papillomavirus (HPV), heptatitis B virus (HBV) and cowpea chlorotic mottle virus (CCMV). The best fitting results from static light scattering (SLS) experiments suggest distinct *in vitro* assembly pathways for the three icosahedral viruses. Simulation experiments introducing synthetic non-covalent mass spectrometry (NCMS) data suggest that richer data sources can lead to substantial improvement in fitting accuracy. Complementary experiments on alternative optimization algorithms based on derivative free optimization (DFO) suggest that algorithmic advances can also substantially improve accuracy of model fits. Together, these results suggest that the methods can effectively reconstruct model parameters and assembly pathways given currently feasible algorithms and data sources, but that there is room for further advancement in improving both experimental and computational technologies underlying the approach.

Acknowledgments

I cannot possibly find enough words to express my gratefulness to my advisor, Dr. Russell Schwartz, who is truly the best type of advisors a student like me can ever wish for. He provides insightful guidance to his students yet leaves enough free time for them to think by their own; he works hard for the better being of research yet asks for little commitment from the people under his supervision; he brings comfort and ease to the lab yet keeps a pure professional relationship to everyone at work; he ensures the students are on the right course to graduation yet never criticizes them with strong words when they make stupid moves. During my training in his lab, he has been carrying out the duty of an advisor to the fullest extent and escorting us through the somewhat rough terrain of academia to reach our highest achievement.

The completion of this thesis is impossible without the contributions from the committee members. I would like to thank Drs. James Faeder, Nick Sahinidis and Fred Homa and for their advice on the direction of research, optimization algorithms, virological insights, enrichment of the content of this thesis, and so much more. Special thanks to Dr. Faeder for advising my third rotation and introducing me to the world of rule-based modeling. The research of this thesis is also made possible by Drs. David Wu and Adam Zlotnick who provided the experimental data.

I would express my deepest sorrow to the loss of Dr. Joel Stiles, who advised my second rotation and opened the gate of MCell to me. He left us too early before I could have absorbed more knowledge from his wisdom.

I would like to thank the past and current members of the Schwartz Lab, a nice bunch who make my graduate life smooth and easy. Among them, I give my gratitude to Greg Smith, who has been sitting next to me for five years and broadening my horizon of entertainment. We have also collaborated on the virus capsid assembly project and his focus was on predicting *in vivo* assembly and visualizing the assembly progress. This project, where my thesis is based on, has been a collective work of Dr. Tiequan Zhang, Blake Sweeney and Rori Rolfs who wrote the original simulator and initiated the project, Rupinder Khandpur and Xian Feng who improved the simulator and made the movie of HPV assembly, Dr. Byoungkoo Lee who implemented the experiments on crowding effects, Senthil Kumar who pioneered on parameter inference, and the other contributors whose credits I failed to straighten out. I give my best wishes to Thomas Marus and Tingting Xu, the two new labmates who have been carrying on this project into the future. The pursuing of science became less monotonic with the promotions of high-tech gadgets and gizmos brought up by Drs. Ming-Chi Tsai and John Kang, the promotions of healthy lifestyle advocated by Dr. Ayshwarya Subramanian, the promotions of athletic activities brought in by Theodore Roman and Dr. Salim Chowdhary, and the constant quest for better coffee and cookies accompanied by two honorary lab members, Dr. Minli Xu and Han Lai.

I owe my appreciation to the CPCB program and the Department of Computational Biology for my predoctoral training. The thriving of the CPCB program and its members accredits to the leadership of its former and current directors, Drs. Robert Murphy, Ivet Bahar, Russell Schwartz, Panayiotis Benos, Daniel Zuckerman and James Faeder, and the team work of its former and current administrators, Thom Gulish, Maureen Hernandez, Nichole Merritt, Kelly Gentile, Dr. Karen Thickman and Dr. Joseph Ayoob. The Department of Computational Biology partially funded the thesis research, and maintained the computer cluster where the computational experiments are executed on.

Over the past few years I have developed emotional attachments to Pauline, a ten-year-old Power Mac who served as my main workstation for three years and half and died in 2014. Her legacy is inherited by the approximately same-aged Proline who processed my images, prepared my slides and compiled my thesis. Personal feelings made me unable to replace them with the new computers my advisor acquired for the lab.

Most importantly, I could never accomplish the predoctoral course without the selfless love, support and sacrifice from my family. My first-love and wife, Wen Gao, has been my loyal and sole companion ever since we left all the rest of our families and stepped onto the new hemisphere. Her devotion fills our dwelling with warmth and care, her spurring motivates me to keep aiming for higher, and her optimism renders the sky of the Steel City less cloudier.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Impact on self-assembly systems

Self-assembly is a ubiquitous phenomenon in the organization of molecules in living systems. A considerable portion of proteins in eukaryotic cells normally function as parts of molecular complexes [1] and almost all critical functions to a cell's life —- including signal transduction, the assembly and degradation of proteins and nucleic acids, cell movement and morphology — depend on self-assembly of some specialized molecular structures, complexes or machinaries. Developing accurate, quantitative models of self-assembly processes is therefore essential to the overall mission of comprehending complex biological systems.

A particularly important example of self-assembly system is the viral capsid self-assembly process, which has proven a fertile field for modeling complex assembly systems. As an essential step of the life cycle of many virus species, capsid assembly may reveal potential targets for anti-viral treatment [2, 3, 4, 5] as well as potential vehicles for gene therapy. The understanding of the mechanism of viral capsid assembly may also enlighten the study of non-organic self-assembly structures.

Despite the importance of virus self-assembly and other complex assembly processes, our understanding of the detailed steps of complex assembly reactions remains primitive, at least at the quantitative level. This limitation is largely due to technological reasons: we currently have no experimental technology that can monitor particle-by-particle assembly of a small, rapidly building structure such as a virus. Existing methods for characterizing the structure of such systems at fine detail rely on averaging over large numbers of particles (e.g., crystallography and cryo-electron microscopy), making them unsuitable for monitoring step-by-step process of a stochastic system, such as virus assembly. Methods for monitoring assembly kinetics at fine temporal scales so far provide only indirect, bulk measures of assembly progress, again making them unsuitable for inferring specific assembly pathways of individual viruses.

This thesis is aimed at revealing possible assembly pathways of virus self-assembly systems with the assistance of computational simulations and optimization methods. The approaches developed in this thesis, in theory, will work with any self-assembly systems that have accessible data on bulk measurements and coarse-grained conformations of basic building blocks.

1.2 Previous studies

Simulation-based methods have long played an key role in studying the detailed dynamics of molecular interaction systems [6, 7], especially where direct experimental observation is infeasible. For example, simulation methods have made it possible to infer various emergent properties of hypothetical assembly models [8, 9, 10, 12], to explore the effects of perturbations in parameter spaces [10, 13, 14, 15], and to examine possible assembly pathways and mechanisms accessible to theoretical models [13, 16, 17, 18, 19]. Systems biologists have developed ever more complex and comprehensive models of biological systems [22], culminating in such recent landmarks as the simulation of whole cells [23]. Explicit models of self-assembly reaction networks, however, have been largely neglected from such efforts to build predictive simulation models. This gap may reflect both the computational difficulty of handling the enormously complex networks of possible reactions produced by even simple assembly processes and the experimental difficulty of precisely measuring the kinetic parameters of any non-trivial molecular assembly process to properly instantiate simulations.

Due to the large size, long time scales, and enormous space of possible pathways a large assembly might pursue [18], deterministic fine-grained simulators such as all-atom molecular dynamics might fail, and successful simulators require significant coarse-graining [20, 21]. Prevailing methods accomplish this by using “local rule” models [9, 11, 33, 34], which concisely represent a system in terms of simplified assembly subunits with sets of discrete binding sites. Local rules provide a concise way to implicitly represent a potentially enormous ensemble of possible reaction trajectories by providing an efficient way to enumerate possible reactions accessible to a system from any starting state. Such local rule binding models can be combined with Brownian dynamics models [10] and/or stochastic simulation algorithm (SSA) models [31] to yield computationally tractable simulations of the assembly of potentially thousands of subunits into icosahedral capsid structures. Nonetheless, these methods have not traditionally been able to provide detailed quantitative models of specific capsid assemblies because they depend on detailed interaction parameters that we currently cannot measure experimentally.

Researchers have made several attempts to improve the accuracy of estimating kinetic parameters, such as using simulation models to find rate ranges plausible for productive assembly [24, 25], using analytical fits of small numbers of parameters to match to light-scattering data [26], applying additional constraints on possible assembly pathways such as those imposed by virus-specific nucleic acid interactions [27], inferring approximate averaged rates from global estimates of the free energy of assembly [26], inferring approximate rate parameters from structural models [28, 29], or scanning parameter spaces to put constraints on the range of possible behaviors of a viral system [10, 12, 14, 15, 19, 30, 31]. Precise rate constants are not known for any real viral system, however, and previous simulation studies have suggested that assembly mechanisms can be highly sensitive to changes in these and other parameters [19], casting doubt on the ability of any such approximations to yield faithful reproductions of real assembly behavior.

Model-fitting methods provide a potential solution to this problem by allowing one to learn experimentally unobservable parameters by fitting simulations to indirect experimental measures of biological systems [32, 35, 36, 37]. Model-fitting in a computational or mathematical model is typically treated as an optimization problem over the parameter space to minimize deviation between the model and some measurable behavior of the real system. Simulation-based model-fitting has proven effective for a variety of simpler network models in biology [35, 36, 37, 38] and has previously been combined with rule-based modeling for systems that face similar problems of combinatorial blowup in pathway space to the capsid system [32, 39, 40, 41, 42, 43, 44, 45]. In general, stochastic models have long been assisting the research of estimating parameters of biological and biochemical systems [46, 47, 48, 49]. We have previously show that the SSA-based approach to local rule-based capsid assembly modeling is particularly amenable to data fitting because it greatly accelerates simulation relative to the more involved Brownian models by avoiding explicit simulations of particle diffusion, while simultaneously reducing the parameter space to a small number of kinetic parameters [31, 50, 51] that are nonetheless sufficient to capture many potential ensembles of assembly trajectories [12, 19].

1.3 Parameter inference framework

The main contribution of this thesis is the development of a parameter inference framework that implements and generalizes the idea of learning kinetic rates of self-assembly models. The framework is built upon an existing coarse-grained rule-based viral assembly model and stochastic simulator [31], and made adaptable to different optimization algorithms and data sources. The major functional elements of this approach and their interactions are shown in Figure 1.1, and the remaining chapters of this thesis will be devoted to explain these function blocks.

Chapter 2 will describe the viral capsid models used in this thesis and the functionality of the self-assembly simulator. The simulator takes capsid models and kinetic parameters as input, and produces temporal status description of the assembly progress. The output can be customized and

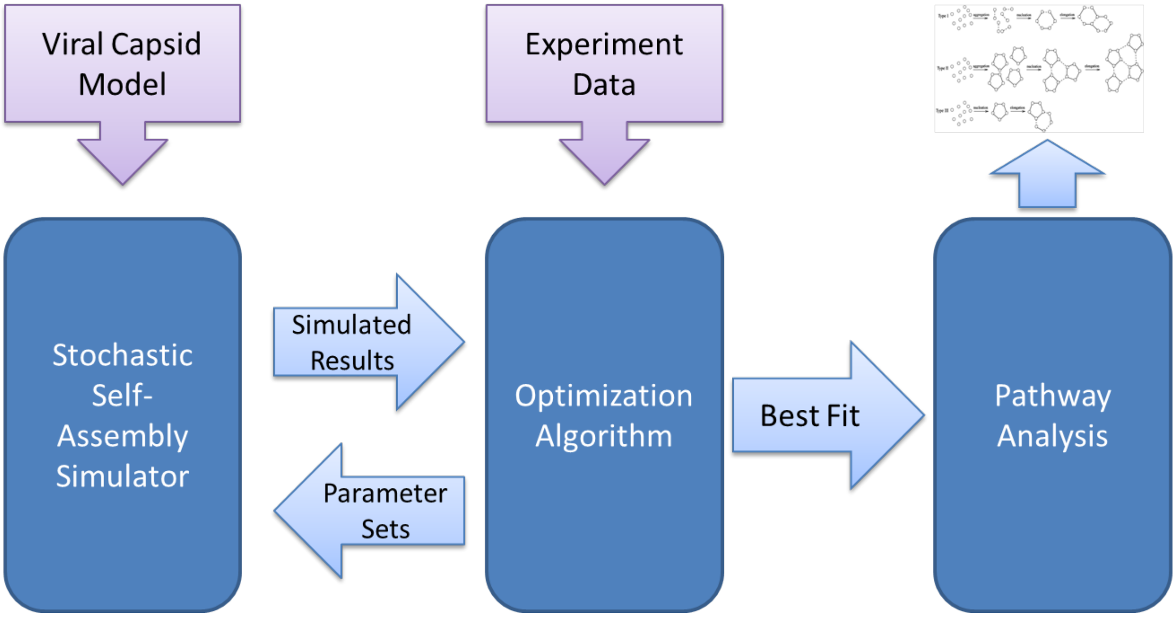

Figure 1.1: Parameter inference framework

processed into a comparable form with the experimentally measured data. As an important step of implementation, the work flow of parallelizing and managing the simulations on a computer cluster will also be explained in this chapter.

Chapter 3 will describe the optimization algorithms that have been tested inside the framework. The optimization algorithms evaluate the deviation between the experimental data and outputs from the simulator, and generate new sets of parameters to be sent to the simulator. The iterative interaction between the two modules will terminate once the algorithms decide that the output from the simulator is close enough to the experimental data.

Chapter 4 will describe the application of the framework onto experimentally measured static light scattering (SLS) data for three viral systems: HPV, HBV and CCMV. The bestfits, number of evaluated parameter sets and final sets of parameters will be compared across different algorithms.

Chapter 5 will extend the application of the framework onto synthetic datasets including SLS data and non-covalent mass spectrometry (NCMS) data. Detailed comparisons will be shown with respect to the deviation from true parameters and the accuracy of making predictions.

Chapter 6 will describe the methods for analyzing possible assembly pathways from the best-fitting trajectories. The results show distinct types of assembly pathway for structurely similar virus species. More results will be provided regarding prediction of *in vivo* assembly based on the inferred *in vivo* parameters and assumptions on *in vitro* environments.

Chapter 7 will summarize the work of the thesis, discuss the advantages and disadvantages of current framework, and provide possible directions for future developments.

Chapter 2

Modeling and Simulation*

2.1 Coarse-grained capsid model

Regardless of their species and structual complexity, all viruses have at least one layer of protein coat that encapsulates and protects their nucleic acid [59]. Due to the limits of viral genome capacity, this protein coat, or capsid, is generally formed in a highly symmetric helical or icosahedral structure by many identical copies of a small number of viral proteins [59]. In this section I will describe the computational models of some virus capsids and their components used in this thesis.

The quality of the virus capsid model that best serves the scope of this thesis is determined by two factors: simulation time cost and the necessity for reproducing experimental measurements. Equipped with state-of-the-art hardware and appropriate approximation algorithms, molecular dynamics nowadays can stretch the length of all-atom simulation for full virus capsid to the scale of milliseconds and provide in depth insights for the viral physiologies [52, 53, 54, 55], however, the duration of such large scale simulations is still way too short for fitting *in vitro* assembly processes that may span minutes to hours. Pure numeric simulations that only track the sizes of

*The capsid model and simulation tools described in this chapter are based on the local rules theory by Berger et al. [9], the simulation work by Schwartz et al. [10, 11, 33], and the discrete event simulator for self-assembly (DESSA) developed by Zhang et al. [31, 58] and later improved by Kandapur and Feng.

intermediate assemblies would result in much faster simulation, but interpreting certain types of experimental measurements, such as dynamic light scattering (DLS) [56] and small angle X-ray scattering (SAXS) [57], requires the information of relative geometric positions of the subunits in the intermediates. To balance the time cost and fineness requirements, the capsid model must be simple enough for efficient computations yet complex enough to carry geometric information.

The basic unit in our virus capsid model is called ”subunit”, which may represent a single coat-protein or coat-protein oligomers, depending on the experimental condition where the data is collected. A subunit interacts with its neighboring subunits through binding sites, and each binding site has a binding rate (on-rate), a breaking rate (off-rate), and a binding partner which is another binding site of a neighboring subunit. The subunit types, binding site types and their geometric coordination are extracted from real virus capsid structure (see an example in Figure 2.1).

A subunit model contains the following information:

- The relative geometric coordinates of its binding sites.
- The partner type of each binding site.
- The on-/off-rate of each binding site.

The on- and off-rates are the parameters that we sought to infer by fitting experimental data. A capsid model consists of all types of its subunit models. Due to its highly symmetric structure, a capsid model may only have a few types of subunits. In this thesis, three capsid models are used: HPV, HBV and CCMV. Their capsid models and corresponding complete capsids are shown in Figure 2.2, and their properities are listed in Table 2.1.

2.2 Simulation methods

The use of rule-based coarse-grained capsid models for dynamic assembly simulation was first introduced in combination with Brownian dynamics simulation [9, 11], and has come to be the

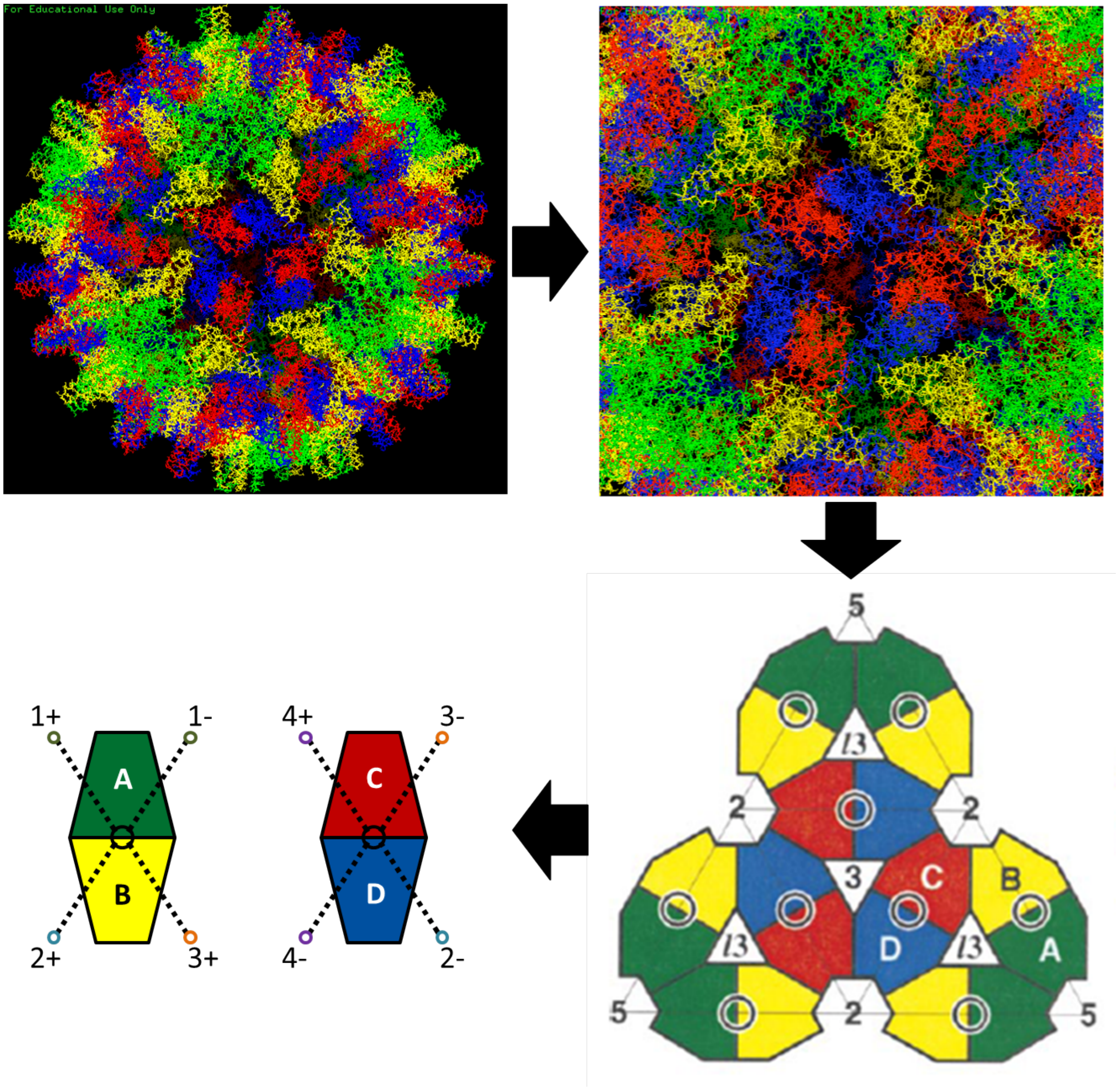

Figure 2.1: Extracting capsid and subunit model from real virus capsid structure. Top left: HBV capsid structure [60] shown in PyMol. Top right: local symmetry of HBV capsid. Bottom right: coarse-grained subunit structure preserving local symmetry [61]. Bottom left: subunit types and their binding sites.

predominent approach for modeling work in this field. In the process of simulation, subunits diffuse and collide following the principles of Brownian motion, and the binding dynamics are quantified with a transition state energetic model when a collision happens. Such model is a significant reduction from molecular dynamics, yet still able to lead to numerous discoveries of successful and unsuccessful assembly mechanisms that became guiding principles of this field.

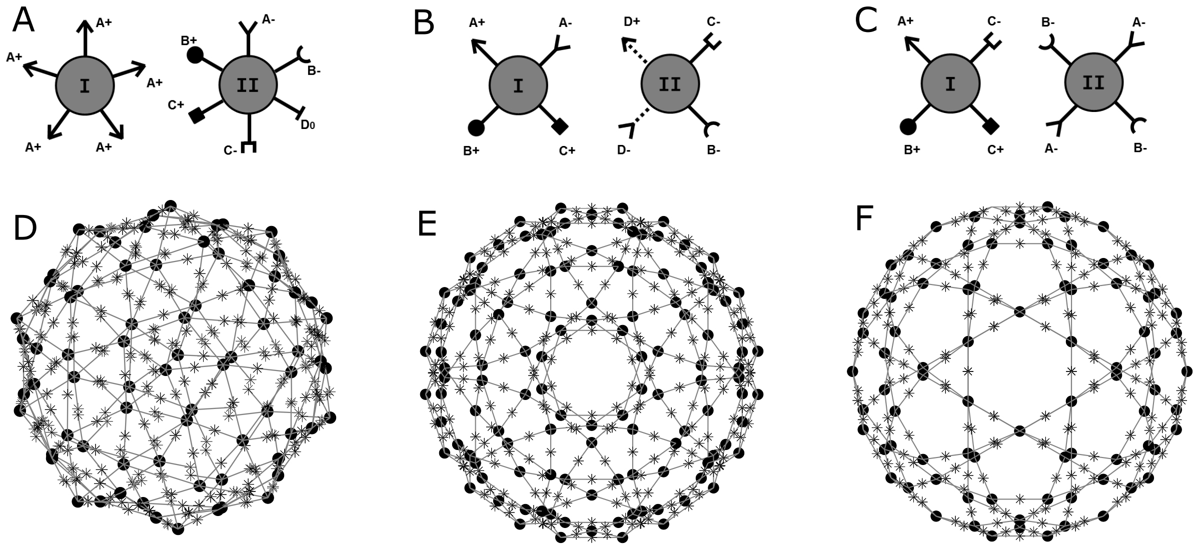

Figure 2.2: Capsid models (A, B, C) and corresponding complete capsids (D, E, F) for HPV (A, D), HBV (B, E) and CCMV (C,F). Rules for binding are shown in Table 2.1. The black dots in the complete capsids represent subunits, and the asterisks represent binding sites that are linked to their belonging subunit by the solid lines.

Such principles include the tradeoff between growth rate and yield due to kinetic trapping [10], the switch in conformation to avoid these traps [11], the need for weak binding to promote assembly efficiency [10, 11], and the potential for exploiting such kinetic effects by capsid-targeted anti-viral strategies [33]. Despite being adopted by many research groups [12, 13, 15] and used for many similar approaches [62, 63, 64, 65, 66], the Brownian dynamics simulation has its own shortcomings that render it incapable of quantitative parameter search. First, the simulation is too slow to be practical for the large number of trajectories required by parameter search. Second, it depends on excessive numbers of physical parameters that cannot be easily linked to experimentally measureable properties of subunits.

Another popular approach to investigate virus capsid assembly is numerically solving a set of ordinary differential equations (ODEs) that describes all the possible reactions among intermediate assembly species obeying the law of mass action. It is a deterministic approach based on the assumptions of reactant abundance and well-mixed environment. It has been used to analyze the assembly of a relatively small dodecahedron virus model that builds from 12 pentamers

Table 2.1: Properties of capsid models

Virus	# of subunit types	Subunit size	# of subunits in a capsid	Binding rules
HPV	2	Pentamer	72	A+ binds A- B+ binds B- C+ binds C- Do binds Do
HBV	2	Dimer	120	A+ binds A- B+ binds B- C+ binds C- D+ binds D-
CCMV	2	Dimer	90	A+ binds A- B+ binds B- C+ binds C-

[8, 26, 30]. Described by 12 cascading ODE equations, this method successfully recapitulates experimentally observed sigmoidal kinetics and explains the lag phase and equilibrium behavior. Stepping up to more complex models, however, the ODE-based model faces a substantial obstacle of combinatorial explosion in complexity as the number of intermediate species and possible reactions grows exponetially with respect to the size of capsid [18, 19]. Furthermore, the fundamental assumptions of ODE-based model may not hold for *in vivo* environments, which impedes the transistion from *in vitro* inference to *in vivo* prediction.

To address the issues of time, space and parameter complexity, Jamalyaria et al.[50] and Zhang et al.[31] developed a stochastic approximation simulation algorithm that is tailored for large scale self-assembly systems. The resulting simulator, named DESSA, reads the geometric and kinetic rules from the capsid model, and creates an event queue of binding and breaking events whose firing times are randomly drawn according to the Gillespie algorithm [67]. The simulator will continue to add, execute or invalidate the events until there is no more event or the maximum simulation time is reached. When a binding event is executed, DESSA will align the two assemblies along the binding site and perform a rotation to put the assemblies in the right positions on a capsid. Like ODE models, DESSA does not consider particle diffusion and assumes well-mixed environment. The output of DESSA can be customizedin various ways. For the particular version (1.5.8) of DESSA used in the present parameter inference work, the output is a text file containing a temporal list of counts of all intermediate assemblies. The source and usage of DESSA 1.5.8 are shown in Appendix A.

2.3 Deployment on computer cluster

One challenge to the task of parameter learning via stochastic simulation is the uncertainty brought by stochastic noise. There are two possible ways to suppress stochasticity: performing simulation with more subunits or averaging the trajectories of many replica simulations. The second solution is more favorable for parallelized execution on computer clusters. The parameter learning task was performed on the Lane cluster administrated by Computational Biology Department, School of Computer Science, Carnegie Mellon University. The cluster provides several queues, each of which has around a hundred processor cores, for public and private submission, and each submitted simulation task is regarded as a job. The jobs on the cluster are managed by the terascale open-source resource and queue manager (TORQUE). Direct submission of thousands of jobs at once may put too much stress on the cluster, so I have written a single-threaded script in MATLAB to submit and monitor the jobs and import their text-based outputs to MATLAB. The script works via a polling loop. Its flowchart is shown in Figure 2.3. The script was designed to accommodate several design constraints:

- It is written in MATLAB due to closer integration to the optimizaion part.
- It should reside on a compute-node, not head-node, of the cluster. Single-threaded execution helps reduce the resource consumption.
- A job is considered ”active” after being submitted until termination, regardless of its status on the queue.
- It is able to limit the number of active jobs on each queue. A “full queue” may refer to the situation that the maximum number of active jobs is reached, while the queue itself may

still have open slots.

- It is able to do on-line adjustments of the job limits of the queues according to their crowdness. Future scripts might add automatic “load-balancing” function.
- It is robust against several exceptions, such as network interruption, non-residing compute-node crash, and corrupted file transfer.
- It is able to set a waiting time between queue queries to avoid stressing TORQUE.

Deployment is tightly coupled with the parameter inference algorithms by feeding them with trajectories required for computing objective values with respect to given sets of parameters. Because the most time-consuming step is carrying out the JAVA-coded simulations, scripting the deployment in MATLAB might not create a concern of cost of time, but it certainly would be more convenient if the system were ported to a light-weight and free scripting platform, such as Python or Shell.

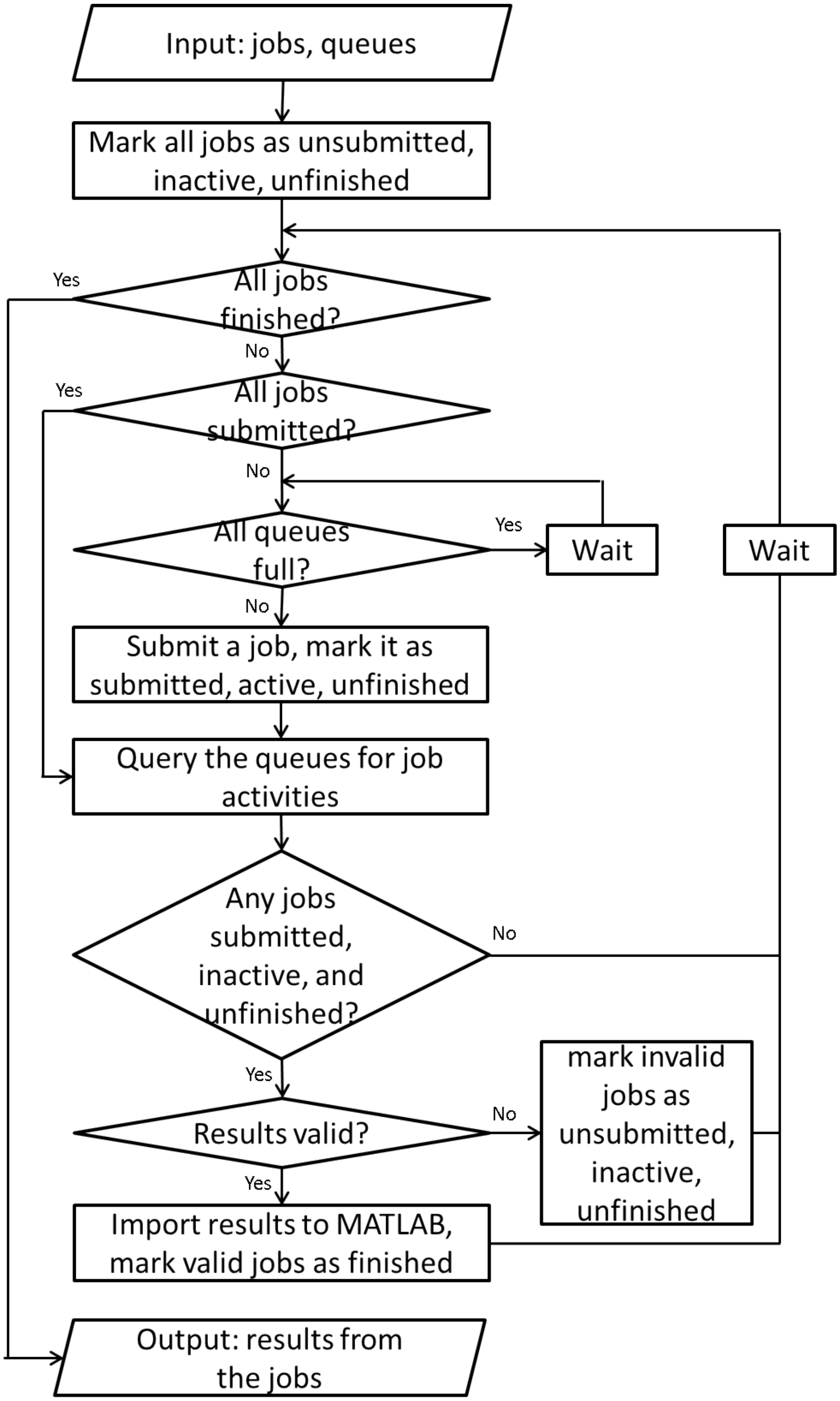
Input: jobs, queues
Mark all jobs as unsubmitted, inactive, unfinished
All jobs finished?
Yes
No
All jobs submitted?
Yes
No
All queues full?
Yes
Wait
Wait
No
Submit a job, mark it as submitted, active, unfinished
Query the queues for job activities
Any jobs submitted, inactive, and unfinished?
No
Yes
Results valid?
No
mark invalid jobs as unsubmitted, inactive, unfinished
Yes
Import results to MATLAB, mark valid jobs as finished
Output: results from the jobs

Figure 2.3: Flowchart of job submission, management, and result collection on TORQUE-managed cluster.

Chapter 3

Parameter Inference Algorithms[*]

3.1 System-specific challenges

In general, the parameter inference task for a computational or mathematical model of a system of interest is posed as an optimization problem. The goal of this optimization is to minimize an objective function $f(\vec{x})$ that measures deviation between simulated and real data with respect to the parameter vector $\vec{x}$ in the parameter space. While a variety of generic optimization methods can in principle be applied to the problems under this broad definition, the choice of appropriate methods for any particular system will depend on many specific characteristics of the system to be fit. Virus assembly systems present several special challenges to the optimization approaches to parameter inference.

One specific characteristic of virus assembly system is the lack of closed-form expressions for non-trivial models of capsid assembly. Even for a relatively small dodecahedron capsid model with 12 subunits and 12 ODEs seemed impractical to be solved analytically [8, 58]. As a result of that, the parameter fitting problem falls under the class of simulation optimization, where the objective function needs to be evaluated through one or several simulation runs [68].

[*] The description of Kumar method is mainly based on the published work of Kumar et al. [32] and Xie et al. [70], and the description of DFO methods is mainly based on the submitted work of Xie et al. [71].

Another striking obstacle is the computational cost of assembly simulations, which can take from minutes to hours for the production of a single trajectory, depending on the capsid models and parameter values. This is largely because of an extensive amount of trial-and-error involved in nucleation-limited growth processes characteristic of virus assembly. A single data fitting task may require sampling up to hundreds of thousands of these trajectories, which may take weeks to months to accomplish on a computer cluster with around 100 Intel Nehalem cores.

An even bigger obstacle is the high stochastic noise, a feature inherent to the stochastic modeling method described in Chapter 2. Traditional numerical optimization done in optimizing quality of fit of a parameter set is accomplished by methods such as gradient descent, Newton-Raphson, or Levenberg-Marquardt that depend on taking derivatives of the deviation between real and simulated data [69]. Stochastic noise in the simulated data results in discontinuities in derivatives, a significant problem for these methods.

To address the above chellenges, Kumar et al. [32] developed a heuristic global/local search algorithm, upon which Xie et al. [70, 71] later improved. Section 3.3 introduces the use of two derivative free optimization (DFO) methods: multi-level coordinate search (MCS) [72] and stable noisy optimization by branch and fit (SNOBFIT) [73].

3.2 Heuristic global/local search scheme (Kumar method)

A remarkable amount of research has been done on designing methods for optimization of stochastic simulation systems [68, 74, 75, 76, 77]. The embedded noise in such systems may introduce errors to gradient estimation. As a result of that issue, some optimization algorithms for stochastic systems, such as quasi-gradient methods and algorithms of type Kiefer-Wolfowitz [78], avoid directly dealing with the gradient of the objective function. The other techniques that have been developed to approximate gradients in these systems, such as specialized finite difference schemes and infinitesimal perturbation analysis, may impose restrictive conditions on the form of the potential surface to be fit [32]. A different yet important class of method is the response surface approach, which fits a smoothed regression model to the potential surface and optimizes with respect to the regression model. The minimum of the regression model is then regarded to be the estimate of the minimum of the objective function in the parameter space [76]. Though they require fewer simulation runs than gradient-based methods, response surface methods can perform poorly under certain conditions, such as poor approximation of the meta-model of the search space, inadequate sampling in the search space, or a search space that is characterized by very sharp ridges and/or large valleys with close to zero curvature [76, 79].

Kumar et al. developed a heuristic global/local optimization strategy for parameter inference of capsid assembly systems to deal with the particular computational challenges introduced by stochastic simulations [32]. In the local optimization part, the method interpolates between response surface and quasi-gradient approximations to provide both fast handling of smooth regions and robust handling of more difficult regions of the objective function. In the improved version of Kumar method [70], the local optimization algorithm proceeds in five steps:

1. Given a grid size s, the algorithm picks a set of vectors surrounding the current parameter $\vec{\mathbf{x}}$ in the parameter space, then conducts simulations and collects objective function values on these vectors. The set of vectors are chosen in an arbitrary way to provide enough samples for fitting a quadratic response surface, while avoiding exponential blow-up: 1) +/- each individual element by s while keep the rest stationary; 2) + +/+ -/- +/- - each pair of elements by s while keep the rest stationary. This results in $2n^2$ vectors for n parameters.
2. The algorithm fits the vectors including $\vec{\mathbf{x}}$ and their objective function values with the following quadratic function:

$$f(\vec{x}) = c_{00} + \sum_{i=1}^{n} c_{0i}x_i + \sum_{i=1}^{n}\sum_{j=i}^{n} c_{ij}x_i x_j \tag{3.1}$$

Where x_i's are the elements in the parameter vector and $f(\vec{x})$ is the corresponding objective function value. The coefficients, c, are obtained by calling *nlinfit* in MATLAB, and

$\vec{x}_{RS}$, the minimum of $f(\vec{x})$ in the grid of $-2s \preceq \vec{x} \preceq 2s$ is found by calling *fmincon* in MATLAB.

3. The algorithm evaluates the gradient of $f(\vec{x})$ by using a subset of the samples and their objective function values:

$$\nabla f(x_i) = \frac{f(<\vec{\mathbf{x}}_1, \cdots, \vec{\mathbf{x}}_i + s, \cdots, \vec{\mathbf{x}}_n >) - f(<\vec{\mathbf{x}}_1, \cdots, \vec{\mathbf{x}}_i - s, \cdots, \vec{\mathbf{x}}_n >)}{2s} \tag{3.2}$$

4. The algorithm selects a new candidate for the minimum of the objective function by an interpolation between response surface model and gradient descent:

$$\vec{x}_{new} = \frac{\vec{x}_{RS} + r(\vec{\mathbf{x}} - s\nabla f/\|\nabla f\|)}{r+1} \tag{3.3}$$

   Here, r serves as a bias factor, which will be discussed in the global search part of this section.

5. The algorithm evaluates the objective function value at $\vec{x}_{new}$.

The steps of the local search are illustrated in Figure 3.1.

The heurisitic global search is posed on top of the local search scheme. If indeed $\vec{x}_{new}$ yields a lower objective value, in which case the search sets $\vec{\mathbf{x}} \leftarrow \vec{x}_{new}$, $s \leftarrow 2s$, $r \leftarrow r/2$. Otherwise, it disregards $\vec{x}_{new}$ and let $s \leftarrow s/2$, $r \leftarrow 2r$. The underlying idea of this heuristic global search scheme is to expand the search region and put more bias towards the prediction of response surface once the objective value is improved, and vice versa. This approach relies on a similar intuition to the standard Levenberg-Marquardt optimization method [80, 81] which interpolates between two forms of fit to empirically adjust between algorithms more suitable to handle smooth or rough regions of the parameter space. The search terminates when the predefined minimum grid size is reached.

Based on the original Kumar method [32], Xie et al. have made the following improvements [70]:

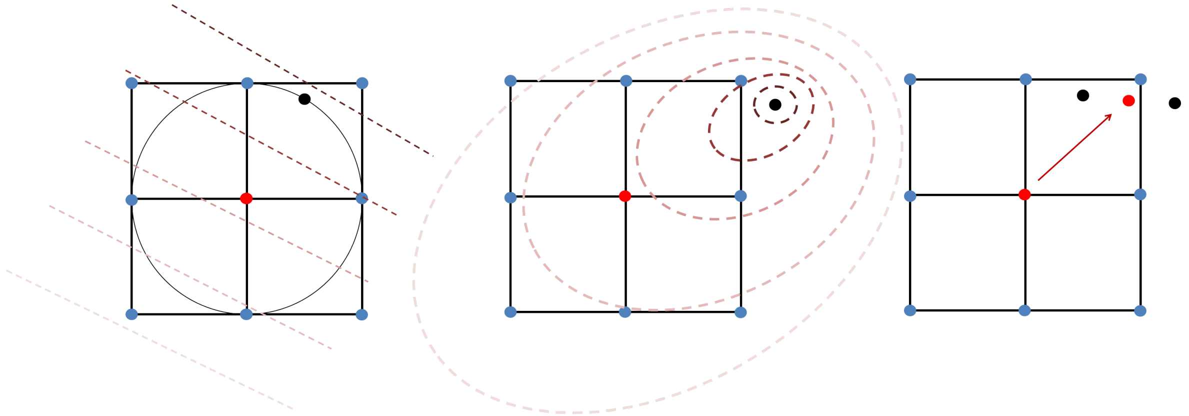

Figure 3.1: Illustration for the local search steps in Kumar method with 2 parameters. Left: estimatiion of gradient. Red dot is the current parameter set, and the blue dots are samples. The darker lines indicate the descending direction of the gradient. The black dot is the candidate predicted by gradient descent. Middle: fitting quadratic response surface model. Right: interpolation between response surface model and gradient descent. The new red dot represents the new candidate for minimum via an approximate $r = 1$ interpolation.

- Automated the choice of samples and quadratic functions for any number of parameters, while in the original method the samples and functions were chosen on a case-by-case basis.
- Decoupled the grid size s and bias factor r, which are controlled by a single factor in the original method. This increased the flexibility in making adjustments and adaptiveness to different profiles of parameter space.
- Put more emphasis on evaulating the new candidates. As the direction of the global search is driven by the objective value of the new candidate, the noise-induced inaccuracy in evaluating new candidates will dramatically mislead the direction of the global search.

The Kumar method is integrated into the parameter inference framework as shown in Figure 3.2. Note that the Kumar method is not a system-specific but a generic solver, and it can be applied to any optimization problem with similar concerns of stochasticity and time complexity. The applications of the Kumar method will be shown in the Chapters 4 and 5.

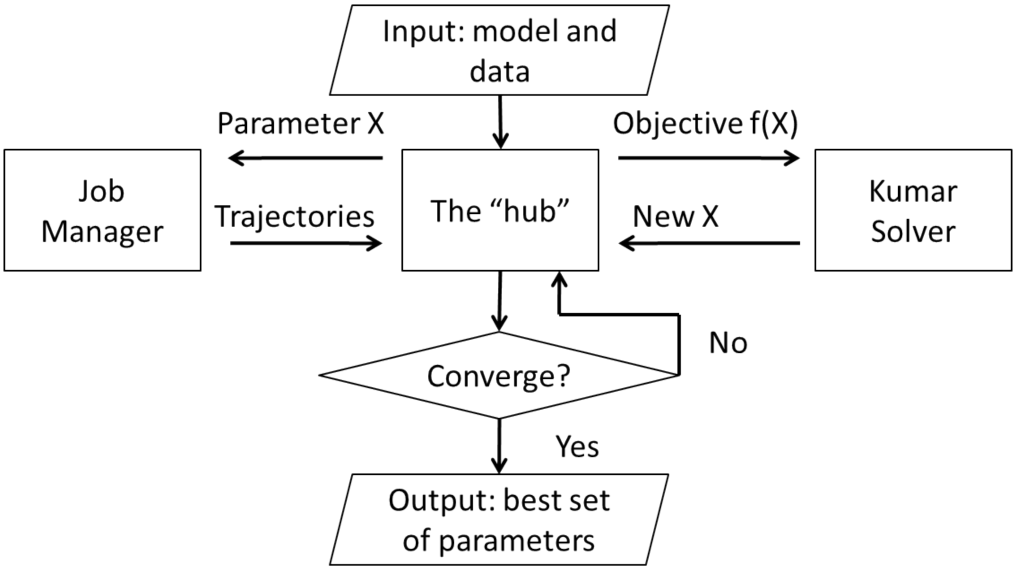

Figure 3.2: The flowchart showing how Kumar method is integrated into the parameter inference framework. The hub prepares jobs for submission to the queue, and translates trajectories into objective function values. The job manager is decribed in Figure 2.3.

3.3 Derivative-free optimization (DFO) methods

As a heuritic algorithm incubated with the consideration of handling stochastic noise, the Kumar method is, nevertheless, partially dependent on the numerical estimation of the first derivative of the objective function, and its quadratic response surface model might be somewhat naïve. A big leap forward would be using optimization methods that completely eliminate the use of derivatives, which motivates the choice of employing DFO methods. As the name suggests, the class of DFO methods avoid computation of derivatives of objective function, making them in principle less susceptible to stochastic noise than are gradient-based methods. DFO methods in general tend to be well suited to systems such as stochastic capsid assembly, that are characterized by high noisy and high computational cost for evaluating the objective function. Rios et al. have published a review of 22 DFO algorithms and their performance comparison on 502 test problems, and found MCS to be the best among the freely available solvers [82]. As a good start to explore the potential of DFO methods for this problem, MCS was chosen as an alternative optimization algorithm in the parameter inference framework. Xie et al. also choose SNOBFIT as an alternative method because it is specifically designed for optimizing noisy systems. They are both global optimization solvers but with different approaches: MCS searches for the global optimum via bisecting boxed regions in the parameter space [72], and SNOBFIT approximates the objective function by using more sophisticated surrogate functions [73].

The integration of the two methods into the parameter inference framework is shown in Figure 3.3. SNOBFIT functions in a similar fashion to the Kumar method, and can be regarded as a black box solver and direct substitute for the Kumar solver. MCS functions as the driver of the optimization progress, and it treats the virus capsid assembly system as a black box optimization problem. The applications of the DFO methods will be shown in the Chapters 4 and 5.

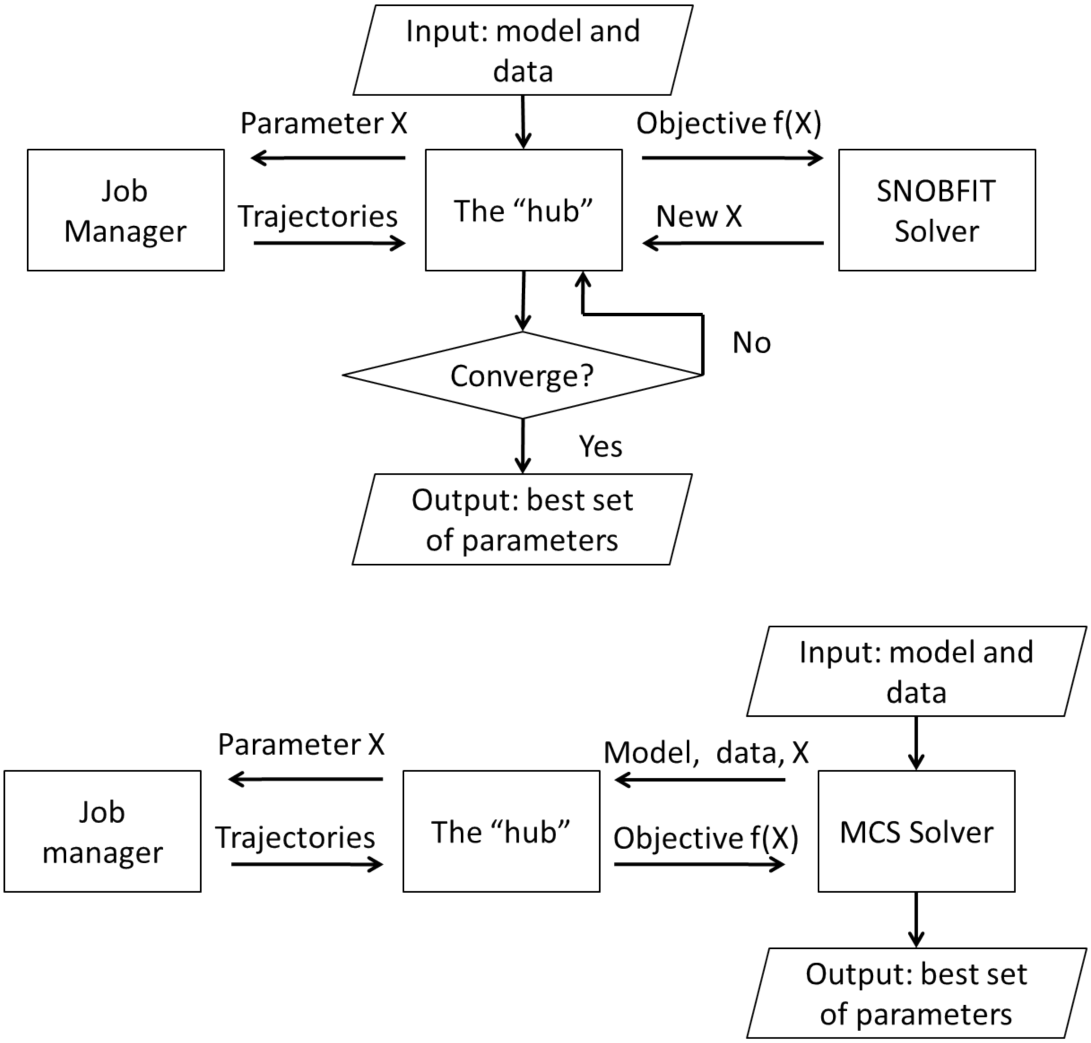

Figure 3.3: The flowchart showing how SNOBFIT (top) and MCS (bottom) are integrated into the parameter inference framework. The functions of the hub and job manager refer to Figure 3.2 and Figure 2.3, respectively.

Chapter 4

Application on Experimental Datasets*

4.1 Sources of datasets

Parameter inference has been conducted on experimental datasets of human papillomavirus (HPV), hepatitis B virus (HBV) and cowpea chlorotic mottle virus (CCMV). Datasets for the three systems were gathered from prior studies by Casini et al. [83] (HPV), Zlotnick et al. [26] (HBV), and Zlotnick et al. [84] (CCMV). In each case, the source of data collection is 90° static light-scattering (SLS) measurements of temporal evolution of *in vitro* capsid assembly systems. The SLS measures the intensity of light scattered by the coat protein solution in arbitrary unit (a.u.), which reflects the turbidity of the solution.

For HPV, light-scattering data was gathered from purified L1 coat protein capsomers in citrate buffer with 0.5 M NaCl at pH 5.20 for 250 minutes per experiment [83]. The L1 protein was expressed in *E. coli* [83]. The data was provided directly by David Wu (Department of Chemical and Biological Engineering, Colorado School of Mines) in electronic format. The parameters are inferred from fitting three curves corresponding to capsomer concentrations of 0.53, 0.72, and 0.80 µM.

*The content of this chapter is mainly based on the published work of Xie et al. [70], and the DFO application part is mainly based on the submitted work of Xie et al. [71].

For HBV, data was gathered from stock solutions of Cp149 coat protein dimers in 0.1 M sodium bicarbonate and 5 mM DTT at pH9.5 for 600 seconds per experiment [26]. The Cp149 protein is truncated from the full length Cp183 protein by discarding the 34 nucleic acid binding residues on the C-terminus [85]. The data points are derived from the appropriate figures in the reference (Figure 4C in Zlotnick et al. [26]). The parameters are inferred from fitting coat dimer concentrations of 5.4, 8.2, and 10.8 µM. The beginning stages of the light scattering curves of the three concentrations (the first 53.7, 26.6 and 10.3 seconds, respectively) are smoothed by numerical average in order to alleviate the influence of instrument noise and the resolution limit of the print media.

For CCMV, data was gathered from solutions of coat protein dimers in 200 mM sodium citrate and 1M NaCl for 300 seconds per experiment [84]. The coat protein dimers are collected via dissociation of viral capsids separated from affected cowpeas [86]. The data points are derived from the appropriate figures in the reference (Figure 1B in Zlotnick et al. [84]). The parameters are inferred from fitting coat dimer concentrations of 14.1, 15.6, and 18.75 µM. An artificial 2.5 second lag phase was added for each curve to account for the timing uncertainty at the beginning of assembly, an important issue for CCMV because of its comparatively more rapid initiation than the other systems.

4.2 Objective function

The objective function measures the quality of fit, but the output from the simulator must be converted in advance to a form that is comparable with the experimental measure. As described in Section 2.2, DESSA 1.5.8 produces counts of intermediate assemblies of each size as a function of time in the simulation. Following Casini et al. [83] we can approximate the SLS curve produced by any given parameter vector $\vec{x}$ over time t as follows:

$$R(t,\vec{x}) = \frac{k \times c \times \sum_{i=1}^{n} \big(N_i(t,\vec{x}) \times i^2\big)}{\sum_{i=1}^{n} \big(N_i(t,\vec{x}) \times i\big)} = k \times c \times S(t,\vec{x}) \tag{4.1}$$

Here $R(t,\vec{x})$ is the value of a simulated SLS curve at time point t with parameter vector $\vec{x}$, c is the concentration of subunits, $N_i(t,\vec{x})$ is the number of assemblies consisting of i subunits at time t for parameter vector $\vec{x}$, and k is a scaling factor. The number of subunits in a full viral capsid, n, is specified by virus species: 72 for HPV, 120 for HBV, and 90 for CCMV. To simplify later formulas, the notation $S(t,\vec{x})$ is introduced for the average assembly size, or equvalently, pre-scaled simulated curve. The parameter vector $\vec{x}$ consists of a set of on- and off-rates for all possible binding interactions described by the local rule set. $R(t,\vec{x})$ is estimated for each parameter vector $\vec{x}$ by averaging over a set of trajectories to minimize stochastic noise, as described in the following section. In contrast to the prior single-curve fitting [32], a generalized multi-curve fitting scheme is designed to fit $\vec{x}$ and possibly k to a set of true light-scattering curves depicted as $E_1, \cdots, E_m$ representing measurements of assembly at m distinct concentrations (see Appendix B.3 for details about fitting multiple curves simultaneously). The objective function is defined as the root-mean-square deviation (RMSD) between true curves and the corresponding simulated curves $S_1(t,\vec{x}), \cdots, S_m(t,\vec{x})$ for a given parameter vector $\vec{x}$:

$$f(\vec{x}) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\Big(\frac{1}{T_j}\sum_{t=1}^{T_j}\big(E_j(t) - k \times c_j \times S_j(t,\vec{x})\big)^2\Big)} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\frac{(\vec{E}_j - kc_j\vec{S}_j)^T(\vec{E}_j - kc_j\vec{S}_j)}{T_j}} \tag{4.2}$$

The objective function is computed over a series of discrete time points $0, \cdots, T_j$, where T_j is the total number of time points measured in curve j. $\vec{E}$ and $\vec{S}$ are $E(t)$ and $S(t)$ in vector form, respectively, and the superscript T stands for vector transpose.

The fitting also requires the scaling factor k, which is handled differently for the three viruses due to the differences in how the datasets are reported in their studies. For HPV, the scaling factor is reported as $k = 7.04 \times 10^{-8}$ [83] and that value is used throughout the parameter inference.

No scaling factor is reported for the HBV and CCMV studies and they are therefore treated as additional unknowns to be learned. To infer k, it is necessary to assume that no assembly event has occurred at the initial time point for each true light-scattering curve, allowing the use of the approximation $E(0) = R(0, \vec{x}) = k \times c$. The scaling factor is then estimated by minimizing the objective function with given parameter vector $\vec{x}$:

$$\hat{k} = \arg\min_{k} f(k|\vec{x}) = \arg\min_{k} \sqrt{\frac{1}{m}\sum_{j=1}^{m}\Big(\frac{1}{T_j}\sum_{t=1}^{T_j}\big(E_j^*(t) + k \times c_j - k \times c_j \times S_j(t|\vec{x})\big)^2\Big)} \tag{4.3}$$

Here E^* is the modified experimental curve that is shifted along the Y axis with the property that $E^*(0) = 0$, so $E^* + k \times c$ starts at $k \times c$, which meets the above assumption. By finding the zero of the derivative with respect to k, the maximum likelihood estimate of k is obtained by:

$$\hat{k} = \frac{\sum\limits_{j=1}^{m}\left(\frac{c_j}{T_j}\sum\limits_{t=1}^{T_j}\Big(E_j^*(t) \times \big(S_j(t|\vec{x}) - 1\big)\Big)\right)}{\sum\limits_{j=1}^{m}\left(\frac{c_j^2}{T_j}\sum\limits_{t=1}^{T_j}\big(S_j(t|\vec{x}) - 1\big)^2\right)} = \frac{\sum\limits_{j=1}^{m}\frac{c_j \vec{E}_j^{*T}(\vec{S}_j - 1)}{T_j}}{\sum\limits_{j=1}^{m}\frac{c_j^2(\vec{S}_j - 1)^T(\vec{S}_j - 1)}{T_j}} \tag{4.4}$$

The fitting strategy varies for the three viruses based on differences in how the data is reported. For HPV, a single known k is reported and therefore the learning of k is omitted. For HBV, a single unknown k is assumed to be shared across all curves, and therefore it is learned before multi-curve fitting, and the E_j in Equation 4.2 should be replaced by E_j^* accordingly. For CCMV, the strategy differs in such a way that each curve must be fit with a separate k, and this is achieved by applying Equations 4.3 and 4.4 to each curve at a time. The strategy of using separate values of k for each curve calls for an alternative objective function that averages the RMSD across curves:

$$f(\vec{x}) = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{1}{T_j}\sum_{t=1}^{T_j}\Big(E_j^*(t) - \hat{k}_j \times c_j \times \big(S_j(t,\vec{x}) - 1\big)\Big)^2} = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{\|\vec{E}_j^* - \hat{k}_j c_j (S_j - 1)\|_2^2}{T_j}} \tag{4.5}$$

Here k_j is the scaling factor for curve j. The reason for using the new objective function is that the k_j's and $\vec{x}$ that minimize the RMSD of each individual curve will also minimize Equation 4.5, but not necessarily Equation 4.2.

4.3 Optimization scheme

The optimization proceeds by an initial scan in a reduced 2-D parameter space derived by assuming all on-rates in the system are equal and all off-rates are equal. The scan allows the methods to identify a good starting location and proper grid size. Due to the limits of computational time, a search in the full-parameter space that requires many more samples to fit would be too expensive. Instead, the procedure performs a series of staged searches with non-decreasing degrees of freedom. The first stage starts in the same 2-D space of the scan, and each following stage uses the optimum of the previous search stage as an initial guess for a search in an expanded parameter space produced by allowing two previously equal off-rates to vary independently. This process repeats until each off-rate is independently fit. A single on-rate is assumed for all binding sites but off-rates are fit independently so as to balance the need for a model with enough degrees of freedom to fit the data against the need for a model simple enough to be computationally tractable. In the prior study of fitting to the HPV model [32], the on-rates are considered independent but the alternative is chosen here to implicitly model binding rate as diffusion-limited and thus essentially equal between binding sites.

The order in which independent rates are introduced is determined manually for each virus according to geometric similarities among binding sites. For HPV, after learning a single on- and off-rate for all binding sites, the off-rates of the four sites are then broken into two groups

(A |BCD), then three groups (A |BC |D), followed by four (A |B |C |D), and finaly the faster simulation of HPV model allows for a five parameter search with the four off-rates plus a unified on-rate. For HBV, the steps by which groups are subdivided are AD |BC, A |D |BC, and A |D |B |C. The scheme for CCMV is AB |C, and A |B |C. No additional on-rate search is added for HBV and CCMV. The labels of the binding sites refer to Figure 2.2.

Both the initial scan and following search are conducted in $\log_{10}$-based parameter space.

To evaluate the objective function with given parameter, simulations are conducted with 720, 600 and 450 subunits for HPV, HBV and CCMV, respectively. The number of replica runs on each sample varies for the three optimization methods: for Kumar method, 40 trajectories are simulated for each concentration on grid samples, and 250 on predictions; for MCS and SNOBFIT, the number is 50 for all samples, as they do not differentiate samples in such way. The Kumar method has been applied on all three virus models, while MCS and SNOBFIT are only applied on HPV and HBV models due to the high time cost of simulating the CCMV model.

In addition to the parameters and function values, SNOBFIT also requires information about the uncertainty of the function values. For simplicity, that is not estimated for every sample, but rather estimated at the starting point by bootstrapping from 1000 trajectories. Each time, 50 trajectories are randomly picked out of the 1000 and the corresponding RMSD is computed. This repeats for 100 iterations with replacements and the standard deviation of the 100 RMSDs is treated as an estimate of stochastic noise for every point sampled.

4.4 Results

The search begins for each virus model with an initial scan of a broad range within a simplified 2-D parameter space, in which all on-rates are assumed to be equal as are all off-rates. These initial scans are used to initialize a broader search, but are also useful for visualizing the complexity of the parameter space. Figure 4.1 shows the results of these two-dimensional scans for the three viruses. Figure 4.1A reveals a seemingly simple objective function for HPV, consistent with

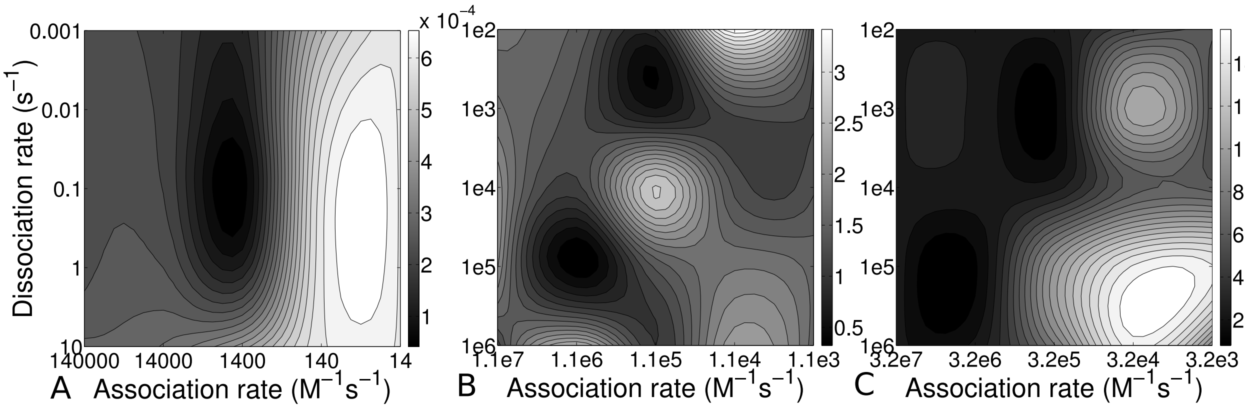

Figure 4.1: Contour plots demonstrating quality of fit in a reduced two-dimensional parameter space. Each plot shows quality of fit as a function of one association rate along the x axis and one dissociation rate along the y axis, with all bonds assumed to exhibit the same pair of rates. Axes are shown on a logarithmic scale. Quality of fit is denoted by shading of the curves, with lighter shades representing high RMSD, i.e., poor fit, and darker shades representing low RMSD, i.e., good fit. (A) Quality of fit of HPV for unified dissociation rates from 0.001 to 10 s^{-1} and unified association rates from 14 to 1.4×10^5 M^{-1} s^{-1}. (B) Quality of fit of HBV for unified dissociation rates from 10^2 to 10^6 s^{-1} and unified association rates from 1.1×10^3 to 1.1×10^7 M^{-1} s^{-1}. (C) Quality of fit of CCMV for unified dissociation rates from 10^2 to 10^6 s^{-1} and unified association rates from 3.2×10^3 to 3.2×10^7 M^{-1} s^{-1}

the results of the prior work [32], with a single strong local minimum and a relatively smooth approach to that minimum across the space examined. Note that this convex appearance does not guarantee that the space is truly convex, as there may be roughness at a finer scale than what is examined in 2-D, or that the full parameter space is similarly simple. Figure 4.1B shows a more complex objective function for HBV, with two comparably deep local minima and at least one other less-pronounced minimum. Figure 4.1C shows the comparable plot for CCMV, revealing a qualitatively similar multi-valley profile to HBV, although with broader and more pronounced maxima and minima. Collectively, then, the three viruses show different portraits consistent with significant variability from system-to-system in the nature of the objective surface and the apparent difficulty of the associated global optimization problem.

The search next proceeds with the application of all three optimization methods. Table 4.1 shows the final fit parameters, number of samples, and lowest RMSDs achieved by all applications. One additional step to make the results more comparable is the normalization of RMSD.

Table 4.1: Parameters, number of samples, and lowest RMSDs from best-fitting experimental SLS data. The unit of on-rate is M^{-1} s^{-1}, and the unit of off-rate is s^{-1}.

Virus	Method	Min RMSD	# samples	On-rate	A-off	B-off	C-off	D-off
HPV	Kumar	0.0754	1524	1.4e3	0.12	0.11	0.12	0.13
HPV	MCS	0.0667	2959	1.4e3	0.21	0.11	0.11	0.11
HPV	SNOBFIT	0.0679	1574	1.5e3	0.044	4.8	4.8	4.8
HBV	Kumar	0.0592	886	1.4e6	1.2e5	1.4e5	1.4e5	1.2e5
HBV	MCS	0.0487	1615	1.24e6	9.86e4	1.15e5	1.15e5	9.91e4
HBV	SNOBFIT	0.0476	1229	8.9e5	5.2e4	1.9e5	2.5e4	9.3e4
CCMV	Kumar	0.0460	515	1.2e6	3.0e4	3.0e4	3.9e4	

The same quantity in raw RMSD may reflect worse quality of fit if the SLS curves have a smaller span of intensity range without normalization, and vice versa. Each RMSD in Table 4.1 is normalized by root mean square height (RMSH) of the curves in its respective dataset. For HPV with known k, the normalizer is:

$$F = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\frac{1}{T_j}\sum_{t=1}^{T_j}E_j^2)} \tag{4.6}$$

For HBV with single inferred k:

$$F(k) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\big(\frac{1}{T_j}\sum_{t=1}^{T_j}(E_j^* + k \times c_j)^2\big)} \tag{4.7}$$

For CCMV with inferred vector $\vec{k}$:

$$F(\vec{k}) = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{1}{T_j}\sum_{t=1}^{T_j}(E_j^* + k_j \times c_j)^2} \tag{4.8}$$

The RMSH is equivalent to the RMSD between the curves and zero lines.

For HPV, the Kumar and MCS methods have a tight agreement on the inferred rate, while SNOBFIT has a 44-fold increase of B, C, D off-rates in comparison to the rates inferred by Kumar method. The normalized RMSDs suggest that the quality of fits are similar, with MCS performing best, followed by SNOBFIT, then Kumar. The methods also show variations in numbers of samples they required to make a fit, with Kumar most efficient by this measure, followed by SNOBFIT, then MCS.

For HBV, the table shows similar parameter inferences as well for the methods, with the most extreme outlier being a variation of 5.7-fold between inferred C off-rates between the Kumar and SNOBFIT methods. The RMSDs suggest that the quality of fits are also similar, with SNOBFIT performing best, followed by MCS, then Kumar. In terms of numbers of samples, Kumar is again the most efficient by this measure, followed by SNOBFIT, then MCS.

Comparing across the virus models, a greater number of degrees of freedom for a model (5 for HPV, 4 for HBV, 3 for CCMV) will lead to a greater the number of sampled points.

The best fit curves show the quality of fits for HPV, HBV and CCMV (Figure 4.2). The three methods gives quite consistent and generally good fits to HPV and HBV for both short- and long-timescale features of the curves, while the two DFO methods gives closer fits for the highest concentration. For HPV, all methods have a slight tendency to over-estimate the middle concentration, and Kumar method to under-estimate the highest concentration. The true HPV data tends to show a sharper initial slope at the end of the lag phase than do the simulated curves, although that may in part reflect deviation between the idealized light-scattering model we assume and the true sensitivity of the instrument for smaller oligomers. For HBV, all methods somewhat underestimated the lowest concentration and overestimated the middle concentration. MCS gives a closer fit to the highest concentration, while SNOBFIT gives a closer fit to the middle concentration.

4.5 Discussion

In general, all of the methods found plausible fits to the experimental curves. Compared to the shapes of best-fit curves and the values of inferred parameters, the underlying assembly pathways might be of much greater interest for revealing the *in vitro* assembly process. In Chapter 6, some computational methods will be presented for analyzing the assembly pathways and theoretical

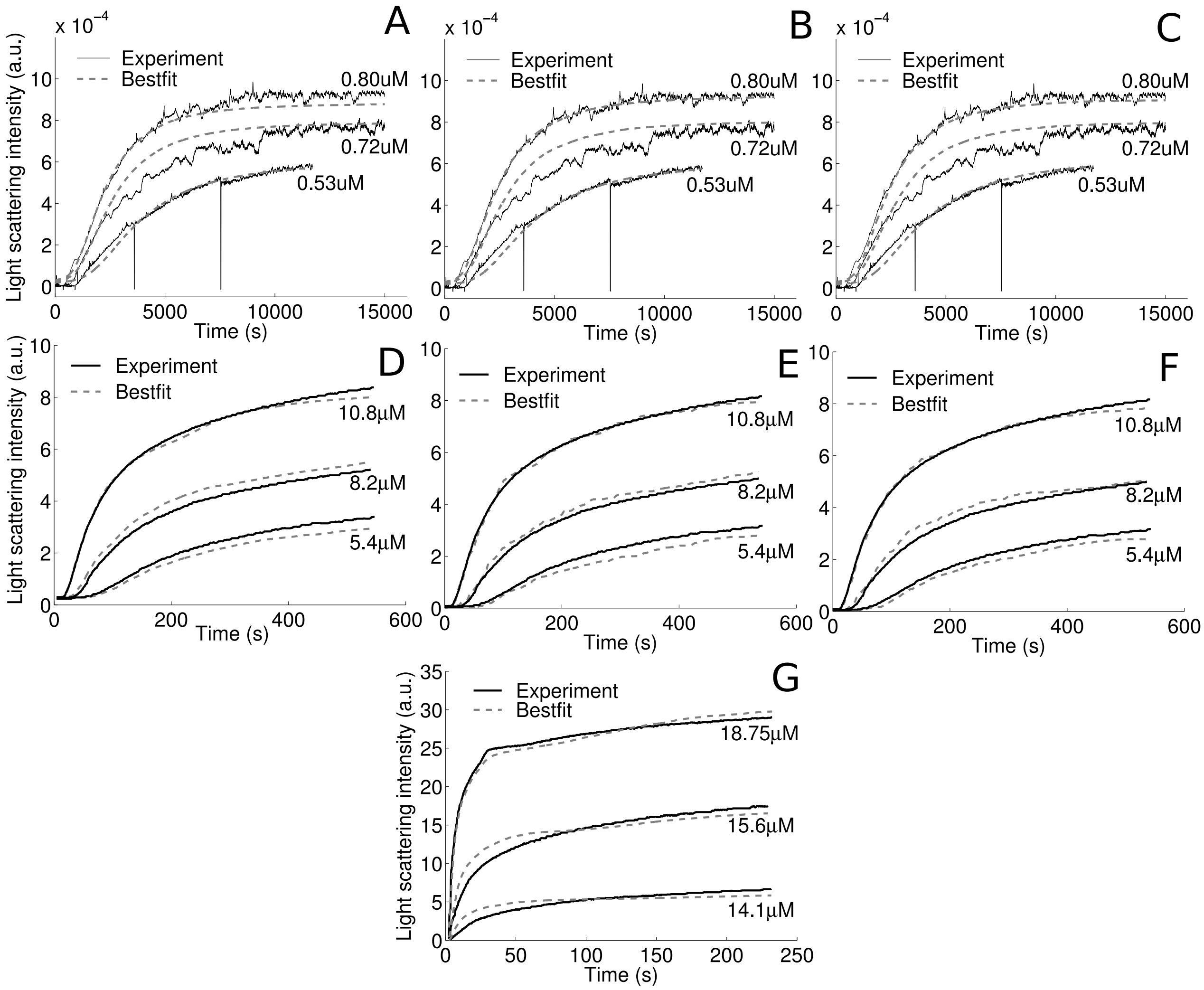

Figure 4.2: Best fit curves for each method to the real SLS data. Each subpart shows three concentrations used in fitting, with true data in solid black lines and the best model fits in dashed grey lines. A: Kumar on HPV; B: MCS on HPV; C: SNOBFIT on HPV; D: Kumar on HBV; E: MCS on HBV; F: SNOBFIT on HBV; G: Kumar on CCMV.

models described for the prediction of *in vivo* assembly.

Given the nature of the SLS measure, one may raise the question that whether a single SLS curve corresponds to a unique assembly pathway, or SLS itself is an under-determined measurement for revealing the complexity of assembly? One possible improvement might be using alternative data sources that have more channels of observations, such as non-covalent mass spectrometry (NCMS), dynamic light scattering (DLS), small angle X-ray scattering (SAXS), etc. In Chapter 5, synthetic SLS and NCMS datasets are used to investigate the uncertainty of inferred pathways from SLS data and the advantages of increased accuracy from fitting NCMS data.

Chapter 5

Application on Synthetic Datasets[*]

5.1 Sources of datasets

To evaluate the qualities of fit resulting from different possible algorithms and data sources, it is necessary to have multiple data types on a common system with a known ground-truth parameter set and simulation model. Since there is, to the author's knowledge, no alternative method for learning these properties of a complex molecular assembly, a synthetic variant of the HBV model was created with the HBV structure (Figure 2.2) but a set of artificially chosen rate constants selected to maintain a realistic nucleation-limited growth mechanism while producing rapid assembly. Such synthetic datasets were produced at four concentrations corresponding to c = 5.3, 6.4, 8.0 and 10.6 µM, if we assume our simulations each represent a cubic volume of dimensions $0.5 \times 0.5 \times 0.5$ m^3 (0.125 fL). For more details about generating data under multiple concentrations please read Appendix B.3. Table 5.1 provides the corresponding on-rates in M^{-1}s^{-1} and off-rates in s^{-1}.

Two types of synthetic bulk-measure data are generated: static light scattering (SLS) and non-covalent mass spectrometry (NCMS). The synthetic SLS data are produced by feeding DESSA 1.5.8 with the parameters in Table 5.1, and then converting its output with Equation B.12. For

[*] The content of this chapter is mainly based on the submitted work of Xie et al. [71].

Table 5.1: On and off-rates at each binding site used in our synthetic data generation and fitting. The binding site labels refer to Figure 2.2.

Binding site	On-rate (M^{-1} s^{-1})	Off-rate (s^{-1})
A	9.48e5	7.94e3
B	5.98e5	1.26e4
C	3.78e5	2.00e4
D	2.38e5	3.16e4

the convenience in fitting synthetic data, we set the values of k such that kc = 1, 1, 1.5, and 2 for concentrations c = 5.3, 6.4, 8.0 and 10.6 µM, respectively.

The synthetic NCMS data are produced by a highly idealized model representative of the ideal theoretically possible from NCMS, assuming the ability of exact peak assignments, deconvolution of contributions of distinct charge states, and precise quantification of mass fractions at each peak. Although real data would be far noisier and more ambiguous, an idealized model would better serve the goal of providing a comparative model of a maximally data-rich system. Under these assumptions, NCMS is simulated by averaging the mass fraction of each intermediate assembly (including full capsid) at every second:

$$M_i(t, \vec{x}_0) = \frac{N_i(t, \vec{x}_0) \times i}{\sum_{j=1}^{n} \left(N_j(t, \vec{x}_0) \times j\right)} \tag{5.1}$$

Here n is the number of subunits in a full capsid (n = 120 for HBV), $M_i(t, \vec{x}_0)$ is the mass fraction of assemblies that have i subunits at time point t with parameter $\vec{x}_0$ from Table 5.1, $N_i(t, \vec{x}_0)$ is the number of assemblies that has i subunits at time point t with the same parameter.

For each concentration, the datasets of both SLS and NCMS are averaged from 10,000 trajectories of 250 seconds assembly time with 600 assembly subunits per simulation. The synthetic datasets are divided into three variants of parameter inference:

1. 1-SLS, which only contains the SLS curve with c = 8.0 µM;

2. 3-SLS, which contains the SLS curves with c = 5.3, 8.0 and 10.6 µM;

3. MS, which contains the NCMS data with c = 8.0 µM.

The synthetic SLS and NCMS datasets under c = 6.4 µM are used to verify how the model will behave at a new concentration using parameters inferred from a different concentration.

5.2 Objective function

The objective function for fitting 1-SLS and 3-SLS datasets is identical to that for fitting real HPV SLS curves with a single known k (Equation 4.2). The objective function for fitting MS dataset is the RMSD between all pairs of mass fraction curves:

$$f(\vec{x}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{1}{T}\sum_{t=1}^{T}\big(M_i(t,\vec{x}_0) - M_i^*(t,\vec{x})\big)^2\Big)} \tag{5.2}$$

As the mass fraction curve for every assembly has the same number of time points, the above equation can be rewritten in the form of Frobenius norm:

$$f(\vec{x}) = \frac{\|\mathbf{M}(\vec{x}_0) - \mathbf{M}^*(\vec{x})\|_F}{\sqrt{nT}} \tag{5.3}$$

Here the mass fractions are represented in matrix form with $M_i(t)$ being the element at index (i, t) of M.

5.3 Optimization scheme

Unlike the real SLS datasets, the fast simulation of synthetic datasets make it feasible to search directly in the 8-parameter space (on- and off-rates for each of four binding sites) with more points evaluated every iteration, rather than using the parameter subdivision scheme we applied to the real data. For the Kumar method, on each search iteration, the fit objective function is evaluated at 128 grid points in the log parameter space and a new minimum candidate is predicted based on the objective values. The grid points are picked in the same fashion as in section 3.2. The function value is evaluated by averaging 40 replica trajectories at each grid point and 1280 replicas at the minimum candidate to minimize stochastic noise. On average, each point is evaluated by $(128 \times 40 + 1280)/(128 + 1) \approx 49.6$ trajectories, which is close in number to the 50 trajectories used for evaluating each point in MCS and SNOBFIT. MCS is again used as a caller for our simulator, with the MCS solver determining which parameter points to evaluate and when to terminate the search. SNOBFIT is fed with values at 128 points and asked for 128 new points to evaluate each iteration, with the search terminating after 10 non-improving iterations.

The uncertainty required by SNOBFIT is estimated by bootstrapping from 10,000 trajectories with the true parameter set. Each time 50 trajectories were randomly drawn to compute the RMSD, and this repeated for 100 times with replacements. The mean of the 100 RMSDs are treated as the estimate of stochastic noise for all sample points. Comparing to the noise estimated from initial point of fitting real HBV SLS data, the two estimates differ by about 20%, suggesting that it is reasonable to apply a common noise value to all points evaluated along the search invoked by SNOBFIT.

The parameters inferred by fitting synthetic datasets were then used to predict the assembly behavior under concentration c = 6.4 μM, where the deviation in SLS curves and mass fractions are investigated.

5.4 Results

The quantitative results of fitting all datasets with all methods and their prediction deviations are shown in Table 5.2. Comparing the deviation of inferred parameters across different datasets and methods, we can conclude that SNOBFIT gives the most consistent estimates across data types, while the Kumar method is most dependent on the quality of the data source. SNOBFIT consistently gives the lowest best-fit RMSD, while the Kumar method gives the highest RMSD. The difference in best-fit RMSD across the three datasets is small, however. All three methods make much better predictions, in terms of RMSD, from fitting to the richer datasets. The RMSD

Table 5.2: Quantitative assessment of synthetic data fitting for the Kumar, MCS, and SNOBFIT methods.
The columns, in order identify the data source, number of functional evaluations required, mean error in parameter fits, RMSD of the best-fit parameters to the training data, RMSD of the best-fit parameters to SLS data at a concentration not used in training, and RMSD of the best-fit parameters to NCMS data at a concentration not used in training.

Method	Dataset	# samples	Mean parameter deviation	Best-fit RMSD	Prediction SLS RMSD	Prediction MS RMSD
Kumar	1-SLS	2710	2.01	0.0667	0.981	1.01
MCS	1-SLS	2021	1.48	0.0325	0.174	0.716
SNOBFIT	1-SLS	2322	0.85	0.0161	0.0545	0.113
Kumar	3-SLS	4645	1.99	0.0787	0.132	0.766
MCS	3-SLS	2361	1.62	0.0572	0.123	0.613
SNOBFIT	3-SLS	3225	0.46	0.0188	0.0166	0.0180
Kumar	MS	3097	0.34	0.0457	0.0558	0.0502
MCS	MS	1657	1.11	0.0296	0.0929	0.0533
SNOBFIT	MS	3741	0.88	0.0198	0.0133	0.0138

of SLS is, again, normalized by RMSH of Equation 4.6 and the RMSD of MS is normalized by:

$$F = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(\frac{1}{T}\sum_{t=1}^{T}M_i(t,\vec{x}_0)^2\big)} \tag{5.4}$$

Given that small changes in RMSD can correspond to significant changes in inferred parameters, the consideration of fit quality is extended to consider not just the best-fit but also the range of fits within the margin of noise of the best-fit for each method and data source. Figure 5.1 shows boxplots for each optimization based on the set of quasi-optimal parameter sets that give RMSD scores within 2 standard deviations of the best-fit. The standard deviation here is the same quantity as what is used to estimate noise levels for SNOBFIT by bootstrapping. This provides a rough assessment of the degree of uncertainty in each parameter to some extent. The Kumar method gives the tightest distribution of quasi-optimum parameter sets, which results from the nature of its local optimization strategy. SNOBFIT gives the widest distribution of quasi-optimum parameter sets. Fitting richer datasets tightens the distribution of quasi-optimum parameter sets for each method, showing that the greater complexity of data is helpful in more

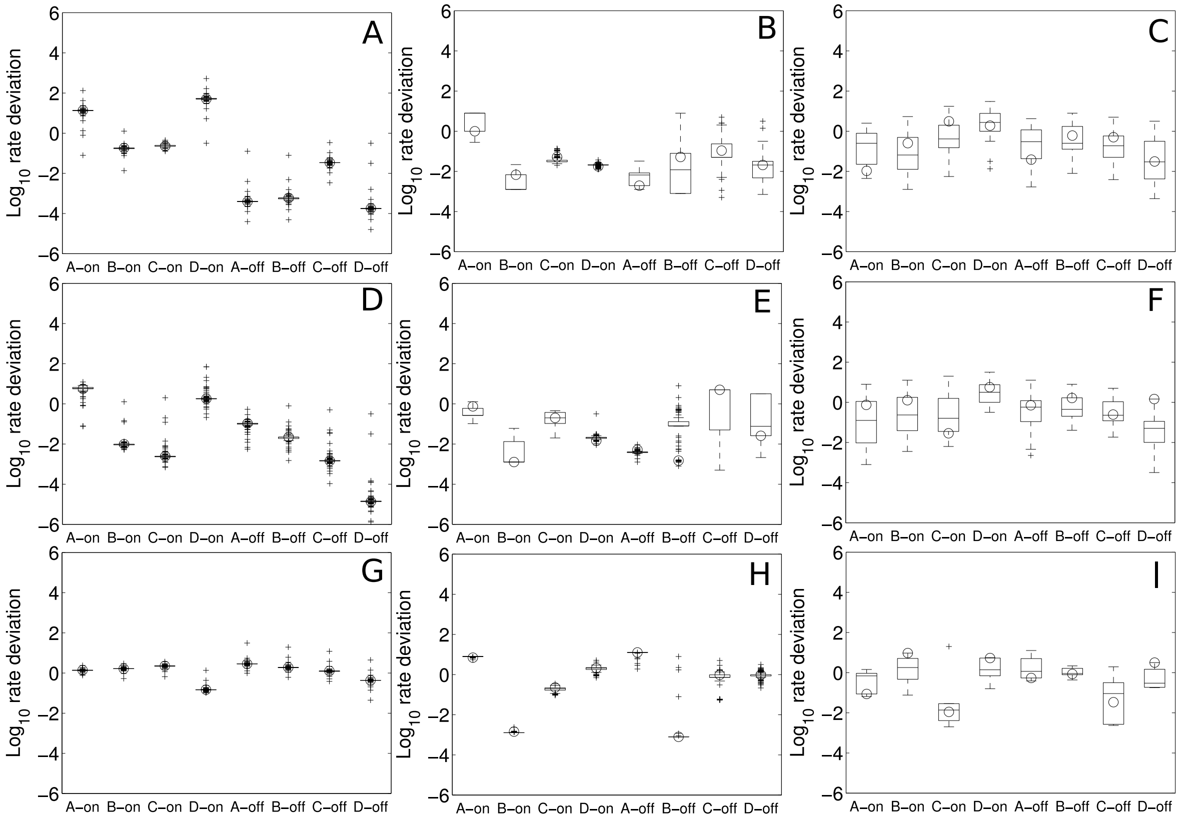

Figure 5.1: Deviation of inferred log parameters from true values on the synthetic data sets for each algorithm and data type. In each plot, circles mark the deviation with minimum RMSD, and boxes mark mean and range of variation for high-quality fits, defined to be those whose RMSD is within two standard deviations of the estimated noise level. The eight parameters in log space on each plot correspond to A on-rate, B on-rate, C on-rate, and D on-rate in M^{-1} s^{-1} followed by A off-rate, B off-rate, C off-rate, and D off-rate in s^{-1}. The subfigures show the three algorithms separated by column (A, D, G) Kumar; (B, E, H) MCS; (C, F, I) SNOBFIT and the three data sources separated by row (A, B, C) 1-SLS; (D, E, F) 3-SLS; (G, H, I) NCMS.

precisely pinning down the true optimal fits.

The quality of fits is next examined in terms of true and inferred profiles of assembly progress versus time, which provide a more direct view of fit quality. Figure 5.2 compares fits to simulated SLS profiles for each data set used in fitting. For the 1-SLS data (Figure 5.2 A), only SNOBFIT gives a close fit to the experimental curve, with best-fits from Kumar and MCS showing substantially shorter lag phases and less pronounced sigmoidal behavior than the true curve. 3-SLS (Figure 5.2 B) leads to an improvement for all three methods, although SNOBFIT still yields noticeably better fits than the others. All three methods give good fits to the NCMS data (Figure 5.2

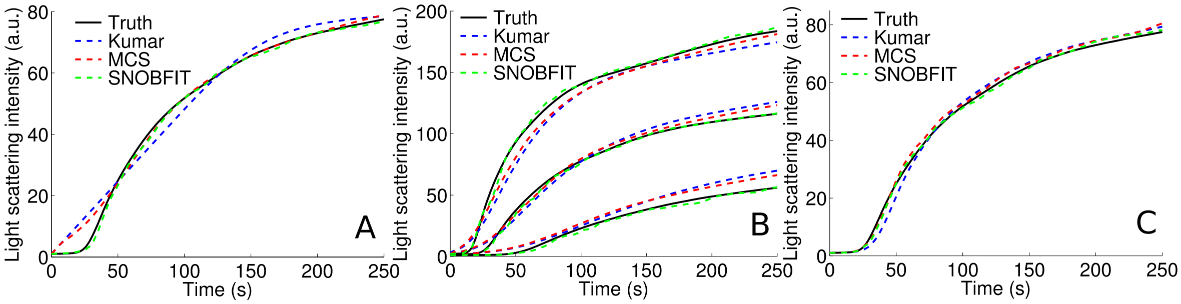

Figure 5.2: Best fit curves for each method to the synthetic data as assessed by light scattering intensity for (A) 1-SLS; (B) 3-SLS; and (C) NCMS. Each subpart shows the curve simulated from the true parameters in solid black lines and the best fits for the Kumar method in dashed blue, MCS in dashed red, and SNOBFIT in dashed green.

C), with apparently similar fit quality for the three.

A more stringent test is given by examining fits to mass fractions of specific intermediate species (Figure 5.3). For this purpose, three sizes of assemblies are selected as representatives of three ranges of abundance, as assessed by mass fraction: full capsid (high abundance), trimer of subunits (medium abundance), and decamer of subunits (low abundance). As expected, the algorithms all do a better job fitting more abundant species, which is unsurprising since those species would have greater weight in computing the objective function. The high abundance capsomer species is fit well by SNOBFIT for all data sources but by Kumar and MCS only when fit to NCMS data. The medium abundance trimer intermediate (Figure 5.3, 2nd column) is fit poorly by all three methods on 1-SLS data, very well only by SNOBFIT for 3-SLS data, and fit well by all three methods for NCMS data. The low-abundance decamer species (Figure 5.3, 3rd column) is poorly fit by all three methods, with peak accuracy of roughly a factor of two for all three methods on NCMS data.

An even more stringent test, to help control for the possibility of overfitting, is to evaluate fit quality at an additional concentration not used in parameter inference. In relative concentrations, this experiment involves learning fits from c = 8.0 μM for 1-SLS and NCMS data or from c = 5.3, 8.0 and 10.6 μM for 3-SLS data, then evaluating the quality of the fit in each case at c = 6.4 μM. Figure 5.4 shows qualities of fit for simulated SLS curves. For 1-SLS (Figure 5.4 A), only

SNOBFIT yields a high quality prediction, with moderate but noticeably worse quality for MCS and very poor fitting for the Kumar method. Prediction from 3-SLS data (Figure 5.4 B) similarly shows a high-quality fit only for SNOBIT. Kumar and MCS show nearly identical fits to one another, with the Kumar fit substantially better than it was for 1-SLS but the MCS fit very similar to that found by MCS on 1-SLS data. All methods give plausible fits with parameters inferred from fitting NCMS data (Figure 5.4 C), although the SNOBFIT curve is still slightly better than those from Kumar and MCS.

The pattern seen in predicted mass fractions (Figure 5.5) is similar to that found in Figure 5.3 testing fit to the training data. The high-abundance capsomer species is fit well only by SNOBFIT for 1-SLS and 3-SLS data, but by all three methods for NCMS data. Fitting is poorer for the medium-abundance trimer species for all methods, but still reasonable for all three with NCMS data, good only for SNOBFIT with 3-SLS data, and poor for all three methods with 1-SLS data. None of the methods achieves a close fit to the low-abundance decamer species from any data set, although all come within approximately a factor of two for NCMS data.

5.5 Discussion

The use of synthetic datasets provided us with a relatively fair testbed for assessing different optimization algorithms and data sources. While a variety of parameters and pathways may result in very similar SLS curves, fitting more than SLS curve at once or using more sophisticated solver such as SNOBFIT can help approximate the true parameters and mass fractions. The advantage of using a rich data source such as NCMS can somehow eclipse the differences in solver performance.

One limitation of this study is in the estimation of parameter uncertainty. In Figure 5.1, the uncertainty in each parameter is estimated individually by considering sub-optimal objective values. These estimates provide only a limited view of the true range of parameter values consistent witht the data. Further studies will be needed to reveal correlations among parameters and/or build a more sophisticated probabilistic model of objective values across the full space of parameters.

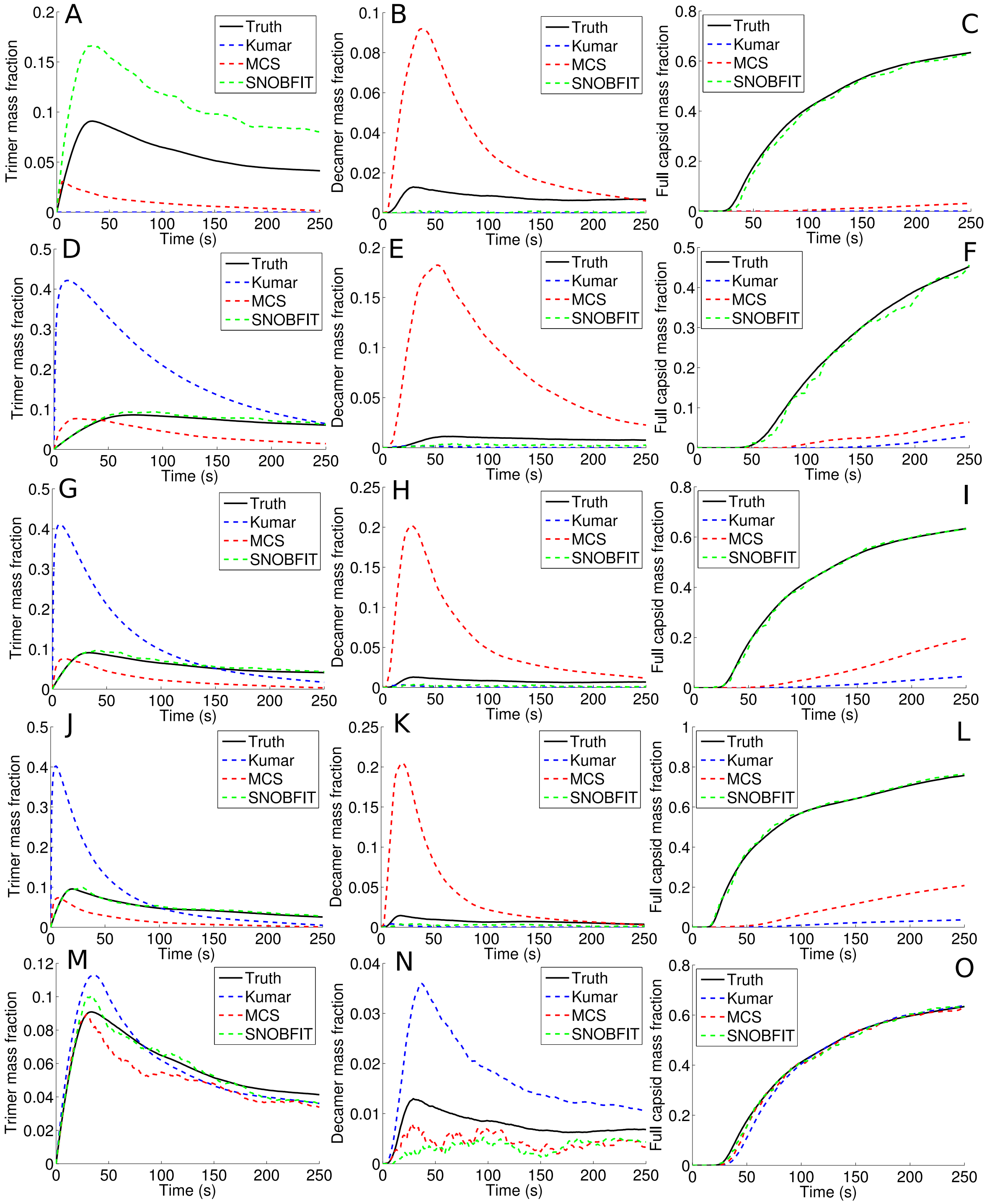

Figure 5.3: Best fit curves from synthetic data for a representative sample of mass fractions. Each subfigure compares true mass fraction in solid black lines versus the best fits for the Kumar method in dashed blue, MCS in dashed red, and SNOBFIT in dashed green. Columns correspond to (A, D, G, J, M) mass fractions of a high-abundance trimer species, (B, E, H, K, N) a low-abundance decamer species, and (C, F, I, L, O) complete capsid. Rows correspond to data sources: (A, B, C) 1-SLS; (D, E, F) 5.3 μM, (G, H, I) 8.0 μM, and (J, K, L) 10.6 μM concentration curves from 3-SLS; and (M, N, O) NCMS.

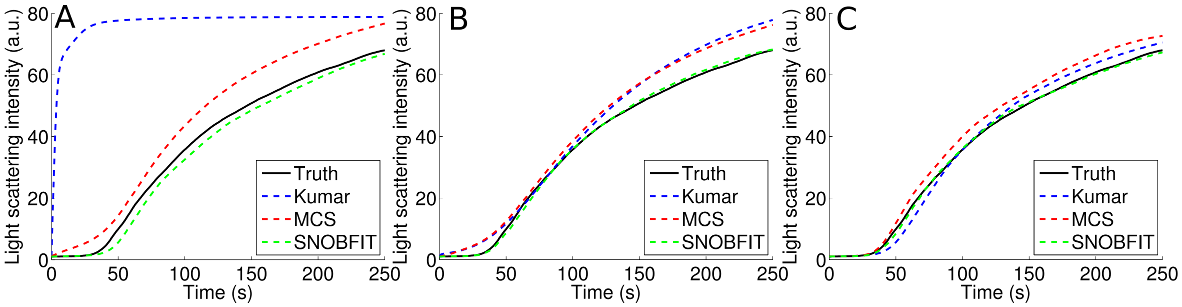

Figure 5.4: Predicted versus true light scattering curves for synthetic data by each method and data source for a concentration not used in data-fitting. Each subfigure compares true SLS in solid black lines versus the predictions of the Kumar method in dashed blue, MCS in dashed red, and SNOBFIT in dashed green for one data source at a concentration omitted from data-fitting. (A) 1-SLS; (B) 3-SLS; (C) NCMS.

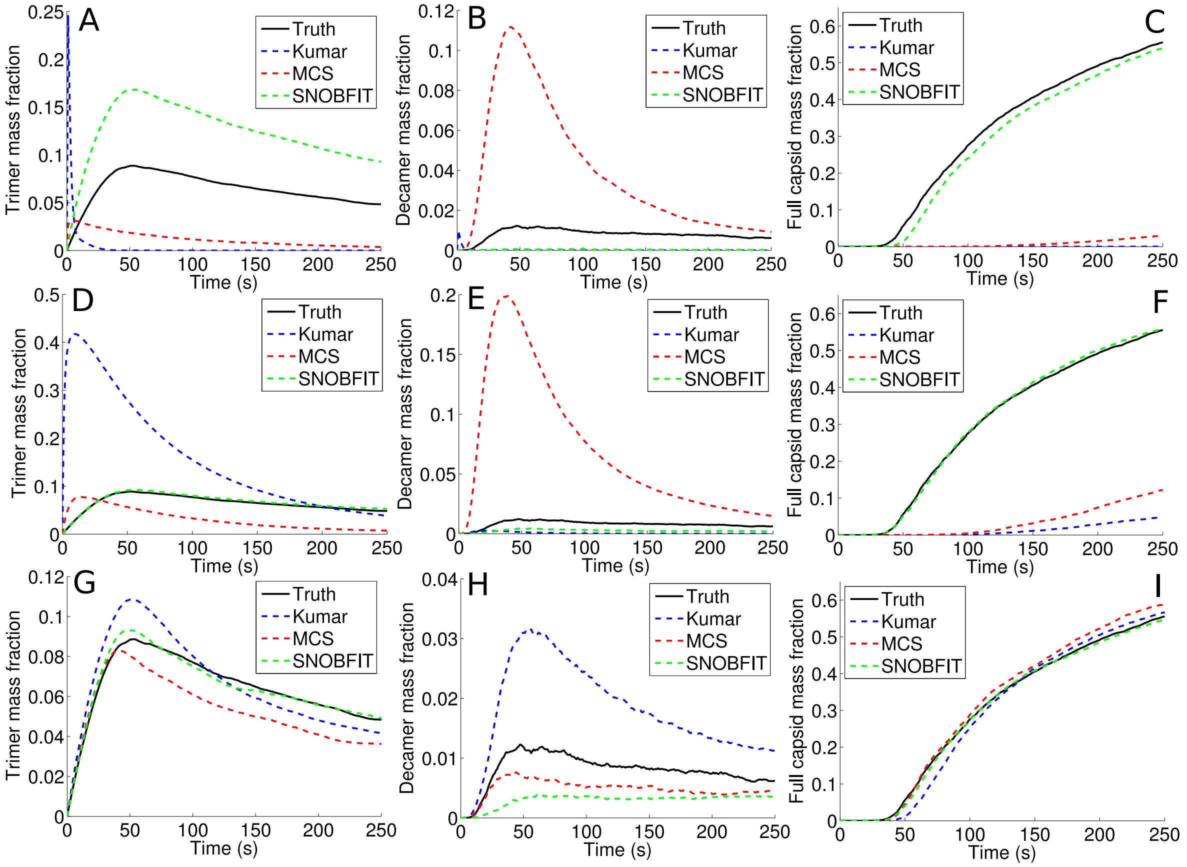

Figure 5.5: Predicted versus true mass fractions for synthetic data by each method and data source for a concentration not used in data-fitting for a representative selection of intermediate species. Each subfigure compares true SLS in solid black lines versus the predictions of the Kumar method in dashed blue, MCS in dashed red, and SNOBFIT in dashed green for one data source at a concentration omitted from data-fitting. Rows correspond to data source: (A, B, C) 1-SLS; (D, E, F) 3-SLS; (G, H, I) NCMS. Columns to intermediate species profiled: (A, D, G) medium-abundance trimer intermediate; (B, E, H) low-abundance decamer intermediate; (C, F, I) high-abundance complete capsomer.

Chapter 6

Self-Assembly Pathway Analysis[*]

A key point of the studies in the thesis is to infer features of capsid assembly pathways and provide a basis for predicting assembly behavior under more realistic representation of the *in vivo* environment. This chapter will describe the methods that help analyze the *in vitro* assembly pathways of virus capsids, and the prediction of *in vivo* assembly made by kinetic approximation of *in vivo* conditions. All the work presented in this chapter is based on the parameters inferred from *in vitro* SLS data.

6.1 Analysis by mass fraction

A mass fraction plot (Figure 6.1) shows the distribution of intermediates as a function of time for a single simulation trajectory. They are generated by re-running the simulator with the best-fitting parameters for experimental SLS data, and collect output every 10 steps (see Appendix A for information about selecting the number of steps).

The mass fraction plot of HPV (Figure 6.1 A) is generated with 720 subunits at 0.80 µM. The curves show no preference for building up any specific pools of small oligomers. Neither do they

[*] The content of section 6.1and 6.2 is based on the published work of Xie et al. [70]; the content of section 6.3 is derived from the Masters thesis work of Feng [87] and submitted work of Smith et al. [88] with partial contribution from the author; the content of section 6 is derived from the published and submitted work of Smith et al. [88, 89], with partial contribution from the author.

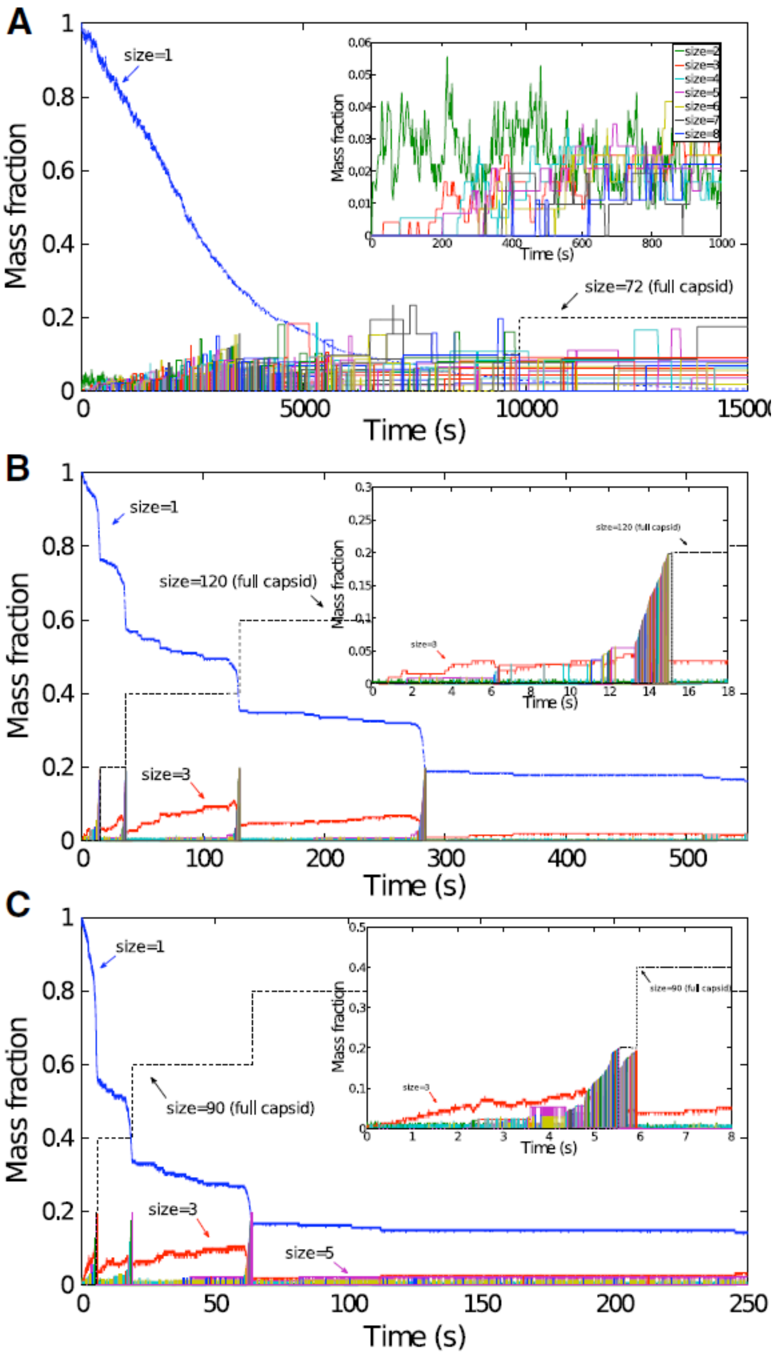

Figure 6.1: Mass fractions of intermediates versus time for sample trajectories of the three capsid systems. Each curve corresponds to a single size of intermediate species. Insets in each plot show magnification of early stages of the reaction. While all possible intermediate sizes are plotted, for simplicity a key is provided only for sizes 28. (A) Mass fractions versus time for 720 HPV capsomer subunits at 0.80 µM. (B) Mass fractions versus time for 600 HPV dimer subunits at 10.8 µM. (C) Mass fractions versus time for 450 CCMV dimer subunits at 14.1 µM.

show a pronounced depletion for larger oligomers, as one would expect for a nucleation-limited assembly. Rather, the plots show the mass fraction of each intermediate coming up slightly later and slightly lower than the next smaller intermediate. This pattern is consistent with a model of assembly by successive accretion of individual capsomers without a defined nucleation step.

Profiles of intermediates present a very different picture for HBV than was seen for HPV. Figure 6.1 B shows a sample simulation trajectory for HBV with 600 subunits at 10.8 mM. The most prominent feature of the plot is a set of intermittent spikes in which the system rapidly cycles through a series of successively larger intermediates, culminating in production of a new capsid. These spikes are the signature of nucleation-limited growth, with each spike touched off by production of some small nucleus followed by rapid completion of the capsid through a series of elongation reactions. Another prominent feature of the plot is the existence of a pool of free trimers (hexamers of coat protein) forming quickly and persisting during the assembly reaction. Monomers and trimers of subunits (dimers and hexamers of coat proteins) apparently reach an equilibrium with one another early in the reaction that readjusts with the production of each capsid.

The overall assembly process of the CCMV model also appears qualitatively more similar to that of HBV than HPV. Mass fractions of CCMV intermediates versus time, shown in Figure 6.1 C for 450 subunits at 14.1 μM, again show the spikes characteristic of a nucleation-limited growth mechanism. Like with HBV, there are also persistent pools of oligomers throughout the assembly, with trimers appearing to dominate as with HBV. Unlike with HBV, a small standing population of pentamers becomes apparent late in the reaction. Both viruses show a qualitatively similar process of gradual accumulation of trimers in the lag between nucleations, which are then rapidly depleted during elongation of a new capsid.

6.2 Analysis by reactant usage

The reactant usage plot (Figure 6.2) visualizes the pathway space by plotting how frequently any given reactant oligomer sizes are used to assemble any given product oligomer size. Each panel in Figure 6.2 is an average from 50 trajectories. The horizontal index is the size of a reactant, and the vertical index is the size of a product. The brightness of each grid box indicates the fraction of times a given size of product is built with a given size of reactant.

For HPV, Figure 6.2 A-C confirms that each oligomer size is normally produced by addition of a single capsomer to the next smaller oligomer size, with the exception of some rare reactions between pairs of oligomers during the earlier steps of assembly. This conclusion is consistent with that found by the prior work of fitting to a single light-scattering curve [32]. Comparison between concentrations shows that the same non-nucleation-limited capsomer addition pathway is used across the concentration range with only variation in the frequency of relatively rare oligomer-oligomer binding steps early in assembly. At higher concentrations, such oligomer-oligomer binding is more often observed, although still rare.

For HBV, the reactant usage profile in Figure 6.2 D-F reveals that monomers and trimers are used at random for individual steps of the assembly, with monomers usually favored but trimers used roughly 10-20% of the time at most elongation steps. The system thus appears to be described not by a single pathway but by an ensemble of many distinct assembly pathways. The production of small oligomers revealed in the inset shows a more complex profile, with more frequent use of trimers and occasionally pentamers, and with production of octamers in particular occurring primarily by binding of trimers to pentamers. While it is difficult to define a precise set of steps as the nucleation from this profile, the results do suggest that a low-frequency pentameric form plays an important role in initiating assembly. The profiles are nearly identical across the twofold concentration change shown in Figure 6.2, D-F.

For CCMV, reaction frequencies visualized in Figure 6.2 G-I shows that assembly most often proceeds by monomer additions, but that trimer additions occur with lower frequencies, pro-

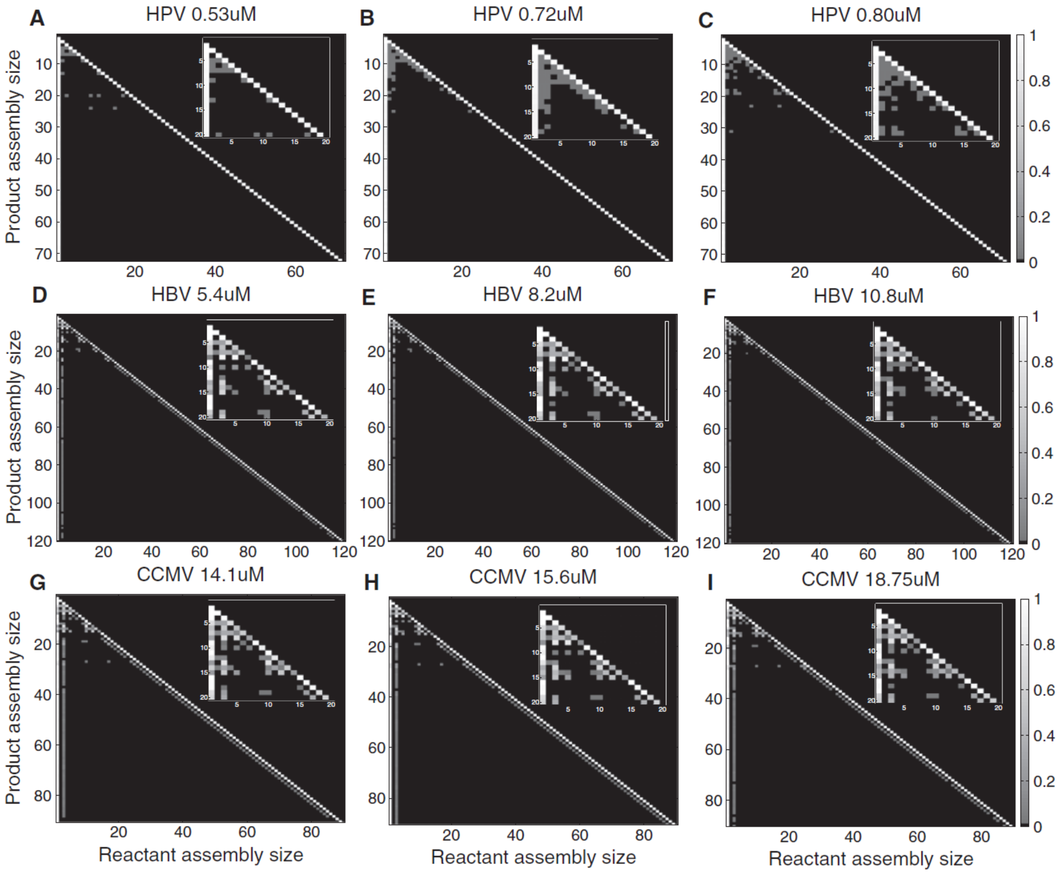

Figure 6.2: Visualization of reaction usage for viral assembly reactions. Each plot is organized into a set of rows and columns such that the shading of the box in row i and column j denotes the fraction of oligomers of size i produced by a binding reaction involving an oligomer of size j. Lighter colors indicate higher frequencies, and darker colors indicate lower frequencies, with black representing a reaction unobserved in the course of the simulations. Each subfigure shows the full plot of all possible assembly sizes. Inset in the upper right show magnification of a portion of the upper-left region of the full plot corresponding to reactions producing smaller oligomers. (A) HPVat 0.53 µM. (B) HPVat 0.72 µM. (C) HPVat 0.80 µM. (D) HBVat 5.4 µM. (E) HBVat 8.2 µM. (F) HBVat 10.8 µM. (G) CCMV at 14.1 µM. (H) CCMV at 15.6 µM. (I) CCMV at 18.75 µM.

viding an ensemble of secondary assembly pathways. Early steps in assembly show a similar frequency profile to that observed for HBV, although there also appears to be a single frequently used step involving large (10-mer and 17-mer) oligomers that is not seen with HBV. Like HBV, CCMV is inferred to assemble through an ensemble of distinct pathways rather than a single defined pathway. Pathway usage appears insensitive to at least the range of concentration changes examined in Figure 6.2 G-I.

6.3 Analysis by particle visualization

The simulation inside DESSA carries much more information than the numerial output of intermediate counts use for fitting SLS and NCMS data. The DESSA simulator can be customized so that it will produce detailed information, such as reactants and products of binding/breaking events, subunit positions of all intermediates, as functions of time.

By back-tracing the formation path from a complete capsid to the originating subunit, Smith et al. extracted the entire pathway and made the *in vitro* assembly progress of a CCMV capsid into a movie by combining continuous MATLAB plots. Figure 6.3 A shows an important snapshot that captures the most plausible structure of a nucleus. The timeline of the movie is proportional to the appearance time of the assembly normalized by the formation time of the capsid. By carefully checking the pathway, Smith et al. found that no significant structure was formed by the presence of the suspected nucleus, and the completion of the capsid rapidly follows this snapshot.

With the help of GPU acceleration, Feng was able to recreate a much more vivid movie of HPV assembly progress (Figure 6.3 B). In his effort, the real all-atom structure of an HPV capsomer is mapped to the simulation-predicted position by spatial translation and rotation. The position of each assembly is randomly picked because of DESSA's disregard of particle diffusion. The frames are drawn from the simulation trajectory at an interval of approximately 40 simulation steps. Each frame contains 170,784 atoms whose positions are determined by GPU implementation of his algorithm and then rendered by PyMol. The movie is composed by 50 such frames with each lasting for 1 second.

6.4 Predictions of *in vivo* assembly

To approximate the transition from *in vitro* to *in vivo* assembly, the model must be improved to account for two major factors: molecular crowding and the effects introduced by nucleic acid. Given the best-fitting parameters inferred from SLS data, necessary modifications are made to these parameters to reflect the two effects on the new model.

The effect of molecular crowding is learned through off-lattice particle simulations implemented with Greens function reaction dynamics [88, 90, 91]. In these simulations, different percentages of the reaction volume are occupied by inert particles and the influence on kinetic rates is monitored. Parameters are modified for CCMV at 15.6 µM, HBV at 8.2 µM and HPV at 0.72 µM under 0% to 45% crowding conditions, and then fed to DESSA for simulated results (Figure 6.4). For CCMV at low crowding levels, Figure 6.4 A shows a pattern of decreasing speed and yield of assembly, with the assembly rate reaching a minimum at 25% crowding. The effect reverses at higher crowding levels, with the assembly rate at 35% crowding approaching that of the uncrowded system, and with 40% and 45% crowding yielding faster assembly at intermediate time points of the simulation. All trajectories go to equivalent levels of completion eventually, although with varying kinetics. Figure 6.4 B shows curves for HBV, which show qualitatively similar behavior to those for CCMV. HBV also shows a pattern of decreasing speed and quantity of assembly at low crowding levels, again reaching a minimum at 25% crowding, but increased assembly with respect to both speed and yield as crowding levels continue to increase. Crowding levels above 30% begin to approach the assembly rate of the 0% crowding state. 45% crowding yields higher assembly rates than 0% crowding levels in the later stages of assembly. HBV yields a higher apparent variance in the final yield of completed capsids than does CCMV. With HBV, assembly yield initially drops along with assembly rate as crowding is introduced, with yields at 1035% crowding well below those of the uncrowded case. Yield approaches that of the uncrowded system by 40% crowding and surpasses it at 45% crowding. Figure 6.4 C shows curves for HPV, which show strikingly different behavior than the CCMV or HBV simulations. HPV shows a monotonic decrease in both rate and yield of assembly with increasing crowding rates. The curves also show a much lower variance than did the HBV or CCMV curves, with the effects of increasing crowding clearly distinguishable from the noise in the individual averaged simulated light scattering curves.

To simulate RNA effects within the stochastic simulator, it was necessary to develop fast approximations to the full effects of RNA on the on- and off-rates of subunit-subunit interactions. For this purpose, we used a set of simple analytical approximations of specific effects RNA would be expected to have on capsid assembly. The modification needed to introduce a model of RNA to CCMV assembly were subdivided into four factors [89]:

- The entropy of RNA chain compression.
- Energetic and entropic contributions to the free energy of RNA-RNA interactions.
- Free energy of RNA-protein interactions.
- Local coat protein concentration by RNA.

The modifications are made to the best-fitting parameters of CCMV 15.6 μM SLS data. Figure 6.5 shows simulated light scattering curves for CCMV under conditions of no RNA factor, each individual factor, and the combination of all four factors. Because of the large difference in time scales between reactions, Figure 6.5 is shown in two versions showing a slow timescale (part A for 100 seconds) and a fast timescale (part B for the first second). The effect of RNA compression moderately reduces the speed of capsid assembly, although assembly is still achieved. The RNA-RNA interaction effect alone prevents any large intermediates from being formed, with nothing above an 8-mer assembled in any simulation run. On the other hand, RNA-protein interaction and increased protein concentration both dramatically increase the rate of capsid assembly. Both of these simulated curves show similar kinetics to the combined RNA effects curve, which also shows a far faster assembly rate than the hollow capsid curve.

6.5 Discussion

Simulation-based approaches have the advantage of providing data in both high volume and fine detail that are well suited to the task of pathway analysis. Three computational methods - mass fraction, reactant usage and visualization - have been explored in this chapter and they serve as analytic tools for understanding the detailed information on capsid assembly contained within the raw simulated trajectories. While each of them reveals different features of interest regarding the pathways, a combined approach may further assist the analysis. For example, the movies made from the assembly process have great visual impact, but it is hard to pinpoint critical structures from fast moving objects; this problem might be better addressed by finding “urn-points” from the associated numerical output where significant changes of rate of assembly progress are observed.

The prediction of *in vivo* capsid assembly depends on three bases: the inferred parameters from fitting experimental data; theoretical and mathematical derivations of free energies of protein-protein, protein-RNA and RNA-RNA interactions; and simulations based on adjusted kinetic rates. The studies regarding this subject are still in preliminary state. Furthermore, other factors missing from the current study, such as chaperone proteins, may play important roles in assisting *in vivo* assembly. Building a more general and realistic model better accounting for likely cellular influences on assembly pathways remains a promising future direction.

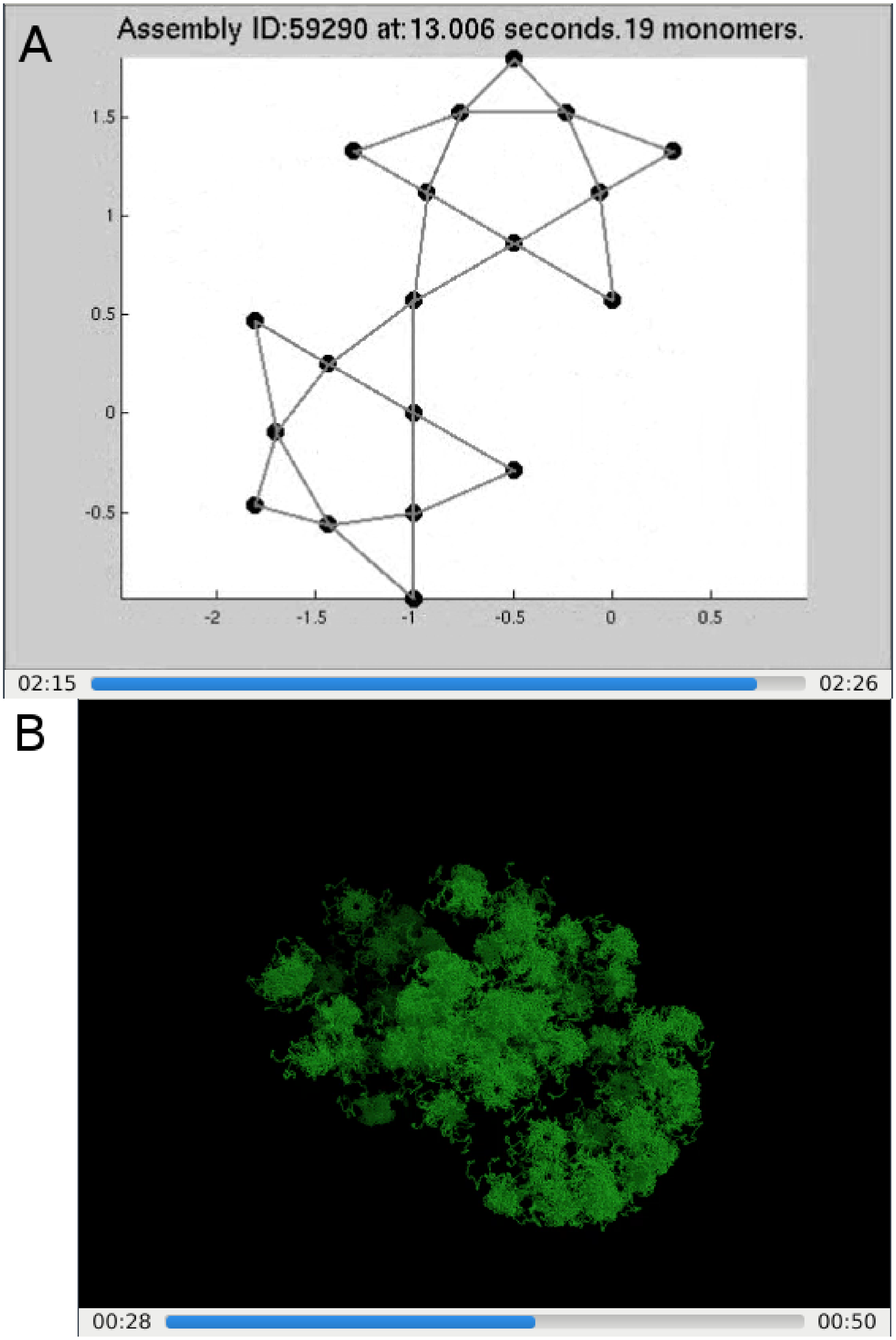

Figure 6.3: Visualization of CCMV and HPV *in vitro* assembly. (A) A snapshot of simulated CCMV *in vitro* assembly at 14.1 µM. Courtesy of G. Smith. [89] (B) A snapshot of simulated HPV *in vitro* assembly at 0.53 µM. Courtesy of X. Feng. [87]

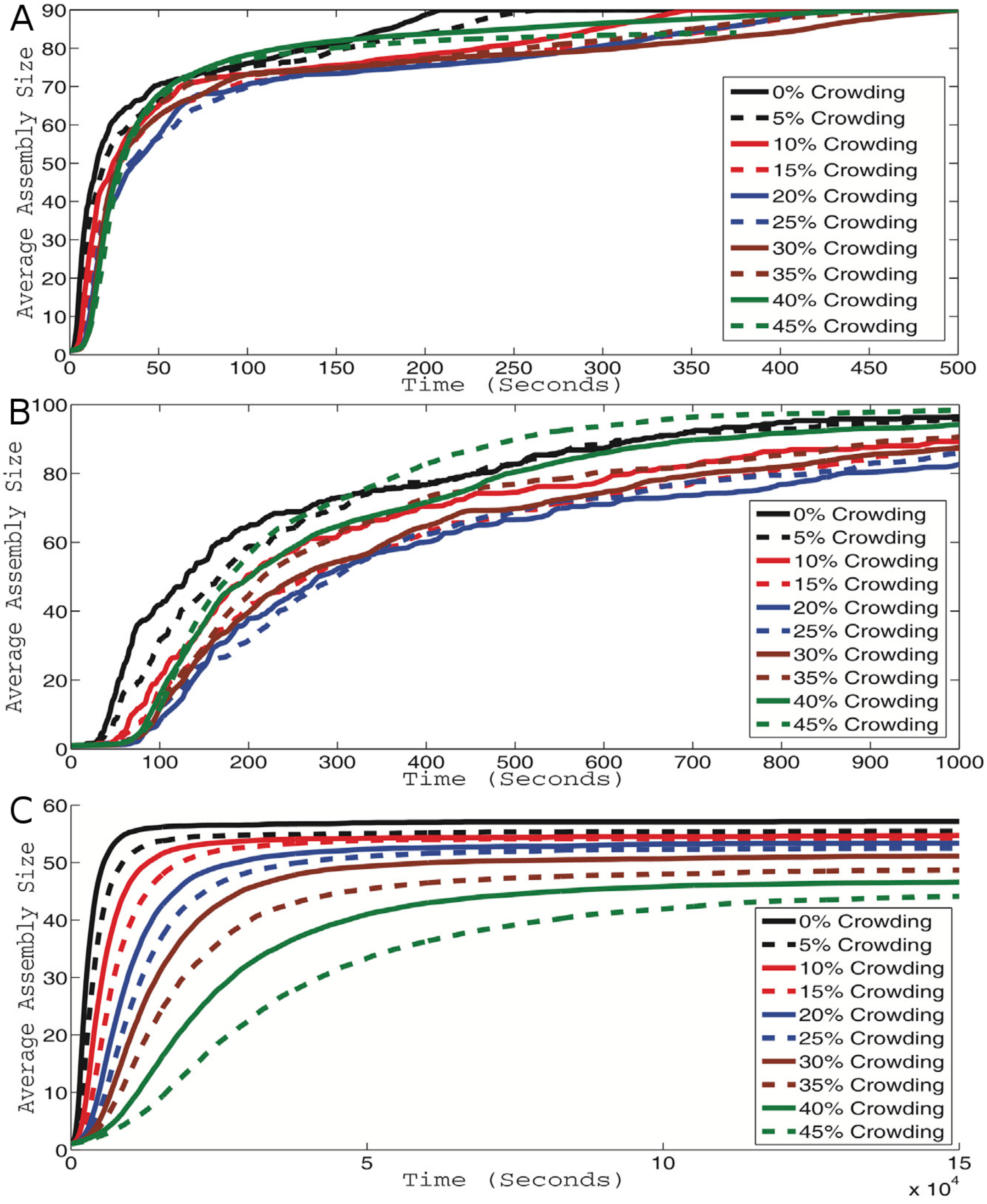

Figure 6.4: Simulated light scattering curves for CCMV at 15.6 μM (A), HBV at 8.2 μM (B), and HPV at 0.72 μM (C). Each curve represents an average simulated light scattering over 100 simulation trajectories. Curves are shown for levels of nonspecific crowding agents from 0% to 45% of simulation solution volume in increments of 5%. (Figure 1 in [88])

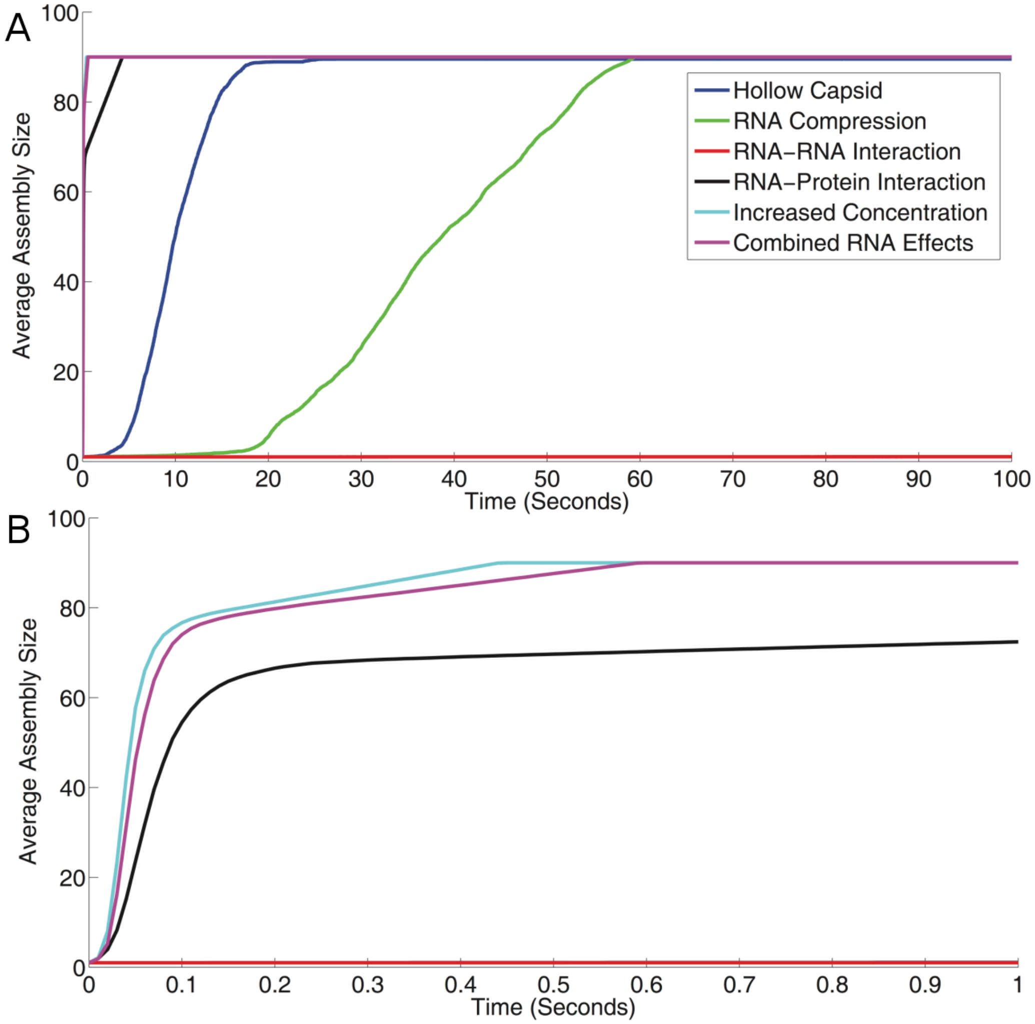

Figure 6.5: Simulated light scattering curves for CCMV capsid assembly under all individual RNA effects as well as the hollow capsid and combined RNA effects case. Each curve represents an average simulated light scattering over 100 simulation trajectories. (A) shows the entire simulation time course while (B) shows the first second. (Figure S1 in [89])

Chapter 7

Conclusion[*]

In this theis, the author has integrated existing methods for simulation-based data fitting for learning physical parameters of self-assembly systems into a broader framework and applied these methods to a study of kinetic rate parameters of three icosahedral virus capsid assembly systems. The results of this work lead to two major conclusions about the methodology and the capsid systems themselves:

- The results demonstrate that the algorithms are, at least in principle, able to learn different kinds of assembly pathways and rate parameters for a small collection of icosahedral viruses with qualitatively similar structures and bulk assembly kinetics. Our parameter estimation technique generates good fits between experimental and simulated light-scattering curves, even when fitting a common parameter set to multiple curves at different concentrations to reduce redundancy of solutions.
- The results are suggestive of some of the variability in assembly mechanisms that may exist between structurally similar viruses. The inferred models show a variety of behaviors, including either nucleation-limited or non-nucleation-limited assembly, monomer-based or hierarchical oligomer-based assembly, and assembly consistent with a single well-defined

[*] A large portion of this chapter is derived from the published work of Xie et al. ([70]) and submitted work of Xie et al. ([71])

pathway or assembly that can only be described as an ensemble of a very large collection of distinct pathways.

7.1 Variety in assembly pathways

Two out of the three viruses, HBV and CCMV, fit pathways with many similar features despite different shell geometries and despite sizeable differences in inferred rate parameters. The third virus, HPV, shows very different behavior in both rates and pathways. These three examples, then, collectively demonstrate that there may not be any consistent paradigm of virus assembly but rather a diversity of strategies for assembly comparable to the great diversity seen in viral structures and other features of their life cycles.

The inferred models raise several questions about assembly of the specific viruses examined. One surprising result is the apparent lack of nucleation-limited assembly in the HPV model, a feature noted in previous work on that system [32]. Numerous theoretical models have suggested nucleation-limited growth as a key feature for preventing kinetic traps in assembly [10, 13, 14, 15, 30] and there are indeed large amounts of incomplete species in the HPV simulations. This lack of nucleation-limited growth may reflect some inability of the model to learn the correct assembly pattern, perhaps because it lacks some important feature essential to true HPV assembly. The prior work left open the possibility that the model-fitting technique might have some inherent inability to discover nucleation-limited parameter sets, although the currently available results from HBV and CCMV now refute that hypothesis.

Another intriguing observation is the apparent lack of a single defined pathway for HBV and CCMV in favor of what is, effectively, a random sampling from a large ensemble of possible pathways involving either incorporation of monomers or oligomers at each step of assembly. While there is no clear reason as to why virus assembly should not proceed by stochastic sampling from a large set of possible pathways, the assumption that a reaction has a defined pathway is nonetheless often implicit in how people reason about biochemical systems. A failure of this assumption may have important implications for the use of simplified theoretical models to describe and reason about such systems, especially on small scales at which the stochastic nature of the pathway space should be particularly pronounced.

Furthermore, an absence of any individual reactions necessary to productive assembly may have important practical implications for efforts to develop capsid assembly targeted antivirals [92, 93, 94]. A final observation is that the similar pathway usage across concentrations suggests that all three viruses sit at points in parameter space that are relatively insensitive to perturbation, in contrast to expectations from theoretical studies [19] that pathway usage is quite sensitive to small changes in model parameters across a large fraction of the parameter space. It is possible that such robustness to perturbation might be a general feature evolutionarily selected in real capsid assembly, although far more examples would be needed to draw any such conclusion with confidence.

7.2 Advantage of using DFO methods

Comparison of the three optimization methods leads to the conclusion that SNOBFIT does better than the other two methods at fitting parameters of the three virus capsid assembly models across data types. MCS is known to be a superior method to SNOBFIT for some other applications [82] but SNOBFIT appears to be particularly well suited to dealing with the high stochastic noise typical of this data fitting problem. Note that this stochastic noise is inherent to the fact of using an SSA model to sample trajectories, a decision that itself has proven necessary for sampling large enough numbers of trajectories in reasonable amounts of time. This same issue would be expected to confront any simulation-based optimization of a system faced with similar combinatorial blowup of intermediate species, a general issue of self-assembly models but also one confronting other systems for which similar rule-based modeling have been applied [39, 40, 41, 42, 43, 44, 45]. It is therefore arguable that the observations in this thesis are likely to be far more broadly applicable than just fitting capsid assembly models. On the other hand, SNOBFIT tends to identify a larger uncertainty in fits than do the other methods. In these cases, the other methods are probably underestimating their true uncertainty since they do not survey the parameter space as thoroughly. The Kumar method largely relies on local optimization and thus might be expected to miss near-optima that also yield plausible fits to the true data. MCS is intended to be a global optimizer like SNOBFIT but might be expected to do a less complete survey of near optima in the presence of noise, explaining why it yields intermediate estimates of variance between the Kumar method and SNOBFIT.

7.3 Potential benefit from richer data

When comparing results across data sources, one must be cautious in considering conclusions given by this thesis as definitive because of the dependence of the results largely on synthetic data. The need for synthetic data, rather than solely real experimental measurements, is largely due to 3 concerns:

- It is impossible to rigorously evaluate accuracy of fitting without a known ground truth, which is unavailable for any real system.
- The interest in understanding the limits theoretically possible for these approaches calls for exploring an idealized model rather than any true data as the representation of maximally data-rich experimental data.
- With synthetic data, it is feasible to work in a parameter domain in which trajectories are qualitatively similar to those of the real system but much faster to simulate, allowing the test of full potential of the algorithms rather than relying on heuristic compromises needed with the real HBV data.

Point (1) is the most difficult issue to sidestep but is also the major reason work in this direction is important: at present, there is, to the author's knowledge, no alternative approach to our data-fitting methods for learning detailed kinetic models of a non-trivial self-assembly system. Point

(2) merits further study using a variety of real data sources, including NCMS [95], DLS [96], and SAXS [97], which should provide information somewhere between the 3-SLS used in the current practice and the idealized model of NCMS in its ability to precisely constrain the feasible parameter set. While all of these methods have been used for capsid assembly studies previously, the author does not know any one system for multiple such data types are available to allow a fair comparison and believe it is wisest to identify likely best practices by purely *in silico* studies like that published here before committing to a major experimental undertaking. Point (3) is an issue of compute power and is in principle solvable by applying more powerful computers for longer times than currently available. To drop the heuristic compromises on the real data for our current cluster hardware (typically consisting of 80 compute nodes in continuous use), however, would require years of continuous compute time and is therefore achievable in principle but not in practice. The kinds of resources that direction would require exist but again it would be wiser to identify best practices with these compromises to either find ways to bring down the cost or develop a clearer justification for a major commitment of compute hardware to this task.

7.4 Future directions

There are many avenues by which this work might be further advanced. The major computational bottleneck in the workflow of parameter inference is the time spent on ruuning simulations. One possible solution is to port Java-based DESSA into a faster computer language such as C++, while another possible solution is to incorporate hybrid SSA-ODE algorithms to reduce the time spent on futile trial-and-error events.

Despite their better ability to identify the global optimum, the DFO methods are, nevertheless, generic optimization methods without any particular optimization for self-assembly problems or reaction networks more broadly. It is possible to derive a more specific optimization method targeted to the stochasticity of such problems. For example, methods with surrogate functions constructed with respect to the likelihood of a self-assembly simulation trajectory given a set of parameters (see Appendix C) might give closer approximate to the objective function than methods with generic surrogate functions.

Although synthetic data filled an important role in evaluating the methods of this thesis, more definitive results will require evaluating parameter fits inferred from real instances of more data-rich measurements. The work of parameter inference from DLS data has already been initiated and led by Thomas as the subject of his Masters thesis in progress. There are, however, substantial technical challenges to using DLS data. For example, the fitting of DLS measurements requires transferring large volumes of detailed geometric information from the Java-based DESSA to the MATLAB-based optimizer, which requires tighter integration between the simulator and optimizer than is available in the current implementation.

Another promising direction is the assessment of parameter sensitivity, i.e., how the objective function would respond to fluctuations of parameters and/or combinations of parameters. The rates in Table 4.1 show that relatively large variations in parameters may still yield similar RMSD values, which raises the question of the uncertainty in the inferred parameters. As a primitive way of assessing uncertainties in the parameters, Figure 5.1 includes parameters that yield RMSDs within two deviations from the optimal value. This single-parameter sensitivity analysis, however, fails to account for the combined influence of groups of two or more parameters. A more sophisticated multiparameter model is needed to more thoroughly account for these cross-parameter effects and to give more accurate estimates of confidence intervals for optimal parameters.

Finally, virus capsids, due to their high symmetry and very large number of possible pathways, would be expected to be a particularly challenging system for such simulation-based data fitting, which has tremendous potential as a way of solving problems in macromolecular assembly that are not amenable to any purely experimental or purely theoretical technologies currently available. While there is much to be done in learning the limits of these methods and establishing best practices for their use, there is strong reason to believe they can be a transformative technology for understanding macromolecular assembly processes and for the much broader project of developing predictive quantitative models of complex systems in biology.

Bibliography

[1] Krogan J, Cagney G, Yu H, Zhong G, Guo X, et al. (2006) Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature* 440: 637-643.

[2] Teschke C, King J, Prevelige P. (1993) Inhibition of viral capsid assembly by 1,1'-bi(4-anilinonaphthalene-5-sulfonic acid). *Biochemistry*, 32: 10658-10665.

[3] Zlotnick A, Stray S. (2003) How does your virus grow? Understanding and interfering with virus assembly. *Trends in Biotechnology*, 21: 546-542.

[4] Sticht J, Humbert M, Findlow S, Bodern J, Muller B, et al. (2005) A peptide inhibitor of HIV-1 assembly in vitro. *Nature Structural and Molecular Biology*, 12: 671-677.

[5] Stray S, Bourne C, Punna S, Lewis W, Finn M, et al. (2005) A heteroaryldihydropyrimidine activates and can misdirect hepatitis B virus capsid assembly. *Proceedings of the National Academy of Sciences USA*, 102: 8138-8143.

[6] Whitesides G, Mathias J, Seto C. (1991) Molecular selfassembly and nanochemistry: a chemical strategy for the synthesis of nanostructures. *Science*, 254:1312-1319.

[7] Whitesides G, Grzybowski B. (2002) Self-assembly at all scales. *Science*, 295:2418-2421.

[8] Zlotnick A. (1994) To build a virus capsid. An equilibrium model of the self assembly of polyhedral protein complexes. *Journal of Molecular Biology*, 241: 59-67.

[9] Berger B, Shor P, Tuck-Kellogg L, King J. (1994) Local rule-based theory of virus shell assembly. *Proceedings of the National Academy of Sciences USA*, 91: 7732-7736.

[10] Schwartz R, Shor P, Prevelige P, Berger B. (1998) Local rules simulation of the kinetics of virus capsid self-assembly. *Biophysical Journal*, 75: 2626-2636.

[11] Schwartz R, Prevelige P, Berger B. (1998) Local rules modeling of nucleation-limited virus capsid assembly (MIT-LCS-TM-584). *MIT Laboratory for Computer Science, Cambridge, MA.*

[12] Rapoport D, Johnson J, Skolnick J. (1999) Supramolecular selfassembly: molecular dynamics modeling of polyhedral shell formation. *Computer Physics Communications*, 122: 231-235.

[13] Hagan M, Chandler D. (2006) Dynamic pathways for viral capsid assembly. *Biophysical Journal*, 91: 42-54.

[14] Zandi R, van der Schoot P, Reguera D, Kegel W, Reiss H. (2006) Classical nucleation theory of virus capsids. *Biophysical Journal*, 90: 1939-1948.

[15] Nguyen H, Reddy V, Brooks C. (2007) 3rd. 2007. Deciphering the kinetic mechanism of spontaneous self-assembly of icosahedral capsids. *Nano Letters*, 7: 338-344.

[16] Zhang T, Schwartz R. (2006) Simulation study of the contribution of oligomer/oligomer binding to capsid assembly kinetics. *Biophysical Journal*, 90: 57-64.

[17] Keef T, Micheletti C, Twarock R. (2006) Master equation approach to the assembly of viral capsids. *Journal of Theoretical Biology*, 242:713-721.

[18] Misra N, Lees D, Zhang T, Schwartz R. (2008) Pathway complexity of model virus capsid assembly systems. *Computational and Mathematical Methods in Medicine*, 9: 277-293.

[19] Sweeney B, Zhang T, Schwartz R. (2008) Exploring the parameter space of complex self-assembly through virus capsid models. *Biophysical Journal*, 94: 772-783.

[20] Pornillos O, Ganser-Pornillos B, Yeager M. (2011) Atomic-level modelling of the HIV capsid. *Nature*, 469: 424-427.

[21] Grime J, Voth G. (2012) Early Stages of the HIV-1 Capsid Protein Lattice Formation. *Bio-*

physical Journal, 103(8): 1774-1783.

[22] Ghosh S, Matsuoka Y, Asai Y, Hsin K, Kitano H. (2012) Software for systems biology: from tools to integrated platforms. *Nature Reviews Genetics*, 12: 821-832.

[23] Karr J, Sanghvi J, Macklin D, Gutschow M, Jacobs J, Bolival B, et al. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, 150: 389-401.

[24] Ceres P, Zlotnick A. (2002) Weak protein-protein interactions are sufficient to drive assembly of hepatitis B virus capsids. *Biochemistry*, 41: 11525-11531.

[25] Parent K, Zlotnick A, Teschke C. (2006) Quantitative analysis of multi-component spherical virus assembly: scaffolding protein contributes to the global stability of phage P22 procapsids. *Journal of Molecular Biology*, 359: 1097-1106.

[26] Zlotnick A, Johnson J, Wingfield P, Stahl S, Endes D. (1999) A theoretical model successfully identifies features of hepatitis B virus capsid assembly. *Biochemistry*, 38: 14644-14652.

[27] Toropova K, Basnak G, Twarock R, Stockley PG, Ranson N. (2008) The three-dimensional structure of genomic RNA in bacteriophage MS2: implications for assembly. *Journal of Molecular Biology*, 375: 824-836.

[28] Reddy V, Giesing H, Morton R, Kumar, A, Post C, et al. (1998) Energetics of quasi-equivalence: Computational analysis of protein-protein interactions in icosahedral viruses. *Biophysical Journal*, 74: 546-558.

[29] Hemberg M, Yaliraki S, Barahona M. (2006) Stochastic kinetics of viral capsid assembly based on detailed protein structures. *Biophysical Journal*, 90: 3029-3042.

[30] Endres D, Zlotnick A. (2002) Model-based analysis of assembly kinetics for virus capsids or other spherical polymers. *Biophysical Journal*, 83: 1217-1230.

[31] Zhang T, Rohlfs R, Schwartz R. (2005) Implementation of a discrete event simulator for biological self-assembly systems. *Proceedings of the 37th Winter Simulation Conference,*

Orlando, FL. 2223-2231.

[32] Kumar M, Schwartz R. (2010) A parameter estimation technique for stochastic self-assembly systems and its application to human papillomavirus self-assembly. *Physical Biology*, 7: 045005.

[33] Schwartz R, Shor P, Berger B. (2005) Local rule simulations of capsid assembly. *Journal of Theoretical Medicine*, 6: 81-86.

[34] Berger B, King J, Schwartz R, Shor P. (2000) Local rule mechanism for selecting icosahedral shell Geometry. *Discrete Applied Mathematics*, 91: 97-111.

[35] Lillacci G, Khammash M. (2010) Parameter estimation and model selection in computational biology. *PLOS Computational Biology*, 6(3): e1000696.

[36] Rudorf S, Thommen M, Rodnina M, Lipowsky R. (2014) Deducing the kinetics of protein aynthesis *in vivo* from the transition rates measured *in vitro*. *PLOS Computational Biology*, 10(10): e1003909.

[37] Elemans M, Florins A, Willems L, Asquith B. (2014) Rates of CTL killing in persistent viral infection *in vivo*. *PLOS Computational Biology*, 10(4): e1003534.

[38] Voit E. (2013) Biochemical systems theory: a review. *ISRN Biomathematics*, 2013: 1-53.

[39] Hlavacek W, Faeder J, Blinov M, Perelson A, Goldstein V. (2003) The complexity of complexes in signal transduction. *Biotechnology and Bioengineering*, 84: 783-794.

[40] Hlavalek W, Faeder J, Blinov M, Posner R, Hucka M, et al. (2006) Rules for modeling signal transduction systems. *Sciences STKE*, 344: re6.

[41] Danos V, Feret J, Fontana W, Harmer R, Krivine J. (2007) Rule-based modelling of cellular signaling. *Lecture Notes in Computer Science*, 4073: 17-41.

[42] Hogg J, Harris L, Stover L, Nair N, Faeder J. (2014) Exact hybrid particle/population simulation of rule-based models of biochemical systems. *PLOS Computational Biology*, 10(4): e1003544.

[43] Mann R, Perna A, Strmbom D, Garnett R, Herbert-Read J, et al. (2013) Multi-scale inference of interaction rules in animal groups using Bayesian model selection. *PLOS Computational Biology*, 9(3): e10002961.

[44] Ollivier J, Shahrezaei V, Swain P. (2010) Scalable rule-based modelling of allosteric proteins and biochemical networks. *PLOS Computational Biology*, 6(11): e1000975.

[45] White D, Kinney M, McDevitt T, Kemp M. (2013) Spatial pattern dynamics of 3D stem cell loss of pluripotency via rules-based computational modeling. *PLOS Computational Biology*, 9(3): e1002952.

[46] Sun J, Garibaldi J, Hodgman C. (2012) Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review. *IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, 9(1): 185-202.

[47] Liepe J, Kirk P, Filippi S, Toni T, Barnes C, et al. (2014) A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. *Nature Protocols*, 9: 439-456.

[48] Komorowski M, Costa M, Rand D, Stumpf M. (2011) Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of National Academy of Science*, 108: 8645-8650.

[49] Srivastava R, Rawlings J. (2014) Parameter estimation in stochastic chemical kinetic models using derivative free optimization and bootstrapping. *Computers and Chemical Engineering*, 63: 152-158.

[50] Jamalyaria F, Rohlfs R, Schwartz R. (2005) Queue-based method for efficient simulation of biological self-assembly systems. *Journal of Computational Physics*, 204: 100-120.

[51] Misra M, Schwartz R. (2008) Efficient stochastic sampling of first-passage times with applications to self-assembly simulations. *Journal of Chemical Physics*, 129: 204109.

[52] Stone J, McGreevy R, Isralewitz B, Schulten K. (2014) GPU-accelerated analysis and vi-

sualization of large structures solved by molecular dynamics flexible fitting. *Royal Society of Chemistry*, 169: 265-283.

[53] Wang X, Xu F, Liu J, Gao B, Liu Y, et al. (2013) Atomic model of rabbit hemorrhagic disease virus by cryo-electron microscopy and crystallography. *PLoS Pathogens*, 9(1): e1003132

[54] Zhao G, Perilla J, Yufenyuy E, Meng X, Chen B, et al. (2013) Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, 497: 643-646

[55] Bernardi R, Melo M, Schulten K. (2014) Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA) - General Subjects*, In Press.

[56] Brown W, Nicolai T. (1993) Dynamic Light Scattering - The Method and Some Appications. *Oxford science publications*, Chapter 6: 272-318.

[57] Tresset G, Le Coeur C, Bryche J, Tatou M, Zeghal M, et al. (2013) Norovirus capsid proteins self-assemble through biphasic kinetics via long-lived stave-like intermediates. *Journal of the American Chemical Society*, 135: 15373-15381.

[58] Zhang T. (2007) Investigation of virus capsid self-assembly kinetics using discrete-event stochastic simulation. Ph.D Theis, Carnegie Mellon University. *ProQuest/UMI Publishing*, 3293505.

[59] Flint S, Enquist L, Racaniello V, Skalka A. (2009) Principles of Virology, 3rd Edition. *ASM Press*, Chapter 4: 82-127.

[60] Bourne C, Lee S, Venkataiah B, Lee A, Korba B, et al. (2008) Small-molecule effectors of hepatitis B virus capsid assembly give insight into virus life cycle. *Journal of Virology*, 82(29): 10262-10270.

[61] Wynne S, Crowther R, Leslie A. (1999) The crystal structure of the human hepatitis B virus capsid. *Molecular Cell*, 3(6): 771-780.

[62] Elrad O, Hagan M. (2010) Encapsulation of a polymer by an icosahedral virus. *Physical Biology*, 7: 045003.

[63] Rapoport D. (2008) Role of reversibility in viral capsid growth: A paradigm for self-assembly. *Physical Review Letters*, 101(18): 186101.

[64] Hagan M, Elrad O. (2010) Understanding the cconcentration dependence of viral capsid assembly kinetics - the origin of the lag time and identifying the critical nucleus size. *Biophysical Journal*, 98: 1065-1074.

[65] Rapoport D. (2004) Self-assembly of polyhedral shells: a molecular dynamics study. *Physical Review E*, 70: 051905.

[66] Johnston I, Louis A, Doye J. (2010) Modelling the self-assembly of virus capsids. *Journal of Physics: Condensed Matter*, 22: 104101.

[67] Gillespie D. (1977) Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81: 2340-2361.

[68] Azadivar F. (1999) Simulation optimization methodologies. *Proceedings of the 31st conference on Winter simulation: Simulation - a bridge to the future*, 1: 93-100.

[69] Schwartz R. (2008) Biological Modeling and Simulation - A survey of practical models, algorithms, and numerical methods. *The MIT Press*, Chapter 5: 75-94.

[70] Xie L, Smith G, Feng X, Schwartz R. (2012) Surveying Capsid Assembly Pathways through Simulation-Based Data Fitting. *Biophysical Journal*, 103: 15451554.

[71] Xie L, Smith G, Schwartz R. (2014) Applying derivative-free optimization to fit kinetic parameters of viral capsid self-assembly models from multi-source bulk in vitro data. *Submitted.*

[72] Huyer W, Neumaier A. (1999) Global optimization by Multilevel Coordinate Search (MCS). *Journal of Global Optimization*, 14: 331-355.

[73] Huyer W, Neumaier A. (2008) SNOBFIT: Stable noisy optimization by branch and fit. *ACM*

Transactions on Mathematical Software, 35: 1-25.

[74] Carson Y, Maria A. (1997) Simulation optimization: methods and applications. *Proceedings of the 29th Winter Simulation Conference*, 1: 118-126.

[75] Fu M. (1994) Optimization via simulation: a review. *Annals of Operations Research*, 53(1): 199-247.

[76] Glynn P. (1986) Optimization of stochastic systems. *Proceedings of the 18th Winter Simulation Conference*, 1: 52-59.

[77] Glynn P. (1989) Optimization of stochastic systems via simulation *Proceedings of the 21st Winter Simulation Conference*, 1: 90-105.

[78] Kiefer J, Wolfowitz J. (1952) Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics* 23 (3): 462-466.

[79] Azadivar F, Talavage J. (1980) Optimization of stochastic simulation models. *Mathematics and Computers in Simulation*, 22(3), 231241.

[80] Levenberg K. (1944) A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2: 164-168.

[81] Marquardt D. (1963) An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11: 431-441.

[82] Rios L, Sahinidis N. (2013) Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3): 1247-1293.

[83] Casini G, Graham D, Heine D, Garcea RWu D. (2004) In vitro papillomavirus capsid assembly analyzed by light scattering. *Virology*, 325: 320-327.

[84] Zlotnick A, Aldrich R, Johnson J, Ceres P, Young M. (2000) Mechanism of capsid assembly for an icosahedral plant virus. *Virology*, 277: 450-456.

[85] Porterfield J, Dhason M, Loeb3 D, Nassal M, Stray S, et al. (2012) Full-Length Hepatitis

B Virus Core Protein Packages Viral and Heterologous RNA with Similarly High Levels of Cooperativity. *Journal of Virology*, 84(14): 7174-7184.

[86] Zhao X, Fox J, Olson N, Baker T, Young M. (1995) *In vitro* assembly of cowpea chlorotic mottle virus from coat protein expressed in Escherichia coli and *in vitro* transcribed viral cDNA. *Virology*, 207: 486-494.

[87] Feng X. (2011) Visualizing virus self-assembly simulation with GPU acceleration. MS Thesis, Computational Biology, Carnegie Mellon University.

[88] Smith G, Xie L, Lee B, Schwartz R. (2014) Applying molecular crowding models to simulations of virus capsid assembly *in vitro*. *Biophysical Journal*, 106(1): 310-320.

[89] Smith G, Xie L, Schwartz R. (2015) Modeling the effect of RNA on capsid assembly pathways via stochastic simulation. *Submitted.*

[90] van Zon J, ten Wolde P. (2005) Greens-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space. *Journal of Chemical Physics*, 123(234910): 1-16.

[91] Lee B, Leduc P, Schwartz R. (2008) Stochastic off-lattice modeling of molecular self-assembly in crowded environments by Greens function reaction dynamics. *Physics Review E* 78(031911): 1-9.

[92] Ternois F, Sticht J, Duquerroy S, Kräusslich HG, Rey F. (2005) The HIV-1 capsid protein C-terminal domain in complex with a virus assembly inhibitor. *Nature Structural & Molecular Biology*, 12: 678-682.

[93] Stray S, Johnson J, Kopek B, Zlotnick A. (2006) An in vitro fluorescence screen to identify antivirals that disrupt hepatitis B virus capsid assembly. *Nature Biotechnology*, 34: 358-362.

[94] Zlotnick A, Lee A, Bourne C, Johnson J, Domanico P, Stray S. (2007) In vitro screening for molecules that affect virus capsid assembly (and other protein association reactions).

Nature Protocols, 2: 490-498.

[95] Knapman T, Morton V, Stonehouse N, Stockley P, Ashcroft A. (2010) Determining the topology of virus assembly intermediates using ion mobility spectrometry - mass spectrometry. *Rapid communications in Mass Spectrometry*, 24: 3033-3042.

[96] Serrière J, Fenel D, Schoehn G, Gouet P, Guillon C. (2013) Biophysical characterization of the feline immunodeficiency virus p24 capsid protein conformation and in vitro capsid assembly. *PLOS ONE*, 8(2): e56424.

[97] Tuma R, Tsuruta H, French K, Prevelige P. (2008) Detection of intermediates and kinetic control during assembly of bacteriophage P22 procapsid. *Journal of Molecular Biology*, 381: 1395-1406.

Appendix A

Simulator usage

The simulations in this thesis are performed by DESSA 1.5.8, which can be obtained from here:

```
http://www.cs.cmu.edu/~russells/projects/dessa/dessa.html
```

The compiliation and execution of DESSA require the presence of *vecmath* package. An example source of *vecmath* would be:

```
http://www.java2s.com/Code/JarDownload/vecmath/vecmath-1.5.2.jar.
zip
```

Assume *vecmath* exists as *vecmath.jar* file in the same folder of DESSA source files. The compilation of DESSA is done by Java 2 SDK in the following way:

```
javac -cp vecmath.jar *.java
```

There are two ways to run DESSA. The first method needs all *.class* files and *vecmath.jar* to be present in the same folder, and call the following command:

```
java -cp .:vecmath.jar Test capsidmodel.xml aaa bbb ccc ddd
```

Here *Test* is the main class, *capsidmodel.xml* contains the capsid model in *.xml* format, *aaa* defines the number of stochastic steps per output, *bbb* defines the length of the simulation in seconds, *ccc* is the random number seed, and *ddd* regulates the width of output table. The DESSA package contains a few samples of capsid models, which describe binding interactions and corresponding times in the *.xml* format.

To create a more portable version of DESSA excutable, you can pack DESSA and *vecmath* into a single *dessa.jar* file as:

```
jar -xf vecmath.jar
echo 'Main-Class: Test' > manifest.txt
echo 'Class-Path: .' >> manifest.txt
jar -cvfm dessa.jar manifest.txt *.class javax
```

and the subsequent excution of DESSA would be simply:

```
java -jar dessa.jar capsidmodel.xml aaa bbb ccc ddd
```

You might need to increase the maximum memory allocated to DESSA if you encounter the out of memory exception. For example,

```
java -Xmx2048m -jar dessa.jar capsidmodel.xml aaa bbb ccc ddd
```

would allocate 2048MB memory to DESSA.

Appendix B

Convertion between time and rate

In the work of this thesis, the parameters of the capsid models are rate constants measured in $M^{-1}s^{-1}$ and s^{-1} for binding and breaking, respectively. In fact, the DESSA simulator can only take inputs in the format of particle numbers and expected reaction time encoded in the *.xml* files. In this section, the author will demonstrate the conversion between time and rate.

For a breaking reaction, the rate is simply the inversion of time. For binding reactions, two types of binding reactions will be discussed separately.

B.1 Heterogeneous binding

For a heterogeneous binding reaction between binding sites A_+ and A_- with rate constant k:

$$A_+ + A_- \xrightarrow{k} AA$$

The molar reaction rate is defined as:

$$\frac{d[AA]}{dt} = k[A_+][A_-] \tag{B.1}$$

Here $[A_+]$, $[A_-]$, $[AA]$ are molar concentrations of unbound A_+, A_- and bound AA, respectively. Let N be Avogadro's number, and V be the volume of the system, the above equation can be rewritten in the unit of number of binding sites:

$$\frac{d\frac{<AA>}{NV}}{dt} = k\frac{< A_+ >}{NV}\frac{< A_- >}{NV} \tag{B.2}$$

Or:

$$\frac{d < AA >}{dt} = \frac{k}{NV} < A_+ >< A_- > \tag{B.3}$$

Here $< A_+ >, < A_- >, < AA >$ are numbers of unbound A_+, A_- and bound AA, respectively. In DESSA, for a binding reaction with expected time T, the expected waiting time for firing a reaction with given number of molecules would be $\frac{T}{<A_+><A_->}$, whose inversion would be the time-based reaction rate:

$$\frac{d < AA >}{dt} = \frac{< A_+ >< A_- >}{T} \tag{B.4}$$

Comparing Equations B.3 and B.4, we have:

$$k = \frac{NV}{T} \tag{B.5}$$

B.2 Homogeneous binding

For a heterogeneous binding reaction between binding sites A_0 and itself with rate constant k:

$$A_0 + A_0 \xrightarrow{k} AA$$

The molar reaction rate is defined as:

$$\frac{d[AA]}{dt} = \frac{k}{2}[A_0][A_0] \tag{B.6}$$

Similar to the approach in the heterogeneous case, the above equation can be rewritten in the unit of number of binding sites:

$$\frac{d\frac{<AA>}{NV}}{dt} = \frac{k}{2}\frac{< A_0 >}{NV}\frac{< A_0 > -1}{NV} \tag{B.7}$$

Or:

$$\frac{d < AA >}{dt} = \frac{k}{2NV} < A_0 > (< A_0 > -1) \tag{B.8}$$

Note that due to the symmetry of this reaction, there are only $\frac{<A_0>(<A_0>-1)}{2}$ pairs of A_0 to trigger possible reactions. The time-based rate is:

$$\frac{d < AA >}{dt} = \frac{< A_0 > (< A_0 > -1)}{2T} \tag{B.9}$$

Again:

$$k = \frac{NV}{T} \tag{B.10}$$

B.3 Conclusion

Comparing Equations B.5 and B.10, we may conclude that the binding rate constant is $\frac{NV}{T}$ for both heterogeneous and homogeneous cases. The volume can be determined by the concentration of a specie in experiment and number of corresponding specie in simulation, for example, once we know the concentration and number of subunits, the volume is:

$$V = \frac{< S >}{N[S]} \tag{B.11}$$

Taking the above equation into $k = \frac{NV}{T}$, we finally have the relation between rate constant and time:

$$k = \frac{< S >}{T[S]} \tag{B.12}$$

Note that the information of concentration is necessary for deriving rate constants, but concentration becomes a degree of freedom when generating synthetic data. In this case, artificial concentrations are chosen for more comprehensive meaning of the parameters, while DESSA does not require such information to operate.

When working with multiple concentrations simultaneously, it is necessary to assure a coherent binding rate across simulations under different concentrations. As Equation B.12 suggests, there are three methods to enforce the coherence against change in concentration:

1. Scale $< S >$ proportional to $[S]$.
2. Scale T proportional to $\frac{1}{[S]}$.
3. Scale $\frac{<S>}{T}$ proportional to $[S]$.

The second method is most favorable for its simplicity, fast simulation and low memory occupancy.

Appendix C

Likelihood of a trajectory

A trajectory of stochastic simulation is generated with respect to the probability of firing such series of reactions. In this section, a likelihood model will be derived based on a simple reversible dimerization reaction described by a forward rate k_1 and reverse rate k_2:

$$A + B \underset{k_2}{\overset{k_1}{\rightleftarrows}} AB$$

The probability of firing a reaction at the i^{th} step is:

$$p(i) = k(i)n(i) \exp\big(- k_1 n_A(i) n_B(i) t(i) - k_2 n_{AB}(i) t(i)\big) \tag{C.1}$$

Here $n_A(i), n_B(i), n_{AB}(i)$ are number of A, B, AB particles at the i^{th} step, respectively. The values of $k(i)$ and $n(i)$ depend on the type of the i^{th} reaction: $k(i) = k_1, n(i) = n_A(i)n_B(i)$ if the reaction is forward, or $k(i) = k_2, n(i) = n_{AB}(i)$ if it is backward. The last variable $t(i)$ is the time gap between the $(i-1)^{th}$ and i^{th} step. In total, the likelihood of a trajectory can be derived as a function of rates:

$$L(k_1, k_2) = \prod_i p(i) = \prod_i \Big(k(i)n(i) \exp\big(- k_1 n_A(i) n_B(i) t(i) - k_2 n_{AB}(i) t(i)\big)\Big) \tag{C.2}$$

And the log-likelihood is:

$$\begin{aligned} \ell(k_1, k_2) &= \sum_i \log k(i) + \sum_i \log n(i) - k_1 \sum_i n_A(i) n_B(i) t(i) - k_2 \sum_i n_{AB}(i) t(i) \\ &= r_1 \log k_1 + r_2 \log k_2 + L_N - k_1 N_1 - k_2 N_2 \end{aligned} \tag{C.3}$$

Here r_1, r_2 are number of forward and backward reactions, respectively. When a trajectory is given, the values of $r_1, r_2, L_N = \sum_i \log n(i), N_1 = \sum_i n_A(i) n_B(i) t(i), N_2 = \sum_i n_{AB}(i) t(i)$ become constant, and the trajectory can therefore be characterized by the five constants.

The main purpose of deriving the likelihood model is to reduce the number of trajectory simulations, which is the most time-consuming step in the parameter inference framework. Under the current framework, a trajectory is only used to evaluate the objective function at its producing set of parameters, but probabilistically such trajectory may also be produced by other sets of parameters with different levels of likelihood. This likelihood model makes it possible to use one trajectory to evaluate multiple sets of parameters, which may lead to the reduction of simulation runs.

The trajectory likelihood model may also shed light on the derivation of the likelihood of objective function value given a set of parameters, as the objective value is calculated through fitting the trajectories. The likelihood model of objective value may contribute to better approximation of the objective function over parameter space.