COARSE GRAINING PARTICLE SIMULATIONS OF BIOLOGICAL SYSTEMS

Exemplified on PML Nuclear Bodies and the SAC

RICHARD HENZE

FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Dissertation zur Erlangung des Akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich Schiller Universität Jena

von Richard Henze
geboren am 12. Nov. 1989 in Weimar
March 2020

betreuer: Prof. Dr. Peter Dittrich

gutachter:

(1.) Prof. Peter Dittrich, FSU Jena

(2.) Prof. Manja Marz, FSU Jena

(3.) PD Dr. Peter Hemmerich, FLI Jena

tag der öffentlichen verteidigung: 3. Februar 2020

ABSTRACT

Simulating biological systems *in-silico* has become a common method to provide deeper insights into the regarding systems than *in-vitro* investigations. Classic approaches like differential equations, Boolean networks or Markov chains are performant, but usually cannot express often desired spatial features. For that reason particle based simulators like Smoldyn, GFRD or ReaDDy came up that are able to examine a reaction network in space, but usually not on a large timescale. Due to methodological and instrumental restrictions it still is elusive to simulate complex systems in full detail over a large period of time. Coarse-graining methods allow the reduction of a high-detailed system into a little-detail system, whereby the qualitative behavior of the simulated systems is conserved. The aim of this thesis is to derive techniques that allow a detailed simulation on a large timescale. For that reason, firstly, a pathway is developed that translates a reaction network between a set of species into space using properties from literature, like mass and diffusion coefficient of all species. Secondly, coarse-graining methods are developed that are automatized applicable to real biological systems. These methods allow the simulation of the generated particle based model in a feasible amount of time, whereby the focus lies on the reduction of the simulation complexity rather than the model's complexity. Thirdly, a novel simulation tool is established that allows a simplified study of self assembling processes by coarse-graining the diffusion.

To study these methods at work, two models of biological systems emerged in the scope of this thesis, namely the spindle assembly checkpoint and PML nuclear bodies. The spindle assembly checkpoint (SAC) guards proper cell division by prolonging the metaphase until all 92 sister chromatids are aligned properly. Even a single unattached kinetochore keeps the SAC active, which is rapidly inactivated after the last attachment. Here, the first detailed biochemical reaction network is developed that explains the activation and inactivation of the SAC, considering all 92 human kinetochores and all involved major proteins. Using the novel methods spatially-stochastic simulations were performed, allowing the realization of ~ 20 real-life minutes. Results of this study support the hypothesis that the SAC signal varies with increasing number of attached kinetochores.

An example of a self-organized spatial assembling are PML nuclear bodies, which are dot-like structures found in the interchromosomale space of most mammals. While their function is mostly known, it is widely unclear how they are involved in the regarding processes. Nonetheless, their life cycle is of interest as they assemble and disassemble during different cell-phases, meaning that the structure of PML bodies is linked to their function. In this thesis, the first spatial rule-based model was developed that explains the formation and life cycle of PML bodies in molecular detail. Next to the wild-type simulation several mutation experiments were performed, validating the model with wet-lab results. Additionally, artificial PML proteins were designed *in-silico* and *in-vitro* to investigate the minimal requirements for the self assembling process.

ZUSAMMENFASSUNG

Die Simulation biologischer Systeme *in-silico* ist eine gängige Methode geworden um tiefere Einblicke in die entsprechenden Systeme zu erlangen, als *in-vitro* Studien es erlauben. Klassische Ansätze, wie Differentialgleichungen, Boolsche Netzwerke oder Markow-Ketten sind performant, können aber in der Regel keine, oft wünschenswerten, räumlichen Eigenschaften repräsentieren. Aus diesem Grund sind partikelbasierte Simulatoren wie Smoldyn, GFRD oder ReaDDy entwickelt wurden, welche Reaktionsnetzwerke räumlich simulieren, normalerweise jedoch nicht auf einer großen Zeitskala. Aufgrund dieser methodologischen und instrumentellen Beschränkungen ist es weiterhin schwierig ein komplexes System in vollem Detailgrad über einen langen Zeitraum zu simulieren. Coarse-graining Methoden erlauben die Reduktion eines Systems mit hohem Detailgrad zu einem mit niedrigem Detailgrad, wobei das qualitative Verhalten des simulierten Systems konserviert wird. Ziel dieser Arbeit ist es Techniken zu entwickeln, welche die Simulation auf einer großen Zeitskala ermöglichen. Dafür wurde erstens eine Abfolge von Methoden entwickelt, welche ein Reaktionsnetzwerk zwischen einer Menge an Spezies mit räumlichen Eigenschaften, wie Masse oder Diffusion, aus der Literatur ausstattet. Zweitens werden coarse-graining Algorithmen entwickelt, welche automatisch auf reale biologische Systeme anwendbar sind. Diese Methoden erlauben die Simulation der generierten partikelbasierten Modelle in einer praktikablen Zeit, wobei der Fokus der Reduktion auf der Systemkomplexität und nicht auf der Modellkomplexität liegt. Drittens ist ein neuartiges Simulationswerkzeug entwickelt worden, welches die freie Diffusion reduziert und dadurch die Untersuchung von Selbstassemblierungsprozeesen vereinfacht.

Um die hier entwickelten Methoden zu demonstrieren sind im Rahmen dieser Arbeit zwei Modelle biologischer Systeme entstanden, namentlich der mitotische Kontrollpunkt und PML Kernkörperchen. Der mitotische Kontrollpunkt (SAC) reguliert die ordentliche Zellteilung, indem die Metaphase so lange verlängert wird, bis alle 92 Schwesterchromosomen richtig angeordnet sind. Selbst ein einziger ungebundener Kinetochor hält den SAC aktiv, welcher nach der letzten Bindung eines Kinetochors deaktiviert wird. Hier wurde das erste detaillierte biochemische Reaktionsnetzwerk entwickelt, welches unter Berücksichtigung aller bedeutenden Proteine sowie 92 menschlicher Kinetochors die Aktivierung und Deaktivierung des SACs erklärt. Unter Anwendung der neuartigen Methoden sind räumlich-stochastische Simulationen durchgeführt worden, welche eine Realisierung von ~ 20 Realminuten ermöglichen. Ergebnisse dieser Studie unterstützen eine Hypothese, welche behauptet, dass die Stärke des SAC Signals von der Anzahl der gebundenen Kinetochors abhängt.

Beispiel für eine selbstorganisierte Assemblierung sind PML Kernkörperchen. Diese punktartigen Strukturen sind befinden sich im inter-chromosomalen Raum der meisten Säugetiere. Während ihre Funktion weitestgehend bekannt ist, ist ihre genaue Funktionsweise weiterhin nicht vollständig aufgeklärt. Nichtsdestotrotz, ihr Lebenszyklus ist von besonderem Interesse, da sie sich in unterschiedlichen Zellphase auf- und abbauen, d. h. die Struktur von PML Körpern hängt mit ihrer Funktion zusammen. In dieser Arbeit wurde das erste räumliche, regelbasierte Modell entwickelt, welches die Formation und den Lebenszyklus von PML Kernkörpern auf molekularer Ebene erklärt. Neben der Wildtyp Simulation sind zahlreiche Mutationsexperimente durchgeführt worden, welche das Modell gegen die Labordaten validieren. Weiterhin sind künstliche PML Proteine *in-silico* und *in-vitro* designt wurden, um die minimalen Bedingungen für den Assemblierungsprozess zu erforschen.

PUBLICATIONS

The full model of the spindle assembly checkpoint and its simulation, coarse-graining methods applied to the SAC model, as well as a guideline to the SRSim software have been published previously in the following:

[HDI17] R Henze, P Dittrich, and B Ibrahim. “A Dynamical Model for Activating and Silencing the Mitotic Checkpoint”. In: *Scientific Reports* 7.1 (2017), p. 3865. DOI: 10.1038/s41598-017-04218-2.

[Hen+18a] R Henze et al. “Multi-scale stochastic organization-oriented coarse-graining exemplified on the human mitotic checkpoint”. In: *Scientific Reports* (2018). *accepted*.

[Hen+18b] R Henze, G Gruenert, B Ibrahim, and P Dittrich. *Spatial Rule-based Simulations: the SRSim Software*. Ed. by William Hlavacek. *to appear*. Springer, 2018.

Parts of the analysis tools and DiCoSAD are publicly available:

[Hen18a] R Henze. *DiCoSAD: Particle Simulator that coarse-grains the Diffusion*. https://github.com/DonNeoMir/DICOSAD. Version 1.0. 2018.

[Hen18b] R Henze. *VirtualMicroscopy: Visualizes Particle Configurations*. https://github.com/DonNeoMir/VirtualMicroscopy. Version 1.0. 2018.

In addition to these official publications all the method and tool source codes introduced here are available in the digital appendix. All models and simulations are also attached in the digital appendix.

It doesn't matter how beautiful your theory is,
it doesn't matter how smart you are.
If it doesn't agree with experiment, it's wrong.

– Richard P. Feynman

ACKNOWLEDGMENTS

There are many people I would like to thank who have accompanied me on my way to this thesis. First of all I want to thank my supervisor Peter Dittrich, who always came up with creative ideas and solutions, and made my time in his group an amazing experience with his enthusiastic capacity for all aspects of science.

Peter Hemmerich, my second supervisor, delivered the biological background, provided his data and also offered the opportunity to visit his research facility, which gave me a great insight into the world of biology and the investigation of an example system.

Furthermore, I want to thank the colleagues and friends in the BioSystemAnalysis group that always were ready for a scientific talk but also to provide a distraction from work, namely, Bashar Ibrahim, Jakob Fischer, Christoph Neu, Stephan Richter, Jan Huwald and Gerd Grünert. Great gratitude goes to the technical support provided by Erik Braun and Uwe Richter, which relentlessly kept our systems running.

Last but not least I am thankful for all the people in my life outside the office that have kept me going and provided variety from everyday working life: friends, teammates, my wonderful parents, my sister and my dear fiancée Nelly.

Financial support: This thesis was financially supported by the "Nachwuchsförderprogramm" of the Carl-Zeiss foundation, the European HIERATIC project [1] FP7 (DyM-CS) and the Friedrich Schiller university.

Tools: This document was typeset using the `classicthesis` template developed by André Miede, available for both LaTeXand LyX[2]. Conducting the research presented in this thesis would have been inconceivable without the help of plenty free-to-use software of which I would like to name a few and thank their developers: ReADDy (Reaction-Diffusion Particle Simulator), SRSim (Spatial Rule-Based Simulator), Copasi (ODE Integrator), XPPAUT/MAN (ODE Solver and Bifurcation Analysis), Inkscape (Vector Graphics), GIMP (Image Program) and SGE (Grid Computing).

[1] http://www.cs.bham.ac.uk/research/projects/hieratic/

[2] http://www.miede.de/#classicthesis

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

APC/C Anaphase Promoting Complex in Cytosol

BCC	Bub Checkpoint Complex
BubR1	BUB1-related protein 1
Cdc20	Cell-Division Cycle protein 20
DAXX	Death-Associated protein 6
DiCoSAD	Disperse Component Self Assembly Dynamics
DNA	Desoxyribonucleinacid
GFRD	Green Function Reaction Dynamics
Mad2	Mitotic Arrest Deficient 2
MCC	Mitotic Checkpoint Complex
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PML	Promyelocytic Leukemia Protein
ReaDDy	Reaction Diffusion Dynamics
SAC	Spindle Assembly Checkpoint
SIM	SUMO Interactive Motif
Smoldyn	Smoluchowski Dynamics
SRSim	Spatial Rule-based Simulation
SUMO	Small Ubiquitin-like Molecule

1 INTRODUCTION

This thesis proposes novel coarse-graining methods that allow a particle simulation of large systems over a long duration and applies them to two concrete biological models. PML nuclear bodies and the mitotic checkpoint are linked to multiple fatal diseases in human, from where the necessity arises to analyze those systems also *in-silico*. Furthermore, established modeling and simulation approaches are introduced.

1.1 AIMS AND OVERVIEW

Investigating biological systems has shifted during the last century from purely wet-lab examinations to *in-silico* studies. The rise of computers with their power of calculation, and the development of modeling techniques and their simulation have become important tools to understand complex systems. However, with the increase of calculation power, the complexity of problems is increasing. While in the 1970s it was a novelty to simulate 10 particles undergoing a couple of reactions [Gil77], in the 2018 it is possible to realize the motion of trillions of particles [PST17]. Despite these tremendous improvements, it is still challenging simulating biological processes as they can typically take from minutes to many hours ($\sim 10^4$s, [Phi+12]), meaning that even a single particle would require 10^{13} simulated nano-second time-steps. A modern Intel i7 CPU can perform up to 375 GFlops, which are around $\sim 10^{11}$ operations per second (state, March 5, 2020), taking a couple of minutes to simulate a single particle for several hours. This problem of the large varying time-scales combined with the complexity of the model and the amount of interacting particles, means that a realization is only possible when performed on a professional parallel cluster. From this point of view it becomes clear that calculation power is not everything, but also the methods and algorithms have to be improved, to simulate larger and more complex systems.

The aim of this thesis is to advance the methodology of understanding complex systems. This is achieved by the development and introduction of multiple methods that coarse-grain the simulation of a process automatically, so that even several hours of real time can be simulated within a couple of hours. Coarse-graining methods find a mapping from the complex original system onto an abstract version that firstly behaves similarly, secondly is faster to simulate, and thirdly, ideally can be mapped back to the original system, which often is not possible (cf. Fig. 1.1). Here developed methods focus on the reduction of particles, space and time. Coarse-graining the diffusion is an entirely novel approach and for this reason the software package DiCoSAD evolved that uses an event driven algorithm to reduce free diffusion. These methods can be applied on a broad range of complex systems and allows their study without the need of a high-end computer cluster, whereby the quantitative outcomes are comparable to the wet-lab measurements. After a formal introduction, two concrete biological systems are investigated, namely the spindle assembly checkpoint and PML nuclear bodies. Guarding proper cell division and preventing diseases like aneuploidy or cancer is achieved by the spindle assembly checkpoint (SAC). The SAC is an evolutionarily conserved mechanism, exclusively sensitive to the states of kinetochores that are already attached to microtubules. Transiting from metaphase to anaphase has to be delayed by the SAC until all kinetochores are attached and then rapidly has to disengage. Phenomenologically it is a switching mechanism, sensitive to the state of attached sister chromatids. During metaphase all chromosomes are aligned in the center of the nucleus, showing also the spatial influence, as a single unattached chromatid holds the SAC active for several hours. Those properties make the SAC an interesting field of research for biologists as well as computer scientists, because spatial distributions, as well as a correct timing, matter to achieve a functioning switch mechanism that in the end prevents cancer. Over the past decades scientists discovered SAC elements, namely the concentration of key

Every chromosome is composed of two sister-chromatids that are pulled apart on their kinetochore.

$$\begin{tikzcd}
X \arrow[r, "\Phi_t"] \arrow[d, "\Xi"'] & X \arrow[d, "\Xi"'] \\
Y \arrow[r, "\Psi_t"] & Y
\end{tikzcd}$$

X: State space with dynamics Φ

Y: State space with dynamics Ψ

Ξ: Coarse-graining $X \to Y$

Figure 1.1: Illustration of a coarse-graining. (X, Φ) and (Y, Ψ) are dynamical systems with a mapping Ξ between them. Map Ξ then is a coarse-graining, if the diagram on the left commutes for all t, meaning that $\Xi(\Phi_t(x)) = \Psi_t(\Xi(x))$ (definition from [Goo+14]).

species, reaction rates, and the timely appearances of events that help to generally understand the SAC, but which do not expose all of its mechanisms, as they are demanding to discover in living cells. For that reason, multiple models emerged, trying to explain these unrevealed SAC features in more detail. This thesis works towards the full understanding of the SAC as a new model is presented, realized as system of ODE (cf. Sec. 1.2.1) and translated to a particle based model (cf. Sec. 2.1). Those models then are simulated using the techniques introduced in the first and second part (cf. Sec. 1.2.1, 1.2.2 and 2.2). Furthermore, an extensive analysis of the switching behavior is performed, revealing the mechanisms guaranteeing a proper switch (cf. Sec. 3.1.3).

In addition to the complexity resulting from a biochemical model and the amount of particles, another one arises from combinatorics, as seen for example in PML nuclear bodies. These $0.2 - 1.2\mu m$ dot-like structures, found in the interchromosomal territories of the nuclei of some cells, are composed of six main components. Still, it is generally unclear how those basic blocks assemble nuclear bodies, but it is believed to be a self-assembling process. Modeling those kind of behaviors benefits from a rule-based approach whereby the state space is too large to be generated a-priori. Rule-based methods are usually the way to go if the description of the interactions is less complex than the system itself. Simulating self-assembling systems in a rule-based manner has the downside that growing protein complexes have a decreased diffusion coefficient and thus become more immobile. For that reason a lot of calculation power is consumed by the Brownian Motion while bringing large clusters close together to be able to interact. Nevertheless, in the scope of this thesis the first spatial model is developed that explains the natural formation of PML nuclear bodies, and validates it against wet-lab mutational experiments using the methods introduced in Section 1.2.2 and 2.1.

Usually particle simulations are powered by a random walk of particles, named Brownian Motion.

1.2 PRELIMINARIES AND BACKGROUND TO SIMULATION APPROACHES

Over the past decades several ideas to investigate the behavior of systems have emerged [MN10; Lac16; And+10]. The quality of an approach depends on the focus of interest. Basically those methods are classified in deterministic or stochastic, and particle or non-particle based. Deterministic systems usually perform well, but lack detail, compared to stochastic ones, that are more realistic. Particle approaches provide the highest grade of detail but are more expensive in terms of calculation power. The following sections will distinguish them further and introduce different approaches and softwares, implementing and simulating them.

1.2.1 *Non-Particle Approaches*

Markov Chains and the Master Equation

Section 1.2.2 discusses the Brownian motion in more detail.

Generally, a Markov Chain with continuous time is referred to as Markov Process.

This often is referred to as memorylessness.

A Markov chain describes a stochastic process and can be depicted as a directed weighted graph, whereby nodes correspond to a state of the system and the directed edges are properties to transit into the regarding state [Mar71]. One of the most important Markov Chains is the Brownian motion [Hid80], key to any particle simulation. Each node of the Markov chain corresponds to a state of the system and not to a specific species, whereby states can represent literally any state of a system such as temperature, velocity, phosphorylation status of a molecule etc. Time, as well as the state space, can be either continuous or discrete, but most simulations use a discrete state and time, as it is sufficient and easier to treat compared with continuous spaces. States during time-step t are denoted as S_t and it is $t \in \{1, 2, ..., T\}$ a countable set and $S_t \in \{1, 2, ..., N\}$ the state-space. Markov chains describe a stochastic process that fulfills the Markov property [MN10], meaning that future states only depend on the current one:

$$P(X_{t+1} = S_{t+1} \mid X_t = S_t, ..., X_0 = S_0) = P(X_{t+1} = S_{t+1} \mid X_t = S_t). \tag{1.1}$$

The probability that the system X will translate into state S_{t+1} depends on the current state S_t and not on previous states $S_{t-1}, ..., S_0$.

Simulating Markov chains requires an initial state of the system and the transition-matrix, describing the probabilities to change the current state. With a discrete set of reachable states and a given start probability distribution of all states ($P(X_0 = i)$) it is possible to determine the probability to be in a certain state at time t with iterative matrix multiplications.

$$\begin{bmatrix} P(X_t = 1) \\ \vdots \\ P(X_t = N) \end{bmatrix} =$$

$$\begin{bmatrix} P(X_{t+1}=1 \mid X_t=1) & \dots & P(X_{t+1}=1 \mid X_t=n) \\ \vdots & \ddots & \vdots \\ P(X_{t+1}=N \mid X_t=1) & \dots & P(X_{t+1}=N \mid X_t=N) \end{bmatrix}^t \begin{bmatrix} P(X_0=1) \\ \vdots \\ P(X_0=N) \end{bmatrix} \tag{1.2}$$

In matrix notation this equation simplifies to the following:

$$\vec{\Phi}(t) = \mathbf{M}^t \vec{\Phi}(0) \tag{1.3}$$

Whereby $\vec{\Phi}(t)$ denotes the probability vector to be in a certain state at time-step t and $\mathbf{M}$ refers to the state transition matrix. With this equation it is possible to determine the probability of a certain state for any time-step and also for the steady states by estimating the limit of the matrix iteration.

The computational complexity of the Markov approach is estimated by the one of matrix-vector multiplications, which is $\mathcal{O}(N^2)$ with N denoting the state space of the system that can go to the millions for only a few species.

Markov chains or processes are a specification of the Master equation [VH55]. This equation describes the probability of a system to be in a certain state in continuous time. Thus, Equation 1.3 transforms to the following (cf. App. A.2).

$$\frac{d\vec{\Phi}}{dt} = (\mathbf{M} - \mathbf{J})\vec{\Phi} \tag{1.4}$$

whereby $\mathbf{J}$ denotes the matrix of ones and $\mathbf{M}$ is the matrix from Equation 1.3. If the matrix $\mathbf{M}$ is regular the solution for $\vec{\Phi}(t)$ is given as follows:

$$\vec{\Phi}(t) = \sum_{i=1}^{n} c_i e^{\lambda_i t} \vec{v}_i, \tag{1.5}$$

whereby λ_i and $\vec{v}_i$ refer to the Eigenvalue and Eigenvector of $\mathbf{M}$, respectively. From the starting conditions follow the constants c_i. Determining eigenvalues analytically is difficult and there exists no general form. Algorithms that approximate eigenvalues have a practical complexity of $\mathcal{O}(N^3)$, N denoting the state space of the system.

As the matrix $\mathbf{M}$ contains state transition probabilities, it follows that:

$$\sum_j \mathbf{M}_{ij} = 1 \qquad \forall i \tag{1.6}$$

implicating that the matrix is not regular and thus at least one eigen-

This simplifies the estimation of other Eigenvalues.

value λ_i is zero.

Another difference to Markov chains is that the transition matrix $\mathbf{M}$ can be time dependent in a Master equation:

$$\mathbf{M} \rightarrow \mathbf{M}(t). \tag{1.7}$$

In this case a general solution appears impossible to find.

Markov chains and the Master equation allow the exact analysis of complex systems if the state transitions are known, and find applications in a wide range of fields, e.g. game theory, sports or biology. Nevertheless, they also can be used to simulate reaction networks, assuming a state as vector of the current concentration of all species, the transition between states then is the change of concentrations with a probability, corresponding to the reaction rate of the regarding reaction.

In a real (biological) system the number of states can be enormous, which makes the transition matrix $\mathbf{M}$ large. Markov chains as well as the Master equation are not performant for those systems. Also, spatial features cannot be specified with those methods.

Ordinary Differential Equations of Species

Previously introduced methods focus on the transition of states of the system. As shown, those methods are unsuitable for simulating reaction networks quantitative, as they are computationally not performant. A more convenient method is to describe the changes in the systems basic blocks, rather than the probability to be in a certain state. Methodically it does not change anything, but a system of ordinary differential equations describes the time course of n species undergoing r reactions and not probabilities changes between N states [Lac16]. Generally the form of the ith reaction is given as follows:

In a biological reaction network, basic building blocks generally are species that are usually proteins or other complexes.

$$c_1^i R_1^i + c_2^i R_2^i + ... + c_n^i R_n^i \xrightarrow{k^i} d_1^i P_1^i + d_2^i P_2^i + ... + d_m^i P_n^i \tag{1.8}$$

$R_s = P_s$

Reactants $R_1^i, \dots, R_n^i$ react with rate k_{on}^i to products $P_1^i, \dots, P_n^i$. The values $c_1^i, \dots, c_n^i, d_1^i, \dots, d_m^i$ denote the stoichiometric factors. Note, that factors can be 0 if the regarding species is not present in this reaction.

A reaction network can then be denoted compact as time independent stoichiometric matrix:

$$\mathbf{N} = \begin{bmatrix} d_1^1 - c_1^1 & d_1^2 - c_1^2 & \dots & d_1^r - c_1^r \\ d_2^1 - c_2^1 & d_2^2 - c_2^2 & \dots & d_2^r - c_2^r \\ \vdots & \vdots & \ddots & \vdots \\ d_n^1 - c_n^1 & d_n^2 - c_n^2 & \dots & d_n^r - c_n^r \end{bmatrix} \tag{1.9}$$

whereby c_j^i and d_j^i represent the stoichiometric factors of species j in reaction i, and the time dependent flux vector:

$$\vec{v}(t) = \begin{bmatrix} L^1(k^1, R_1^1, \dots, R_n^1) \\ L^2(k^2, R_1^2, \dots, R_n^2) \\ \vdots \\ L^r(k^r, R_1^r, \dots, R_n^r) \end{bmatrix} \tag{1.10}$$

with k^i being the reaction constant, and the rate law $L^i(R_1^1, \dots, R_n^1)$ that applies for the regarding reaction i. Kinetic laws are defined independently for every reaction and are usually chosen as Mass-Action or Michaelis-Menten kinetics as often no further information about the reaction is known. Reversible reactions are usually defined as two irreversible reactions. The system of ODEs then can be derived easily as product of matrix **N** and the flux vector $\vec{v}(t)$

Usually rate laws depend on the concentration of the reactants, but also can be constant.

$$\frac{d\vec{S}(t)}{dt} = \mathbf{N}\vec{v}(t) \tag{1.11}$$

Here, $\vec{S}(t)$ is the vector of every species' concentration during timestep t. The ODE for a particular species $S_j(t)$ extends to:

$$\frac{dS_j}{dt} = \sum_{i=1}^{r} (d_j^i - c_j^i) L^j(k^i, R_1^i, \dots, R_n^i) \tag{1.12}$$

It is seen easily that this system is deterministic and, dependent on the rate law, nonlinear in the most cases and thus hard to solve analytically. For a single fusion process:

$$A + B \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} C \tag{1.13}$$

the exact steady state solutions for the three species A, B and C with initial concentrations A_0, B_0 and C_0, respectively, is the following:

$$[A]_S = \frac{1}{2}\left(A_0 - B_0 - \frac{k_{off}}{k_{on}} + \text{Dis}\right) \tag{1.14}$$

$$[B]_S = \frac{1}{2}\left(-A_0 + B_0 - \frac{k_{off}}{k_{on}} + \text{Dis}\right) \tag{1.15}$$

$$[C]_S = \frac{1}{2}\left(A_0 + B_0 + 2C_0 + \frac{k_{off}}{k_{on}} - \text{Dis}\right) \tag{1.16}$$

with $\text{Dis} = \frac{\sqrt{(A_0 k_{on} - B_0 k_{on} - k_{off})^2 + 4k_{on}k_{off}(A_0+C_0)}}{k_{on}}$ which is shown in Appendix A.1.

Even for this minimal model it is not simple to derive an analytical solution and the general form is not intuitive. From this point it is easy to see that an analytical solution from a real biological system is nearly impossible to derive. Anyway, solving the problem numerically is straightforward and plenty of algorithms are available to solve systems of ODEs. The downside of this solution is the transition from continuous space to discrete time-steps and the approximated solution. The time complexity for Runge-Kutta algorithms is $\mathcal{O}(nr)$ per step, which is significantly faster than solving the Master equation, as it solely depends on r, the number of reactions, and n, the amount of species, and not on the states of the system, that can be infinite. The Master equation on the other hand is stochastic and not deterministic, making it more realistic. Nevertheless, ODEs can be generalized by adding a noise term, modeling random fluctuations resulting in stochastic differential equations (SDEs, [GS79]). In conclusion, the transition from Master equation to ODE systems trades accuracy for a reduced calculation effort to derive an approximation of the time course of a biological model.

Most common are the broad range of Runge-Kutta Schemes that differ in the approximation error [JST81].

Partial Differential Equations of Species

ODEs only contain functions and derivations of one variable, giving them the name ordinary. In contrast, partial differential equations (PDE) can contain multiple variables and their partial derivations:

$$f(x, y, y', ..., y^{(n)}) \to f(x_1, \dots, x_n, y(x_1, \dots, x_n), \frac{dy}{dx_1}, \dots, \frac{dy}{dx_n}, \dots). \tag{1.17}$$

$\frac{dy}{dx} = f(x, y, y', ...)$ *whereby y denotes the concentration vector, and x the variable that usually corresponds to the time.*

With this flexibility it is possible to describe spatial parameters, like the extension of a reactor, velocity of species, temperature, acceleration etc. PDEs find wide range of applications in physics, like heat, sound or fluid distribution in media. In biological systems PDEs can be used to describe phenomena like convection, diffusion or simply the spatial distribution of proteins. Those equations typically look like:

$$\frac{dS}{dt} = D\Delta S - \vec{V}\nabla S + R(S). \tag{1.18}$$

S refers to a species' concentration, D is this species' diffusion constant (cf. Ficks second Law [Fic95]), $\vec{V}$ the velocity field and R describes the reactions where S is involved. Systems of PDEs then are used to describe the reaction network of several species (R), that are undergoing a random diffusion process (D), and a directed movement ($\vec{V}$).

R is comparable to the right-hand side of Equation 1.12.

Solving PDEs analytically is even harder than solving ODEs. Dependent on the form of the PDE, there exist some approaches of a closed form, but generally it is not possible to determine one. Most prominent algorithms to approximate the problem numerically are finite differences, finite elements and finite volumes [RST08; CIR52; ZT77] . Those methods discretize up to all the variables and compute from initial values on those grid-points approximations using neighboring values. Next to the size of the reaction network, those algorithms' complexity also depends on the chosen grid-size, as usually every grid-point needs to be evaluated at every time-step. The same way as ODEs, PDEs are generalized by adding a random white noise term, making the system stochastic (SPDE, [Wal86]).

Other Approaches

As well as the above introduced method, plenty of others exist to model and simulate biological systems. Some of them are Boolean, and some are petri nets that are mentioned briefly only for completeness [Kau69; Pet62], as they are not used in the scope of this thesis.

Boolean networks are in mathematical terms a directed graph, in which nodes correspond to a species, a gene, a protein or even molecules or only atoms and have a discrete value, often from the set $S = 0, 1$, corresponding to its absence or presence. Each node has input values from other nodes that is mapped by a Boolean function to an output value. The dynamic behavior of the system is investigated through a time discrete update, which can be either synchronous or asynchronous.

Petri nets are a bipartite graph that describes the functioning of a system. The nodes of the graph correlate to places and transitions between them, for what reason petri nets often are referred to as place/transition nets. Petri nets can be derived from boolean nets and are used to investigate the qualitative, as well as the quantitative behavior of complex systems. Despite these strong limitations, these networks have been used to analyze biological processes precisely [Ste+14; DB08; MCN08].

1.2.2 Particle Approaches

Markov chains also allow a particle based treatment.

Previously introduced methods to study biological systems treat species like a concentration, which is a simplification as molecules are discrete. The concentration of a species [S] is directly proportional to the amount of particles of species S_n via the volume V it is in.

$$[S] = \frac{S_n}{V} \tag{1.19}$$

It appears that only knowing the concentration of a species is not enough as the volume has to be known as well. Knowing the exact particle count on the other hand is unambiguous, and for this reason it seems more natural to treat species as particles. The other advantage of a particle based model is that it can easily be translated into space, where particles only need a coordinate, which is not possible with a concentration. To simulate a particle based model several approaches exist. Most of them have in common some basic concepts which are introduced below.

Following descriptions assume a spatial particle simulation.

Particles usually present a certain species that can be an atom, a molecule, a protein or technically any discrete structure. They have a position in a predefined reactor, with given boundary conditions. The force that drives particles motion is known as Brownian motion which in simulations is mimicked by a Wiener process resulting in a random walk of all particles [Hid80]

$$W_{t+\mu} - W_t \sim \mathcal{N}(0, \mu) \tag{1.20}$$

Equation 1.20 describes the time-continuous Wiener process with Gaussian increments, which is incorporated into Brownian Dynamics [EM78]:

$$\frac{d\vec{r}}{dt} = -D\frac{\nabla V(\vec{r})}{k_B T} + \sqrt{2D}W_t. \tag{1.21}$$

Solving this ODE is done numerically, usually by applying Euler's scheme (cf. Eq. 1.24). In addition to these time-discrete approximations, a few methods exist that are event based and allow an exact treatment of continuous time. Generally, these algorithms calculate stochastically when the next reaction is going to take place.

Some approaches allow a space exclusion by particles and give them a velocity or torque, resulting in a particle-particle interaction, which is one of the great benefits of a particle based model as these reactions can create spatial phenomena. In the following some approaches that have been used in this thesis are presented in more detail.

Gillespie

The algorithm published by Gillespie is not a classical particle simulation as it treats the system particle-wise but does not have spatial properties [Gil77]. Nevertheless, it generates an exact species over time trajectory because every individual reaction is performed stochastically. After the initialization of all species' molecules, two random numbers are generated every time-step. The first one determines which reaction occurs next, dependent on the number of molecules of a certain species, the second one determines when the chosen reaction happens, based on the reaction rates. Basically it is a dynamic Monte Carlo method that has every time-step a complexity of $\mathcal{O}(n)$ whereby n is the amount of species (note that the time-steps are not defined to a fixed size and thus the complexity is not comparable to other approaches).

Later a spatial version has been formulated [SL96].

Due to the fact that every reaction is carried out individually, Gillespies algorithm originally (1976) only could handle tens of molecules. Nowadays, as computational power increases steadily, it is heavily used as it delivers correct results and no numerical approximations [Dyk15].

Smoldyn

Smoldyn (Smoluchwoski Dynamics, [AB04]), extends the Smoluchowski model [Smo18], which is a system of ODEs that describes the flux and local concentration of point-like particles in continuous time. However, the Smoldyn software uses discrete time-steps and follows the classical particle paradigm. After the random placement of all particles, every particle undergoes free diffusion. If the interaction radius of two particles overlaps, a bimolecular reaction can happen. Smoldyn supports 0th to 2nd order interactions, meaning birth and death, conversion, or bimolecular reactions are possible. Smoldyn models can be simulated in one-, two- and three-dimensions, and they have been applied to a wide range of real world problems [And+10; LWH15], as it is quite performant and includes a lot of features such as membrane interactions or signal processing. It is still under development and the latest addition is the possibility of including BioNetGen and rule-based reactions (more about rule-based in Section 1.2.2).

Typically Smoldyn is used to investigate biophysical problems.

Green's Function Reaction Dynamics

Reaction-Diffusion models are many-body problems that do not have an analytical solution. The approach of the Green's Function Reaction Dynamics software (GFRD, [ZW05]) is to decompose the problem into one and two-body problems that have a general form given by Green's Function. Einstein's diffusion equation describes the probability for each particle to be at position $\vec{r}$ during time-step t, giving its origin $\vec{r}_0$ during time-step t_0:

$$\frac{\partial P(\vec{r}, t \mid \vec{r}_0, t_0)}{\partial t} = D\Delta P(\vec{r}, t \mid \vec{r}_0, t_0) \tag{1.22}$$

It can be shown that Equation 1.22 results in the well known mean-square displacement $6Dt$.

which coincides with the diffusive part of Equation 1.18. Green's function delivers a closed solution for the PDE 1.22:

$$P(\vec{r}, t \mid \vec{r}_0, t_0) = [4\pi D(t - t_0)]^{-\frac{3}{2}} \exp\left(-\frac{\mid \vec{r} - \vec{r}_0 \mid^2}{4D(t - t_0)}\right). \tag{1.23}$$

GFRDs decomposition is achieved through an event driven algorithm that only allows one- or two-body problems that can be solved analytically, with Equation 1.23 and additional terms for the first or second order reactions.

In terms of computational costs of a particle simulation, the Brownian motion is the most expensive. To overcome this, GFRD uses a Gillespian algorithm to estimate the next time a reaction could take place, based on every particle's diffusion coefficient. So created events are resolved exactly with Equation 1.23, and the system is propagated, meaning that all particles get re-sampled according to the calculated time-step. According to the authors, the achieved speedup is up to six magnitudes faster than conventional Brownian Dynamics.

ReaDDY

The ReaDDy software simulates Reaction Diffusion Dynamics [SN13] classicaly, similar to Smoldyn. It updates the position of every particle for discrete time-steps according to Brownian Dynamics:

$$\vec{r}_{t+\Delta t} = \vec{r}_t - D\frac{\nabla V(\vec{r}_t)}{k_B T} + \sqrt{2D\Delta t}\vec{\mathcal{N}}(0, 1) \tag{1.24}$$

Cf. Equation 1.21.

whereby V describes force fields, T the temperature, k_B the Boltzmann-constant and $\mathcal{N}(0, 1)$ a vector holding normal variables.

Contrast to previously introduced approaches is that ReaDDys particles have an extension, which implies a harmonic force between them if their interaction radii overlap. Further particle-particle potentials can be defined and implemented. Reactions between particles occur according to a defined reaction rate, once they overlap. The main purpose of ReaDDy is to study crowding effects in cellular environments [Sch+14].

SRSim

Biochemical reactions like $A + B \rightarrow C$ are modeled and simulated easily as ODE, as their state space is *a-priori* defined. Assembly processes such as the polymerization $A + A \rightarrow A.A$ are more challenging, as chains can reach infinite length. Approaching this with an ODE would require the definition of every chain-length, which would firstly limit the process, and secondly result in a giant model description. To overcome this hurdle rule-based modeling is used (e.g. BNGL [Bli+04], or Kappa [DL04]), which describes the reaction implicitly, when the reactions are easier to describe than the system they result in. Describing the above mentioned polymerization is done with only one rule, $A(\mathtt{free}) + A(\mathtt{free}) \rightarrow A(\mathtt{bound!1}).A(\mathtt{bound!1})$, namely that two different A molecules with one free binding site each can form a bond, independent of all other binding sites.

It appears that infinite length is a figure of speech, as infinite particles are required. The expression states that the resulting chain lengths are not known a-priori.

A spatial version of a rule-based simulation is the SRSim software [Gru+10], which has been introduced in Section 1.2.2. The spatial-rule-based simulator SRSim brings a new approach to particle dynamics which is the concept of rule-based reactions instead of chemical ones. Particle propagation and treatment is the same as in ReaDDy or

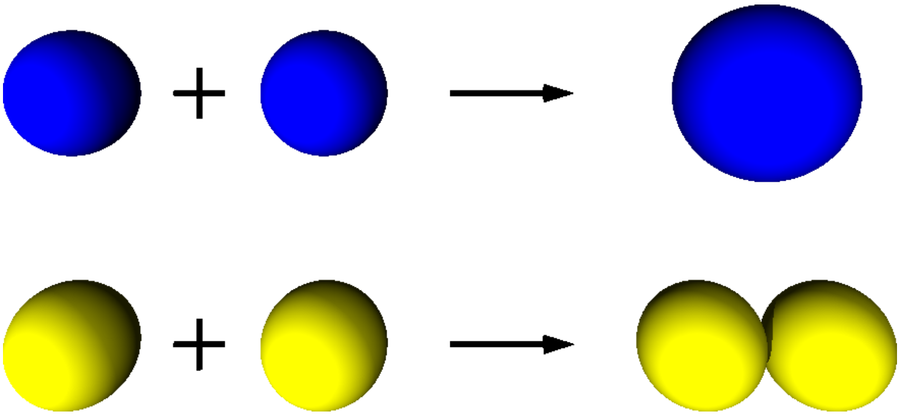


Figure 1.2: Comparison between conventional reaction system and rule-based systems in space. The blue interaction shows that two particles undergoing a conventional chemical reaction form another single particle. This reaction follows a probability distribution on a spatial encounter and is explicitly described *a-priori*. The reaction of the yellow particles shows a spatial rule-based approach, where particles undergo a rule to form a bond. If the rule can be applied depends on the probability distribution on encounter, as well as on the configuration of the binding sites of each particle (not shown). This is the major difference compared to a conventional chemistry, as the reactions are not listed *a-priori*.

any conventional approach, particles are updated discretely, undergoing a Brownian motion and additional potentials (cf. Eq. 1.24). Conventional particle simulating approaches allow chemical reactions within a predefined set of species, implying that no new species emerge. In contrast to reactions, rules are defined between regions of molecules and not the whole molecule. Carrying out a rule does not result in the transformation to another species but the formation of a bond between them.

These binding regions often are referred to as binding sites.

$$\text{conventional: } A + B \to C \tag{1.25}$$

$$\text{rule-based: } A + B \to A.B \tag{1.26}$$

Here, two species A and B do not react to species C but to the compound $A.B$ (cf. Fig 1.2). Assuming particles as spheres, the complex $A.B$ consist of two connected spheres and not of only one, as species C would. Those complex formations are called rules; the list of applicable rules replaces the list of explicit chemical reactions.

Usually, a rule-based system is chosen over a chemical one if the description of the model is less difficult than the system itself. Assuming a protein A, that can bind two molecules A, so A can form a chain polymer. Describing this conventionally requires a definition for each polymer of a different length. With the rule-based approach it is possible to generate the same model with only one rule, which is that one site of A can react with the site of another A.

There exist various rule-based systems (BNGL, or Kappa) but SRSim is the first one that brings the concept into space. Additionally, SRSim can apply forces to form a certain angle or even dihedral angle. With this feature SRSim focuses mainly on the structural analysis and the self-assembling of protein complexes [GD11; Hen+15].

1.2.3 *Conclusion*

This sections presents numerous modeling approaches to investigate complex systems. Usually, they are distinguished by their spatiality and stochasticity. In fact, the method to be chosen depends on those criteria and the aim of the model. Here introduced methods are used in the following chapters to simulate two concrete biological problems. Formulating a system can be done in several other ways, which are not introduced in this scope as they are unfitting for the here investigated systems.

Table 1.1 concludes and compares the introduced methods in respect to the system time, the simulation time treatment, the representation of species, spatiality and accuracy, with the following meaning of the used symbols:

$$\begin{aligned}
\text{Time} &\in \{\text{Continuous}, \text{Discrete}\} \\
\text{Time treatment} &\in \{\text{Discrete step}, \text{Fixed step}\} \\
\text{Species} &\in \{\text{Boolean}, \text{Concentration}, \text{Particle}, \text{State-vector}\} \\
\text{Space} &\in \{1\text{D}, 2\text{D}, 3\text{D}\} \\
\text{Accuracy} &\in \{\text{Exact}, \mathcal{O}()\} \\
\Delta h &\ \text{Space discretization} \\
\Delta t &\ \text{Time discretization} \\
d &\ \text{Spatial dimension}
\end{aligned}$$

The accuracy of the numerical approaches heavily relies on the used method, which usually is the Euler scheme, as it is the cheapest in terms of required calculations.

Table 1.1: **Comparison of state of the art simulation methods.**

Method	Time	Time Treatment	Species	Space	Accuracy	Published	Ref
Markov Chain	Continuous	Fixed Step	State-vector	-	Exact	1906	[Mar71]
Master Equation	Continuous	Fixed Step	Concentration	-	Exact	1955	[VH55]
ODEs	Discrete	Fixed Step	Concentration	-	$\mathcal{O}(\Delta t)$	1828	[Lac16]
PDEs	Discrete	Fixed Step	Concentration	1-3 D	$\mathcal{O}(t\Delta h^d)$	1770	-
Boolean Network	Discrete	Fixed Step	Boolean	-	-	1969	[Kau69]
Petri Nets	-	-	Discrete	-	-	1962	[Pet62]
Gillespie	Continuous	Discrete Step	Particle	-	Exact	1977	[Gil77]
GFRD	Continuous	Discrete Step	Particle	3D	Exact	2005	[ZW05]
Smoldyn	Discrete	Fixed Step	Particle	1-3 D	$\mathcal{O}(\Delta t)$	2004	[AB04]
Srsim	Discrete	Fixed Step	Particle	3D	$\mathcal{O}(\Delta t)$	2010	[Gru+10]
ReaDDy	Discrete	Fixed Step	Particle	1-3 D	$\mathcal{O}(\Delta t)$	2013	[SN13]
RichardSim	Discrete	Discrete Step	Particle	3D	$\mathcal{O}(\Delta t)$	2018	[Hen18a]

2

NOVEL METHODS TO COARSE-GRAIN PARTICLE SIMULATIONS

Firstly, a procedure is described that moves from an ODE model to a particle model. Secondly, automatable coarse-graining methods are developed that allow the simulation of the retrieved models. These methods are in further sections applied to a real-life system. Thirdly, a new software is introduced that allows the study of self-assembling processes in space due to its event driven algorithm.

In this section different coarse-graining approaches will be discussed that lead to the development of later introduced models and simulations, generally aiming for a reduction of dimensionality without losing the essential behavior of the system. Automatable methods are of special interest, as no information about the investigated system or its components is necessary, for what reason they are normally hard to find, unfortunately. Anyway, coarse-grainings do not have a general purpose, meaning that every issue has its own methods to reduce complexity, e.g polymers [Tsc+98], solid molecular dynamics [RB98] or the multi-scale protein folding simulation, which was even awarded the Nobel Prize in 2013 [LW75]. These examples show that no unique coarse-graining algorithms are available. In the following, some techniques are introduced and novel methods of coarse-grainings developed, that are applicable to a broad range of systems.

One of the most prominent coarse-graining approaches is a reduction of the model's complexity, whereby removing redundant species is the key, but the gain usually is minimal. Reducing a model effectively requires essential knowledge about its components and its functioning, because no crucial parts should be neglected.

For this reason most coarse-graining methods focus on reducing the complexity of the simulation rather than the model. Time discrete non-particle approaches usually are quite efficient and coarse-graining does not gain a lot in terms of accelerating the simulation, but nevertheless can be achieved by adapting the time-step. Time continuous models can hardly be accelerated. On the other hand, particle based models contain spatial information and are computationally very expensive, for what reason they are the focus of most coarse-graining studies. The next sections describe how to generate a particle model out of ODEs and will further develop methods to simulate them over a large time-scale.

2.1 FROM ODE TO PARTICLE MODEL

This method is used in Section 3.1.

Modeling a system with ODEs gives easily an insight into the behavior of the system but does not depict it realistically. An ODE mostly is deterministic and operates on a continuous concentration of species. More realistic is a particle model, as proteins or other macromolecules consist of atoms and molecules, meaning they are discrete instances, interacting in a stochastic matter. On the other side, more details of the model demand more computational effort, which makes it hard or even impossible to simulate a full system with every detail. Nevertheless, Section 2.2 shows how systems with a high grade of detail can be simulated for a sufficient amount of time. Now, this section provides a pathway of how an ODE model can be translated to a particle simulation. Requirements for a particle based model are the amount, shape, size and diffusion coefficient of all species, as well as their interaction rates.

2.1.1 *From Concentration to Particle Number*

The amount of a certain protein, or more general a species, usually is given as molar concentration, which is relative to the size of the reactor [MW97], meaning that 1mol in 10µl contains more particles than 1mol in 1µl. More desirable is an absolute amount that is independent of the reaction vessel. In 1811 Amedeo Avogadro postulated the relation between reaction volume, concentration and amount of particles of ideal gases [Avo11]. Avogadro formulated this relation in the following equation

$$N = cN_AV \tag{2.1}$$

whereby

$$\begin{aligned} N &= \text{amount of particles of a species} \\ c &= \text{concentration of the species in } \mu\text{M} \\ V &= \text{volume of the reactor in dm}^3 \\ N_A &= \text{Avogadros constant } (6.022 \times 10^{23}\text{mol}^{-1}) \end{aligned}$$

Assuming that the size of the reactor V, that was measured in, is known, Equation 2.1 allows the conversion from the concentration c to the amount of particles N.

2.1.2 *From Mass to Radius and Diffusion Coefficient*

ODEs can be seen as a well stirred reaction vessel with its species moving infinitely fast. While transitioning to a particle based model, all species need a spatial extension as well as a realistic diffusion coefficient. Databases usually provide the molecular mass of macromolecules, as they can be measured easily in wet-lab via mass spectrometry, gel-filtration [Dem18; And65] or even be calculated analytically, given their structure. Simplifying particles as spherical objects with equally distributed mass allows the calculation of the radius of the particle with a certain mass [Eri09]:

$$r_P = 0.066\sqrt[3]{m_P} \tag{2.2}$$

whereby m_P corresponds to the molecular mass of the spherical particle P in Da and r_P to the radius of the simplified particle in nm.

Wet-lab estimation of the diffusion coefficients of molecules is usually done by a fluorescence correlation spectroscopy (FCS, [Tho02]). Nevertheless, there is no complete database, as the coefficient depends on many factors, such as temperature, surrounding material, etc. The Stokes-Einstein equation is a special form of the Einstein relation that provides a mathematical model for the diffusion of spherical particles:

$$D_P = \frac{k_B T}{6\pi\eta r_P} \tag{2.3}$$

with

$$\begin{aligned}
D_P &= \text{Diffusion coefficient of particle P in } \frac{m^2}{s} \\
k_B &= \text{Boltzmann constant } (1.380 \times 10^{-23} J/K) \\
T &= \text{Temperature in K} \\
r_P &= \text{Radius of the assumed spherical-particle P in m} \\
\eta &= \text{Viscosity of the medium at 300K} \\
&\quad (0.891 \times 10^{-3}\frac{Ns}{m^2} \text{ in water, } 6.75 \times 10^{-3}\frac{Ns}{m^2} \text{ in nucleus}).
\end{aligned}$$

Young et. al. [YCB80] has shown that Equation 2.3 mostly correlates with wet-lab results, and for this reason it is possible to approximate the diffusion coefficient, based on a particle's radius that has been estimated initially.

Particles represent species with their relative size and diffusion coefficient, solely based on their molecular mass. Setting the reaction vessel according to the reaction volume conserves the concentration of the regarding species.

2.1.3 *From Macro to Micro Rates*

While the reactions of the ODE model can be inherited, the regarding reaction rates have to be adapted. The macroscopic reaction rate of an ODE reaction (k_{mac}) describes the amount of educts that are transformed within a fixed amount of time, which is formulated as time-dependent variable in the differential equation. This macroscopic behavior is not applicable in a particle simulation, as only discrete reactions happen with a certain probability. If two particles P_i and P_j encounter at distance d_{ij}, a microscopic reaction rate (k_{mic}, based on the macroscopic one) estimates the probability of a reaction. Erban et. al. [EC09] described the relation between macroscopic and microscopic reaction rates as follows:

$$k_{mac} = 4\pi(D_i + D_j)\left[d_{ij} - \sqrt{\frac{D_i + D_j}{k_{mic}}}\tanh\left(d_{ij}\sqrt{\frac{k_{mic}}{D_i + D_j}}\right)\right] \tag{2.4}$$

Microscopic rates then determine the Poisson distributed probability for a bi-molecular reaction to take place during the time interval Δt [Kam92].

$$P(\Delta t) = 1 - e^{-k_{mic}\Delta t} \tag{2.5}$$

Uni-molecular reaction rates do not have to be scaled as they happen spontaneously, while bi-molecular rates are not carried out deterministically but follow the probability given in Equation 2.5.

2.2 COARSE-GRAINING A PARTICLE SIMULATION

This method is used in Section 3.1.

Time-steps cannot be varied because large jumps in space lead to numerical instabilities (cf. Eq. 1.24).

Interesting coarse-graining methods are those for particle based models, as they consume far more calculation power. As most particle approaches are time discrete, controlling the time-step-size is one idea, which can not be varied freely as the diffusion limits this process. Easier to vary is the amount of particles, which is described in the following section. The simulated real-time can be modified by the reaction rates, which will be elaborated below.

2.2.1 *Reduce Amount of Particles*

Controlling the amount of simulated particles enhances the efficiency significantly. Particle-particle interactions are the outstanding benefit of particle simulations but also the most expensive ones, as the evaluation of those forces is of naive $\mathcal{O}(n^2)$ order and of order $\mathcal{O}(n \log n)$ in best case [Gri+13], n denoting the amount of particles.

Reducing particles is achieved by introducing so called pseudo particles, representing a lump of original particles (cf. Fig. 2.2 Left). These pseudo particles have an enlarged radius compared to their arisen particles, whilst the diffusion coefficient is the one from the original particles. In fact, some small particles are merged into one bigger one. While the granularity is reduced the simulation is kept intact, because the effective reaction rate is well adapted which is described below. Reactions of those pseudo particles represent multiple simultaneous reactions of the original particles at once.

Alternative the reactive volume could be conserved.

Now, let n be the number of original particles and N be the number of pseudo particles, with

$$n \gg N. \tag{2.6}$$

The spatial properties of a pseudo particle have to be adapted in a way that the half-life of all reactions is conserved (cf. Sec.2.1.3 for details on the connection of macro -and micro rate and spatial values).

Uni-molecular reactions have a half-life of $\frac{\ln(2)}{k_{mic/mac}}$ which is independent of the concentration of the educt and it is $k_{mac} = k_{mic}$ (cf. Sec. 2.1.3), and for that reason nothing has to be adapted as only the rate matters. The bimolecular reaction $A + A \xrightarrow{k} C$ has a half life of $\frac{1}{k_{mac}[A]_0}$, whereby $[A]_0$ is the initial concentration of A. If the amount of particles is halved by introducing $N = n/2$ pseudo particles the rate k_{mac} has to be doubled. This is not a problem for a deterministic ODE, but requires some adaption for the particle model where a micro rate k_{mic} is used and it is (cf. Eq. 2.4):

With the amount of particles also the concentration is halved.

$$k_{mac} = f(d_{ij}, D_i, D_j, k_{mic}). \tag{2.7}$$

To double the resulting macro rate could be achieved by varying any of the arguments D_i, D_j, d_{ij} or k_{mic}. Naturally, k_{mic} would be the intuitive choice, but it is a probability and $\lim_{k_{mic} \to \infty} f = 4\pi d_{ij}(D_i + D_j)$ a constant. For this reason K_{mic} is not a suitable parameter for scaling just as little as the diffusion coefficients due to its limitations (cf. Sec. 2.2.2. The last remaining option is the interaction radius, which has to be adapted for every bimolecular reaction in a way that the resulting

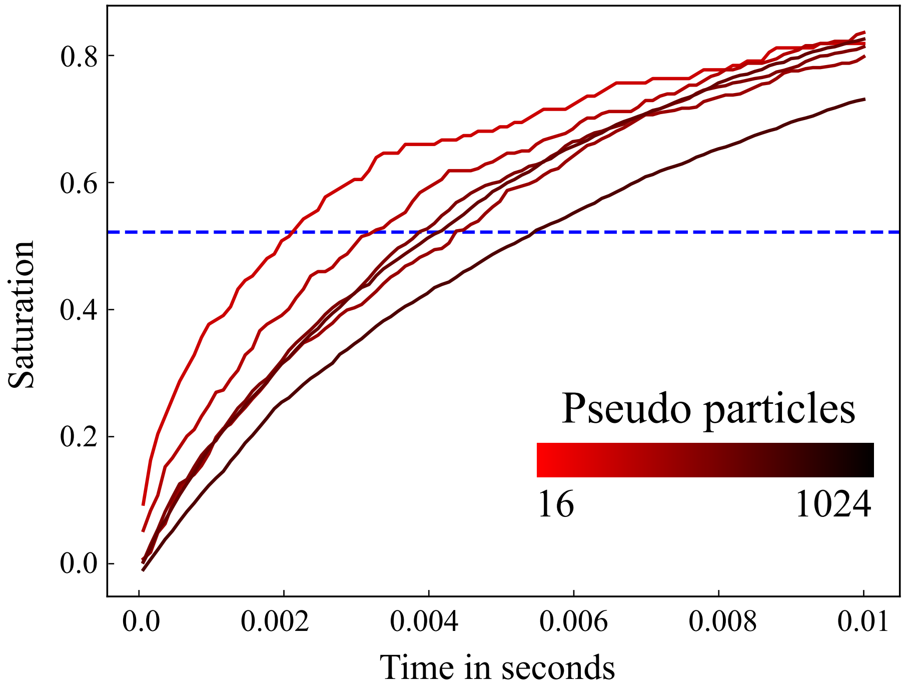


Figure 2.1: Activation over time of a species C undergoing a fusion process $A + B \rightarrow C$ with the half-time plotted in blue. Different color-labeled curves correspond to different amount of pseudo-particles, whereby the legend is shown in the lower right corner. All variations are averaged over 20 runs and show clearly that the amount of particles influences the half-time using the linear approximation of the interaction radius. To conserve the half-time precisely an optimization is required.

k_{mac} rate is increased by the factor that particles have been reduced:

$$\frac{n}{N} f(r_{ij}, D_i, D_j, k_{mic}) = f(R_{ij}, D_i, D_j, k_{mic}) \tag{2.8}$$

with r_{ij} and R_{ij} being the interaction radii of original and pseudo particles, respectively. Note, R_{ij} cannot be calculated analytically but can be estimated by a convex optimization.

Assuming the bimolecular reaction $A + B \rightarrow C$ (probably used more frequently) the half-life cannot be determined if $A[0] \neq B[0]$. For this reason the interaction radius is not known *a-priori* and has to be found by optimizing the simulation, which again rises a time issue. Approximating the interaction radius R_{ij} linear from the concentration (same scaling for $A + A \rightarrow C$) results in some inaccuracies, shown in Figure 2.1.

The particle reduction represents increased time-steps, without actually increasing them.

The amount of pseudo particles N is chosen freely, depending on the desired granularity of the outcome. Introducing pseudo particles is an automatable coarse-graining method that gains a lot, as the results do not lose much detail (cf. Fig. 2.2 Right). Only reactions are lumped together (cf. Fig. 2.1), while the simulation efficiency benefits

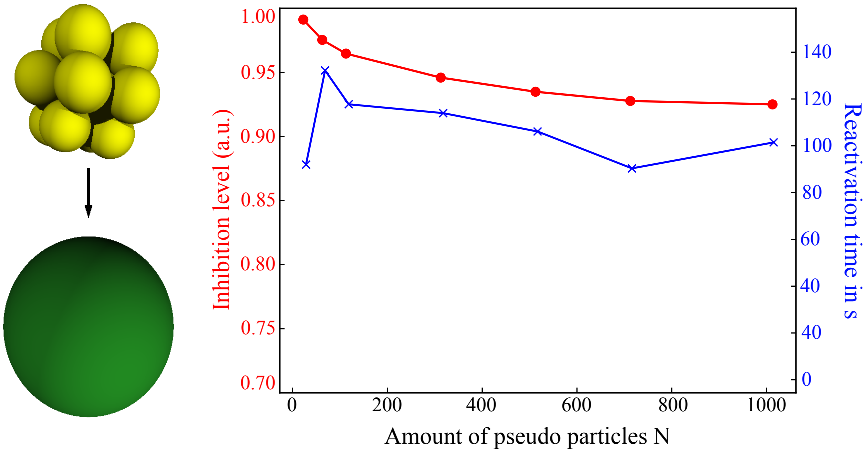


Figure 2.2: **Left:** Illustration of the particle reduction. $\frac{1{,}000}{N}$ original yellow particles are replaced with a green pseudo-particle with adapted radius, according to Equation 2.8. **Right:** Results of the Doncic model with a differing amount of pseudo particles N on the x-axis averaged over 5 runs [DBJB05]. The red graph shows the level of inhibition, a qualitative indicator of the model that should be above 90%. Apparently, the inhibition is strong, independent of the chosen amount of pseudo particles, indicating that the model yields the right behavior. The blue graph gives further proof based on the reactivation time of the model, which is another qualitative feature that could be constant. This example of a biological system shows how the qualitative behavior is conserved through the introduction of pseudo particles, while the consumed CPU time is radically decreased with decreased number of pseudo particles (times not shown). The fluctuation of the inhibition curve results from the approximation of the interaction radius.

tremendously. Note that if the amount of pseudo-particles is too little, the results of the simulation may be distorted. A general threshold of "too little" cannot be given as it depends on the simulated model.

2.2.2 *Reduce Time*

Biological processes within a cell take place on a timescale from femto-seconds to hours, e.g. the separation of photosynthetic charges takes pico-seconds [Was09] while mitosis lasts around one hour [CH97]. The focus of this thesis lies on the realization of the large time-scales. Most particle simulations update the position of every particle by adding a Brownian motion step. To keep the simulation stable a time-step of $\approx 10^{-9}$s is necessary (larger time-steps can lead to unexpected particle jumps). Thus, more than 10^{10} time-steps have to be executed for every particle to simulate one hour of real time. Assuming a small amount of particles would already end up with a need for immense calculation power, for a rather simple simulation. It

Particle simulations can hardly handle the simulation of hours or even minutes.

becomes obvious that parameter studies are elusive with these timescales. One way to study this system via a particle based model is explained in the following, based on a time and space dilation.

Steady state means negligible changes in concentration, as it technically would need infinite time to reach.

Assuming an ODE system, the time it takes to reach steady state is defined through the reaction rates. Increasing all rates by the same factor will accelerate the simulation by this factor, assuming an unchanged integration method. For a process that takes time T to approach steady state, a factor $c_t > 1$ is chosen to reduced the time it takes to $\hat{T}$:

$$\hat{T} = \frac{T}{c_t}, \text{ with } T \gg \hat{T}. \tag{2.9}$$

All reaction rates k_i of the system being adapted with the factor c_t to the rates $\hat{k}_i$ in the grained time:

$$\forall i : \hat{k}_i = c_t k_i. \tag{2.10}$$

The half-life $t_{1/2} = \frac{\log(2)}{k_i}$ solely depends on the rate k_i.

With this scaling it is assured that the relative appearance of events is conserved. In fact, the whole process runs in quick motion, accelerating every individual reaction by the same factor, resulting in a faster approached steady state.

ODEs species move "infinitely fast" and for this reason space can be ignored by the time dilation.

This procedure works fine for ODEs where space does not play a role. In a particle simulation, space is the crucial benefit and also needs to be adapted. Assuming it takes one particle time t to move from position $P = (p_x, p_y, p_z)$ to $Q = (q_x, q_y, q_z)$ in the original system. In the system with reduced timescale $\hat{T}$ it has only $\frac{t}{c_t}$ time to cover the same distance, which is impossible under unchanged conditions, because the maximal movement of particle is limited by its diffusion. The aim is now to adapt the new system, such that the particle can move from P to Q within the grained time.

In a d-dimensional space the mean-square-displacement (MSD) estimates the area that one particle has moved through within a time T undergoing free diffusion D [Mic10]:

$$\mathit{MSD} = 2dTD. \tag{2.11}$$

Taking the square-root of the MSD results in the average distance one particle moved from its initial position after time T.

$$\mathit{RMSD} = \sqrt{\mathit{MSD}} \tag{2.12}$$

Now this distance RMSD will be conserved through the time scaling, whereby $\hat{D}$ denotes the adapted diffusion system:

$$\begin{aligned} \sqrt{2d\hat{T}\hat{D}} &\overset{!}{=} \sqrt{2dTD} \\ \Leftrightarrow \quad \hat{T}\hat{D} &= TD \\ \Leftrightarrow \quad T\hat{D} &= c_t TD \\ \Leftrightarrow \quad \hat{D} &= c_t D \end{aligned} \tag{2.13}$$

It follows that only the diffusion coefficient of every particle has to be adapted with the time scaling factor c_t. Obviously, to make particles move further in the same amount of time an increase of their mobility is required, corresponding to their diffusion-coefficient.

Nevertheless, this is only a theoretical consideration, as the diffusion-coefficient has a limited range of $\approx 0.5 - 30\frac{\mu m^2}{s}$ in real biological systems [WC55]. Exceeding these values results in instabilities due to uncontrolled jump in space under the chosen step-size. This can be fixed by decreasing the step-size, which has a contradicting affect to the attempt to accelerate the simulation, as more steps would be necessary. For that reason the diffusion-coefficient D will be seen as a constant that cannot undergo any scaling.

Diffusion-coefficients cannot be increased freely, similar to the fact that the time-step cannot be decreased. As $\Delta t \propto \frac{1}{D}$ they can be adapted combined.

2.2.3 *Reduce Space Based on the Time Scaling*

Another way to make particles move from P to Q in the scaled time $\hat{T}$ is to decrease the distance between the points, by transforming the reactive volume via a centric scaling, also including the scaling of all particles radius by the same factor. Assuming a point (X, Y, Z) in a 3-dimensional space S that is transformed to a point $(\hat{X}, \hat{Y}, \hat{Z})$ in space $\hat{S}$

$$P = (X, Y, Z) \in S \to \hat{P} = (\hat{X}, \hat{Y}, \hat{Z}) \in \hat{S} \tag{2.14}$$

whereby the transformation is a linear mapping:

$$P \to f(P) = \left(\frac{X}{c_s}, \frac{Y}{c_s}, \frac{Z}{c_s}\right) = \hat{P}. \tag{2.15}$$

This transformation is a uniform scaling of all space axes for a yet unknown factor $c_s > 1$, for simplicity's sake. Assuming a particle that moves from P to Q in space S in time T, factor c_s has to be chosen so that the same particle moves from $f(P)$ to $f(Q)$ in space $\hat{S}$ in time $\hat{T}$. As scalings are length-conserving (see App. A.3 for a proof)

Heterogeneous axis dilations also would be possible, but convert spherical particles into ellipsoids.

the distance $d(P,Q)$ between two points P,Q in the original systems is scaled by factor c_s in the new system $\hat{S}$:

$$d(f(P),f(Q)) = \frac{d(P,Q)}{c_s}. \tag{2.16}$$

This relation allows a comparison of the original RMSD with the one of the grained system($\widehat{\text{RMSD}}$) as they present distances.

$$\begin{aligned} \widehat{RMSD} &= \frac{RMSD}{c_s} \\ \Leftrightarrow \quad \sqrt{2dD\hat{T}} &= \frac{\sqrt{2dDT}}{c_s} \\ \Leftrightarrow \quad \sqrt{\hat{T}} &= \frac{\sqrt{T}}{c_s} \end{aligned} \tag{2.17}$$

Incorporating Equation 2.9 results in the following relation

$$c_s = \sqrt{c_t}. \tag{2.18}$$

Due to numerical restrictions the space dilation should be chosen first.

With Equation 2.18 the space dilation can be determined once the time dilation is chosen, or vice versa. Scaling the time and space axis produces a miniature model system, with the same timely appearance of events, making it possible to simulate large systems for longer times.

2.2.4 *Reduce Diffusion*

This method is used in Section 4.

Events are rather rare compared to Brownian motion, for what reason it consumes more calculation power.

While previously introduced methods focus on the acceleration of the simulation process, now the aim is to speed up the algorithm. Probably the most powerful way of accelerating particle based models is improving the simulation of free diffusion as this usually is the most time consuming process during a particle simulation. If studying self-assembling processes and particles or molecule complexes are far away from each other, it would take multiple time-steps of only free diffusion and no interactions until they are close enough to interact. To overcome this delaying behavior van Zon developed a method of an event driven algorithm [ZW05] that only simulates time-steps where a reaction could take place, which has been described previously in Section 1.2.2. However their algorithm only allows the treatment of point particles with no extension and undergoing chemical reactions, meaning that spatial features and assembling processes cannot be studied with the GFRD software.

Here, components have the same meaning like in graph theory: a structure in which any two particles are connected by a path.

In this thesis, the next step is taken and components are no longer point-lie particles, but objects with a radius, that can form complex molecules, composed of multiple particles. The basic idea of GFRD

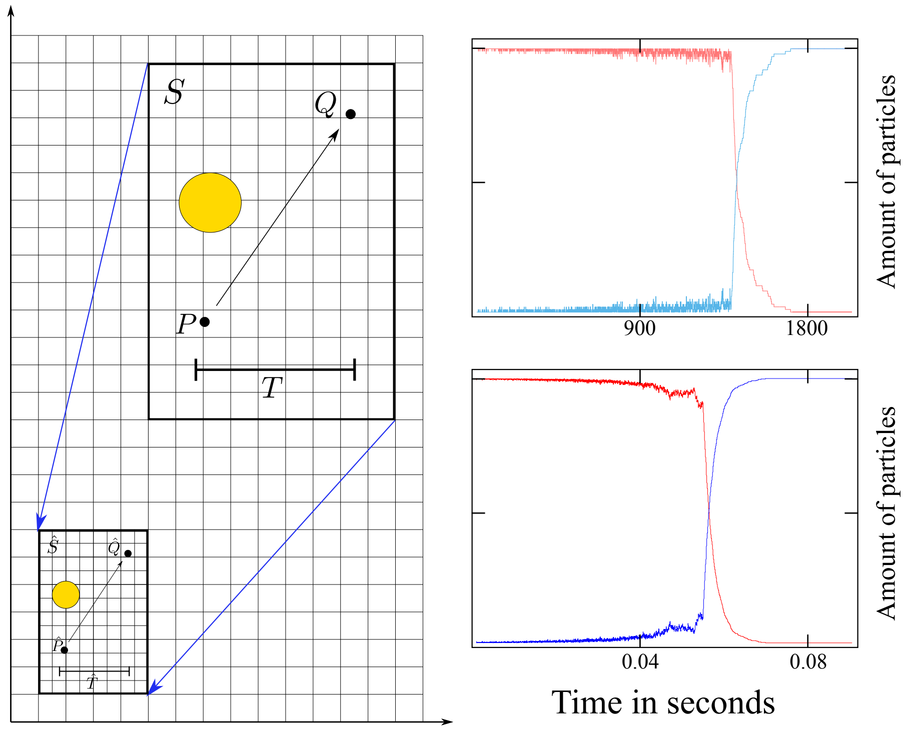


Figure 2.3: **Left:** Schematic illustration of the coupled space-time dilation. The original system S is scaled to system $\hat{S}$, whereby points in the coordinate system and each particles radius are dilated as well. Dilating the time from T to $\hat{T}$ results in a shorter time axis, as seen in the scheme, allowing longer simulations. **Right:** Comparison of the original and scaled system of the Doncic model [DBJB05]. The upper panel shows the full time simulation, which is done using an ODE simulation (performed in Copasi with a hybrid algorithm, threating species partially stochastic), as the ReaDDy particle simulation would run, extrapolated, for nine years. Bottom panel shows the results of the reduced time-axis, simulated with ReaDDy that took only ~ 6 hours. It is seen clearly that the qualitative behavior is nearly the same, while the simulated time is scaled by the factor $c_t = 22,500$ and space by $c_s = 150$, and the gain in calculation time is enormous.

is to calculate the next time that an event can take place, based on the diffusion coefficient of every single particle, similar to Gillespie's algorithm [Gil77]. Difficulties arise, as it is not easy to estimate the time of the next encounter between two components. An intuitive approach would be to determine the convex hull of a component and approximate it with a sphere and estimate its diffusion coefficient, based on the particles it is composed of threating it as a rigid object, assuming the component would not change its shape during the free diffusion that is skipped.

Components represent pseudo-particles that do not change their inner structure during a space leap.

Consideration of the shape during the diffusion ends up with two major problems: Firstly, it is unclear which shape the component will end up with, and secondly it may even happen that the shape becomes bigger than the estimated sphere it is contained in. The second

Components may increase their size, e.g. an unwrapping coiled coil, which would break the simulation.

issue could be overcome by estimating the maximum size of the component and choosing the size of the sphere based on this value. This assures that every event allows a maximum of one interaction, as it may happen that the maximum extension will not be reached, so no reaction will take place at all.

The uncertainty of the shape of a component can be overcome by several methods:

(I) run a micro simulation, containing only the component

(II) sample over multiple micro simulations and choose the shape based on the derived probability distribution

(III) break all bonds of the component, choose the position of every individual particle based on its free diffusion and assemble all bonds.

Attempts (I) and (II) are incongruous as they demand the amount of calculation power that would be gained by coarse-graining the diffusion; bottom line is that nothing is achieved in terms of computational savings. Method (III) is the most promising one but comes with the huge disadvantage that particles could move too far from each other and could result in instabilities while snapping back together, and therefore none of the approaches (I) - (III) is used in further considerations.

Assuming a relaxed simulation whereby components have no internal force and thus are in the most probable shape, the shape of a component won't vary over the diffusive coarse-graining, solving both of the major issues.

2.3 INTRODUCING THE NOVEL DICOSAD SOFTWARE

Within the scope of this thesis DiCoSAD (**Di**spersing **Co**mponent **S**elf-**A**ssembly **D**ynamics) was developed, which is a particle based simulator that combines rule-based reaction networks with an event driven algorithms like GFRD, namely that large jumps of molecule-complexes in space and time are possible [Hen18a]. This feature allows the study of self-assembling processes that can take several minutes, hours or technically any length of time. Basically it uses Brownian motion as the power engine and spatial rule-based reaction rules if particles intersect (cf. SRSim, [Gru+10]). It coarse-grains the diffusion as introduced in GFRD if the next event is not during the next time-step [ZW05]. The software package DiCoSAD was developed in Python 2.xx and is freely available.

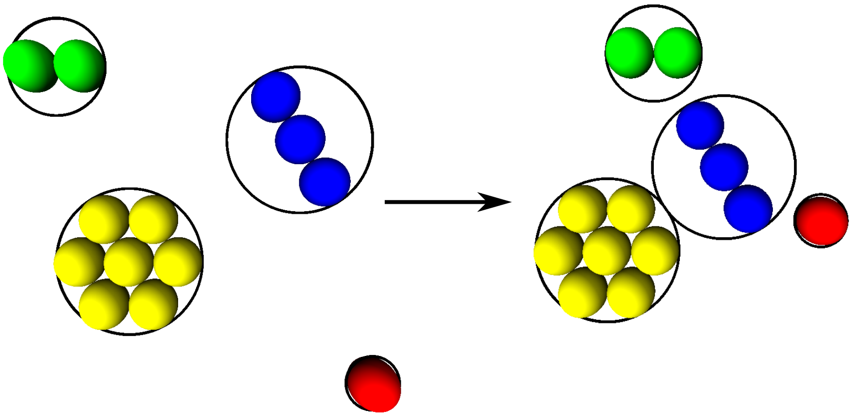


Figure 2.4: Illustrating the coarse graining of the diffusion. Shown are four different colored components undergoing one event time-step of the simulation indicated by the arrow. The next reaction to happen is between the blue and the yellow components as their interaction radii is closest, meaning that a reaction between them could happen in the next time-period. Note, that the internal structure of all components is unchanged as discussed in Section 2.2.4.

2.3.1 *Concept*

DiCoSAD mostly follows the basic concept of a particle based simulator, but has a significantly different approach to the time treatment. The reactor is a cubical volume with reflective boundaries and particles are described as points in euclidean space with a diffusion coefficient, mass and extension.

Diffusion and mass correlate, and one is obtained from the other, but they are both required for certain calculations.

Conventional particle approaches like ReaDDy or Smoldyn, discretize time into equidistant steps Δt and at every time-step every particle's position $\vec{x}_i(t)$ is updated according to the following equation (cf. Eq. 1.24):

$$\vec{x}(t+\Delta t) = \vec{x}(t) + F_B(\vec{x}(t)) + F_S(\vec{x}(t)) \tag{2.19}$$

whereby F_B denotes the basic motion, usually a normally distributed random displacement in all dimensions, and F_S the system motion, like wall potentials, harmonic potential between particles or other force fields.

Every individual particle is a node in the so called binding graph that initially is unconnected. Bonds are represented as edges between the regarding particle nodes in the graph. **Components** are connected subgraphs of the binding graph. Initially every single particle is a component, as no bonds are formed yet.

G = binding graph, V = individual particles, E = bonds.

Components can only interact if they are close to each other, making random displacements unnecessary if their relative distance does not change. DiCoSAD calculates, based on the MSD and the distance between two components c_i and c_j, the time ΔT_{ij} it takes for these two components to intersect (cf. App. 2.3.4). The minimum of these timings results in the time ΔT of the next reaction that possibly could happen which determines the classification of the basic motion F_B:

(1) $0 < \Delta T < T_t$ **Brownian Motion:** no time-steps should be skipped, Brownian dynamics is applied (cf. Sec. 2.3.3);

(2) $T_t < \Delta T$ **Disperse:** the minimal time ΔT is greater than a chosen threshold T_t, all components are dispersed according to the time-step of the next possible interaction (cf. Sec. 2.3.4).

The threshold T_t is set initially and can be chosen freely, whereby a high value would not benefit much from the diffusional coarse graining. For spatial arrangements it is necessary to have an area of Brownian dynamics, so complexes can assemble. Without this threshold it would be unlikely for bonds to form, as they have to hit the right spatial conditions immediately. Following sections will elaborate the estimation of the next event and discuss different basic motions in more detail.

2.3.2 *Calculate Time Until Next Possible Interaction*

A list of particles transits to a list of components that are indexed with i.

For simplicity's sake, components c_i furthermore are treated as spheres with radius r_i and a diffusion coefficient D_i, resulting from the combined mass of the particles it consist of. Their center $\overrightarrow{C_i}$ is the average position of particles in the component, while the radius r_i is estimated from the maximal inner distance of particles in the regarding component (trivial if the component consist of a single particle):

$$\overrightarrow{C_i} = \frac{1}{|c_i|} \sum_{p \text{ in } c_i} \overrightarrow{x_p} \tag{2.20}$$

$$r_i = \max_{p,q \in c_i} \left(\|\overrightarrow{x_p} - \overrightarrow{x_q}\| + r_p + r_q \right) \tag{2.21}$$

$|c_i|$ denotes the amount of particles in component c_i, $\overrightarrow{x_p}$ the position of particle p, and r_p its radius; r_i is the radius of the component c_i, whereby r_i and r_p coincide in case of a one-particle component. This simplification is valid, as the estimated radius r_i of every component is larger than its real extension, assuring that components cannot intersect directly after their dispersion.

The diffusional radius $r(D_i, \Delta T)$ is the maximum distance that a component c_i can have moved away from its original central position $C_i(0)$ after time ΔT undergoing diffusion D_i. For that purpose, the pairwise distance between all components centers is calculated:

$$d(c_i, c_j) := \min_{\forall c_i, c_j, c_i \neq c_j} \|\overrightarrow{C_i} - \overrightarrow{C_j}\|_2. \tag{2.22}$$

whereby c_i and c_j denote two different components and C_i and C_j , respectively, their positions. Then, the minimal time is estimated which it takes for two components to intersect, based on their initial distance $d(c_i, c_j)$. The MSD (introduced in Section 2.2.2) is a measurement of how far one particle may have moved from its origin after a time T undergoing free diffusion D:

$$MSD = \sqrt{6DT} \tag{2.23}$$

In fact, after the spherical simplification, components are larger pseudo-particles, making them applicable for the *MSD*. Combining the MSD for two components with radii r_i and r_j and their diffusion D_i and D_j, respectively, results in the following:

$$d(c_i, c_j) = \sqrt{6D_iT_{ij}} + \sqrt{6D_jT_{ij}} + r_i + r_j \tag{2.24}$$

Whereby T_{ij} represent the minimal time it takes for component c_i

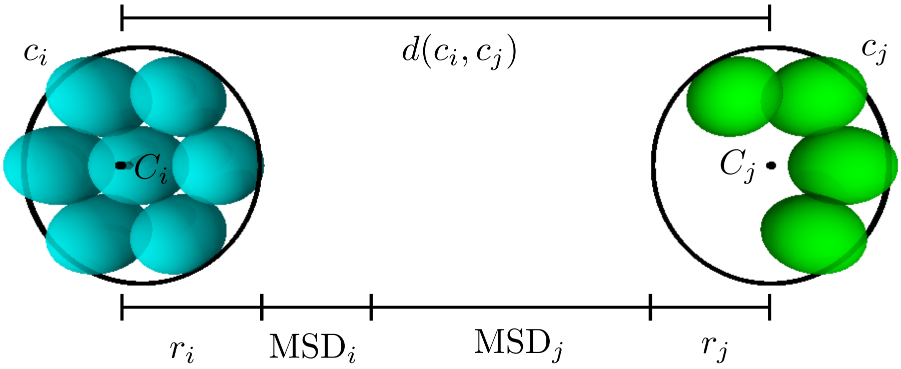


Figure 2.5: Shown are two components c_i and c_j with their radii r_i and r_j and their center points C_i and C_j, respectively. It is seen that the distance $d(c_i, c_j)$ between them can be split up into a sum of the radii and the *MSD* of the components. This scheme is a visualization of Equation 2.24.

and c_j to intersect. Compare Figure 2.5 for a visualization of Equation 2.24. With all the values given, Equation 2.24 can be solved for T_{ij}:

$$T_{ij} = \frac{1}{6}\left(\frac{d(c_i, c_j) - r_i - r_j}{\sqrt{D_i} + \sqrt{D_j}}\right)^2 . \tag{2.25}$$

Apparently, the time ΔT of the next possible event is the minimal T_{ij}.

$$\Delta T = \min_{\forall c_i, c_j, c_i \neq c_j} T_{ij} \tag{2.26}$$

2.3.3 *Brownian Motion*

Close components are those, which estimated reaction is sooner than T_t.

If two components are close to each other and not connected yet, it is desirable that they form a bond naturally, meaning they intersect after a Brownian motion step and not as result of a dispersion. While this is the case, time-steps have a fixed size Δt and the basic motion F_B of all particles follows equation:

$$F_B(\vec{x}_i) = \sqrt{2\Delta t D_i}\vec{\mathcal{N}}(0, 1) \tag{2.27}$$

Every particle position $\vec{x}_i$ is displaced by a normally distributed random step, based on its diffusion coefficient D_i.

2.3.4 *Dispersion*

If the time of the next event ΔT exceed the predefined threshold T_t all components are dispersed following a normal distribution as well:

$$F_B(\vec{x}_i) = \sqrt{2\Delta T D_i}\vec{\mathcal{N}}(0, 1). \tag{2.28}$$

which is valid, as components are spherical pseudo particles. Every component is treated as one rigid object, resulting in the same displacement of all particles in this component, based on the component's diffusion coefficient D_i, meaning that only one displacement is necessary for every component and not every individual particle.

The algorithm ensures that until $t + \Delta T$ no reaction can happen, meaning that the diffusional radii $r(D_i, t)$ do not intersect. Therefore, components can be sampled freely within their diffusional radii without influencing or overlapping any other component.

2.3.5 *Rule-System*

All individual particles are nodes in the binding graph that is initially unconnected. Reactions are formulated as rules that modify this graph, which is done by either changing the state of a node, connecting two nodes or breaking the bond between two nodes. In DiCoSAD only Uni- and Bimolecular rules are allowed that follow the BNGL description [Bli+04], whereby higher order reactions can be converted to those of order two. The general functioning of a rule-based system is described in Section 1.2.2 and more exhaustively in [Gru+10; Ibr+13; Hen+18b].

Unimolecular Reactions

Unimolecular reactions are of the following form:

Currently, unimolecular reactions are not implemented in DiCoSAD , yet.

$$E \xrightarrow{k_{off}} P_1 + \cdots + P_n \tag{2.29}$$

where one educt E decays into products $P_1, \dots, P_n$ with a reaction rate k_{off}. Following Gillespie's approach [Gil77] a set S_{Uni} is created, containing every unimolecular reactions time Δt^i_{Uni}, when it is going to happen next, based on the current concentration $[E]$, the reaction rate k^i_{off} and a uniform random variable $\mathcal{U}(0,1)$:

The index i refers to the ith unimolecular reaction.

$$\Delta t^i_{Uni} = \log\left(\mathcal{U}(0,1)^{-1}\right) \times \left([E]k^i_{off}\right)^{-1}. \tag{2.30}$$

If a reaction is chosen to happen, the individual component is chosen randomly, and the rule either modifies the node corresponding to the species or splits it into multiple components, by removing the connection in the graph. After the update of the binding graph, a new time Δt^i_{Uni} is drawn for the regarding reaction and added to S_{Uni}.

Bimolecular Reactions

Bimolecular reactions are of the following form:

$$E_1 + E_2 \xrightarrow{k_{on}} P_1 + \cdots + P_n \tag{2.31}$$

where two reactants E_1 and E_2 react to form the products $P_1, \dots, P_n$ with a reaction rate k_{on}. These reactions do not need the calculation of the time in which they happen, because they only take place when two particles intersect each other, which is checked after the application of the basic motion. If a rule between two components is possible, they react based on a probability, proportional to k_{on} (cf. Sec. 2.1.3), resulting in the formation of a bond, by connecting the two particles

Rules are formulated in the BNGL language, that specifies the exact particles of a component that interact.

in the graph by an edge, or changing the state of the nodes of either or both of the reactants in a catalytic manner ($A + B \to A + B^p$ describes a reaction where A phosphorylates B).

2.3.6 *Algorithm*

Previous explanations taken together result in the basic DiCoSAD algorithm:

(1) **Input:** Set of species, their initial amount and spatial definition; set of reaction rules (S_{Uni} and (S_{Bi}) between the defined species and their reaction rates k^i; definition of the reacting volume; threshold T_t; fixed stepsize Δt; maximum time-steps T_{steps} or maximum time T_{end}; system force fields F_S.

(2) **Initialize:** Place all particles randomly in the reaction volume; create unconnected binding graph; $t \leftarrow 0$; $t_{steps} \leftarrow 0$; create set of times for unimolecular reactions: $\forall_{R \in S_{Uni}} \Delta t_R \leftarrow \log(\mathcal{U}(0,1)^{-1}) \times ([R_E]k^R)^{-1}$.

(3) **Iterate:** Repeat while $t < T_{end}$ and $t_{steps} < T_{steps}$:

(3.1) **Decide Type of Next Event:**

(3.1.1) **Next Bimolecular Reaction:** Calculate the next possible time of a bimolecular reaction ΔT_{Bi}, following the procedure in Section 2.3.2.

(3.1.2) **Next Unimolecular Reaction:** Estimate the next possible time of a unimolecular reaction $\Delta T_{Uni} \leftarrow \min_{R \in S_{Uni}} \vec{t_R}$.

(3.1.3) **Time of Next Reaction:** Next possible reaction time $\Delta T \leftarrow \min(\Delta T_{Bi}, \Delta T_{Uni})$.

(3.2) **Apply Basic Motion** F_B

(3.2.1) $\Delta T \leqslant T_t$**:** Apply Brownian Dynamics on all particles $F_B \leftarrow \sqrt{2\Delta t D_i}\vec{\mathcal{N}}(0,1)$.

(3.2.2) $T_t < \Delta T$**:** Disperse every component $F_B \leftarrow \sqrt{2\Delta T D_i}\vec{\mathcal{N}}(0,1)$.

(3.3) **Apply System Motion** F_S Update all particles position following the user defined potentials.

(3.4) **Apply the appropriate Rule:**

(3.4.1) **Unimolecular Reaction** In case a unimolecular time was taken in step (3.1) update the graph according to the rule.

(3.4.2) **Apply Bimolecular Rules** If two particles intersect apply rule if possible and in case update the binding graph.

(3.5) **Update:** $t \leftarrow t + \Delta t$ if $\Delta T < T_t$ else $t + \Delta T$; $t_{steps} \leftarrow t_{steps} + 1$.

(4) **Finish:** If $t > T_{end}$ or $t_{steps} > T_{steps}$ finish and clean up.

2.3.7 *Application: Polymerization*

After introducing DiCoSAD , this section demonstrates its general speed up and application to a biological system.

Firstly, as a simple model to evaluate the gain of DiCoSAD in terms of calculation speed, a simple polymerization process was chosen. Only one kind of particle A exists that can bind two particles A at opposing sites, forming a chain. If the reaction volume is diluted it takes a long time for large polymers to form, because most of the time particles are freely diffusing without any reactions. DiCoSAD tackles exactly this problem and demonstrates in Figure 2.7 its advantage. While the calculation time on a single CPU is $2-3$ longer than a conventional method, the system time that is simulated is $\sim 1,000$ times larger. As measurement for the complexity of this system the average length of a chain has been chosen, being initially 0 and N in steady state, whereby N is the amount of reacting particles. To show that the results of both methods are comparable the distribution of polymers after a fixed simulated time is shown in Figure 2.6.

The length of polymers determines the complexity of the system, whereby the formation of rings was neglected.

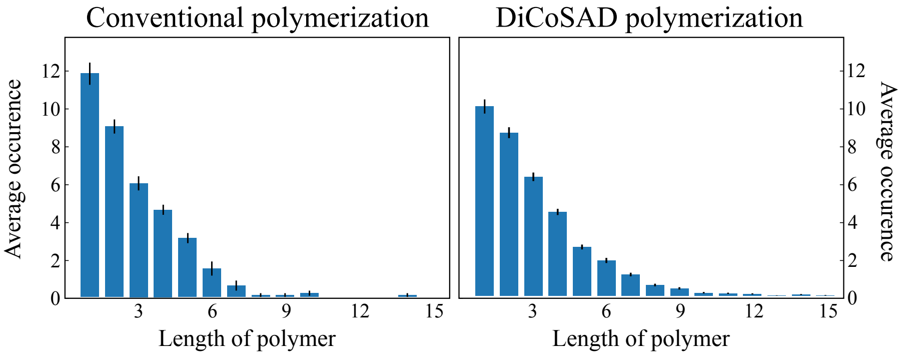


Figure 2.6: Qualitative comparison of DiCoSAD with a conventional particle based approach. Shown are the statistics (mean and standard error) of the polymer distribution of 50 independent polymerization simulations, without coarse-graining of the diffusion (**Left**) and with allowed time-jumps (**Right**). All runs have a simulated time of 0.005s to compare the qualitative behavior of DiCoSAD . It is seen that the general distribution is similar and follows roughly an exponential behavior.

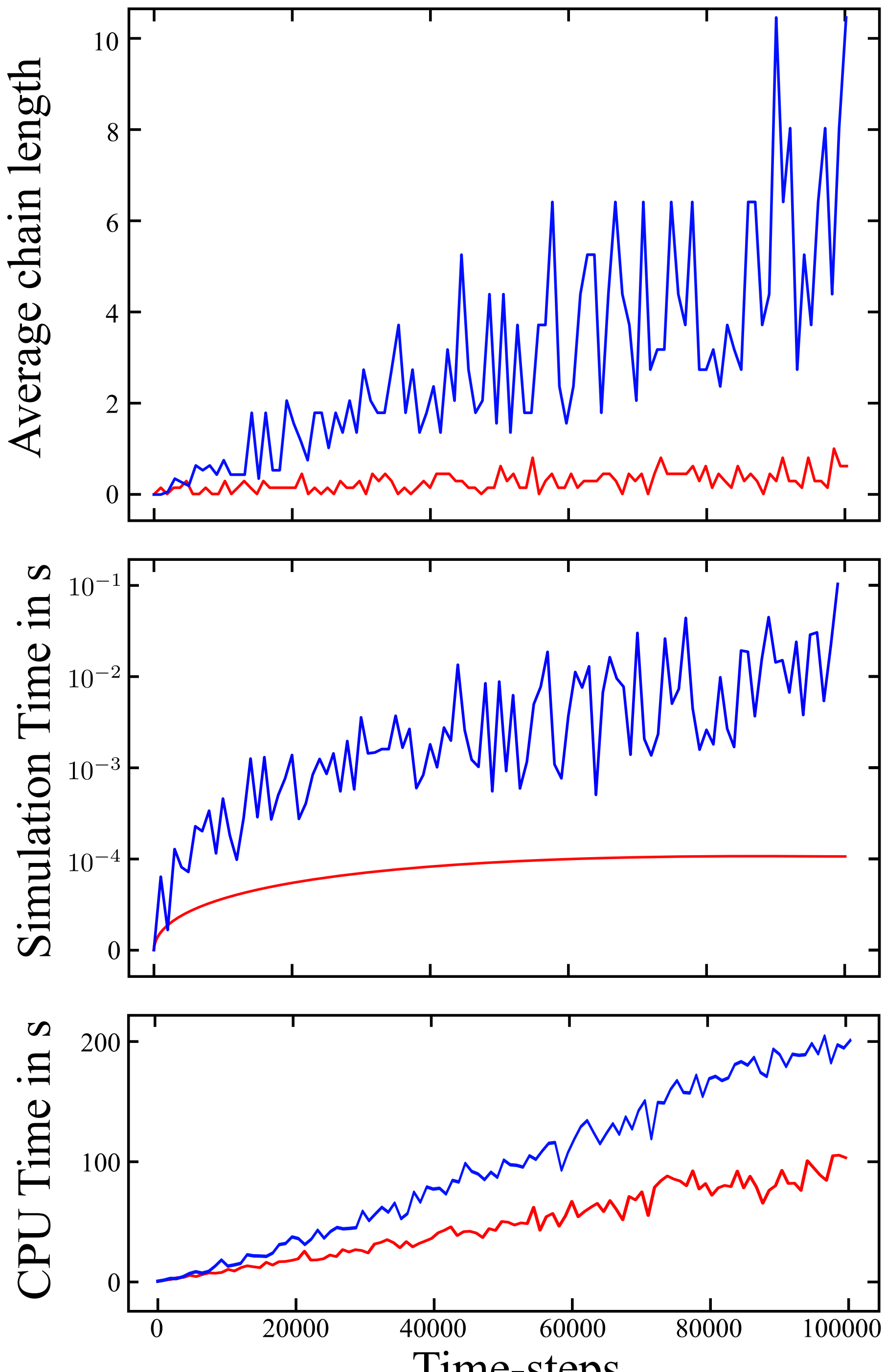


Figure 2.7: Quantitative comparison of DiCoSAD with a conventional particle based approach. Shown are 100 independent simulations of 30 particles that ran up to 100, 000 time-steps (cf. algorithm DiCoSAD ; x-axis), without coarse-graining of the diffusion (red curves) and with allowed time-jumps (blue curves). Compared are the average chain length of forming polymers (initially 1 and steadily increasing), the time it took a single core CPU to compute, and the simulated system time (logarithmic time-scale). Its shown clearly that, while the CPU time of DiCoSAD is increased, the simulated time outweighs the conventional one several magnitudes on a logarithmic scale. While the red curve increases linear (fixed time-steps), the blue one is stochastic, as the time-step depends on the next event (cf. Sec. 2.3).

2.4 CONCLUSION

With the methods presented in this section (cf. Sec. 2.1) it is possible to convert an ODE model to a particle based model. Starting with an ODE model, consisting of a finite list of species and their interactions, particles are created that are assumed to be spherical, representing these species. As the particle model will be simulated in space further properties from literature are required, like mass and diffusion coefficient. The diagram 2.1 shows the pathway to achieve a particle based model out of an ODE model with the here developed methods.

Furthermore, Section 2.2 provides automatized methods to simulate the generated model in a feasible amount of time, which is done by choosing either the space (c_s) or the time scaling (c_t) freely. Note that the coupled space scaling c_s is not entirely free as too small particles usually undergo too strong forces and break the reactor.

As can be seen in the diagram there are three coarse-graining methods applied. Two of these are coupled (space and time), whilst the other one is independent (amount of particles). Those methods allow the simulation of a particle based model, which is quantitative altered, but still delivers comparable qualitative results. In fact, the following two chapters demonstrate the functioning of these methods, as they are used to model and simulate two concrete biological systems.

Furthermore, Section 2.3 introduced the novel simulation software DiCoSAD that combines techniques from SRSim and GFRD to create a rule-based event algorithm, able to simulate large time-scales and study self assembling processes. Figure 2.6 compares DiCoSAD with a conventional approach in terms of correctness. Simulated is a simple polymerization process and it is shown that the distribution of polymers after a fixed simulated time is identical. Figure 2.7 then demonstrates the computational advantage of the software as it requires a fractal of the time-steps, of a conventional approach, to reach a certain state.

Table 2.1: **Summarizing the transition from ODE to particle model with the methods introduced in the section.**

			ODE model				
		↓	↓		↓		
Literature		Concentration	Rates		Time		Space
↓	↓	↓	↓		↓		↓
	Mass	Particle Number	Updated Rates	←	Reduced Time	↔	Reduced Space
	↓	↓	↓		↓		↓
	Radius	Pseudo Particles					
	↓	↓					
Diffusion Coefficient							
↓							
			Particle model				

3 COARSE GRAINING MODELS OF THE SPINDLE ASSEMBLY CHECKPOINT

The general switching mechanism in biology as well as the spindle assembly checkpoint are introduced. Based on previously proposed SAC models, here an extended and biochemically reliable spatial SAC model is presented. Using methods from the previous section this model is coarse-grained on multiple levels, simulated and analyzed. The essential switching behavior is analyzed analytically and the crucial rate is found to coincide with the one measured in wet-lab. Lastly, an exact stochastic model is derived, demonstrating that the SAC is in fact a global signal. *This chapter is based on work previously published in [HDI17; Hen+18a].*

3.1 SWITCHES AND THE SPINDLE ASSEMBLY CHECKPOINT

3.1.1 *Introduction*

The cell cycle is a crucial event for all living beings. Without this chain of events, leading to new cells out of old ones, life would not exist. As the life span of most cells is limited it is necessary to reproduce to stay alive.

A common cell cycle consists of three phases: interphase, mitosis and a resting phase. Interphase and mitosis can be subdivided into more specific phases. An overview over the whole process is provided in Figure 3.1. Preventing cells from failing their cycle in either of the stage transitions is achieved by multiple checkpoints that only allow the cell to advance to the next phase if certain requirements are fulfilled [Mur94]. Most notable checkpoints are the G1, G2 and mitotic checkpoint, also called the spindle assembly checkpoint (SAC, [RM96]).

Mitosis replicates the cell nucleus, which is governed by the SAC.

Correct segregation of the DNA is the fundamental process during mitosis that relies on amphitelic attachment between chromosomes, principally through kinetochores and spindle microtubules (cf. Fig. 3.6). Mistakes, meaning an incorrect multiplication of the genome, lead to different sets of cells. These sets are phenomenologically identical but have different information stored in their nucleus and thus

Mistakes are either a wrong alignment, a missing alignment or a double alignment.

may lead to many human health problems, most notably aneuploidy and cancer [KWC04; Mar+15]. The most sensitive part of the mitosis is the transition from meta- to anaphase where the sister-chromatids are split into the two daughter cells. The SAC guards this transition and delays the onset to anaphase until all chromosomes are attached correctly, meaning that every chromosome's two kinetochores are connected to opposing centrosomes via microtubuli (cf. Fig. 3.1 Right panel phase IV).

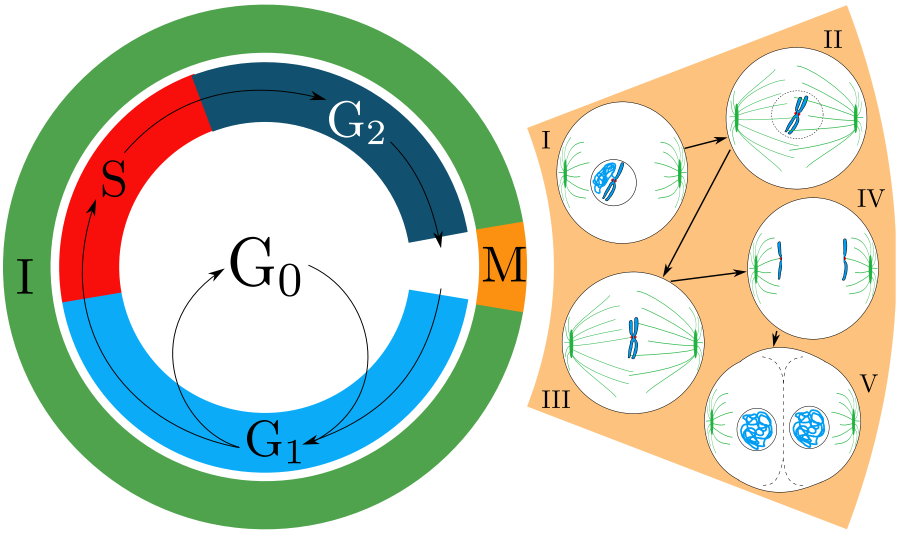


Figure 3.1: **Left:** Schematic illustration of the cell cycle. Basically the mitosis, replicating the nucleus, and the interphase, which is divided into more subgroups (S, G_0, G_1 and G_2). **Right:** Schematic illustration of the mitosis. Shown are the five phases, namely prophase (I), prometaphase (II), metaphase (III), anaphase (IV) and telophase (V). The spindle assembly checkpoint guards the transition from meta- to anaphase.

All eukaryotes have a spindle assembly checkpoint mechanism.

The SAC is an evolutionally conserved mechanism, exclusively sensitive to the states of kinetochores that are already attached to microtubules and their drag towards opposing centrosomes. Transiting from meta- to anaphase requires two core proteins, namely Separase and Cdk1. Free Separase cleaves cohesin, which physically connects the two sister-chromatids of every chromosome [GHN03]; Cdk1 initiates the mitotic exit and signals the proceeding to anaphase. As this transition only shall take place once all kinetochores are attached properly, meaning that a pair of sister-chromatids is connected to opposing centrosomes, Separase is inhibited by Securin and Cdk1 by CyclinB. The ubiquitin ligase anaphase-promoting complex/cyclosome (APC/C) recruits Cdc20 and ubiquitinates Securin and CyclinB [MS07]. During metaphase, the APC/C is inhibited by the mitotic-checkpoint-complex MCC, which binds APC/C directly to hinder Cdc20 from activating it (the forming complex is the APC/C:MCC). If a single kinetochore stays unattached, whether naturally or artifi-

MCC is a protein complex consisting of BubR1:Bub3, Cdc20 and Mad2, inhibiting the APC/C to prevent premature anaphase.

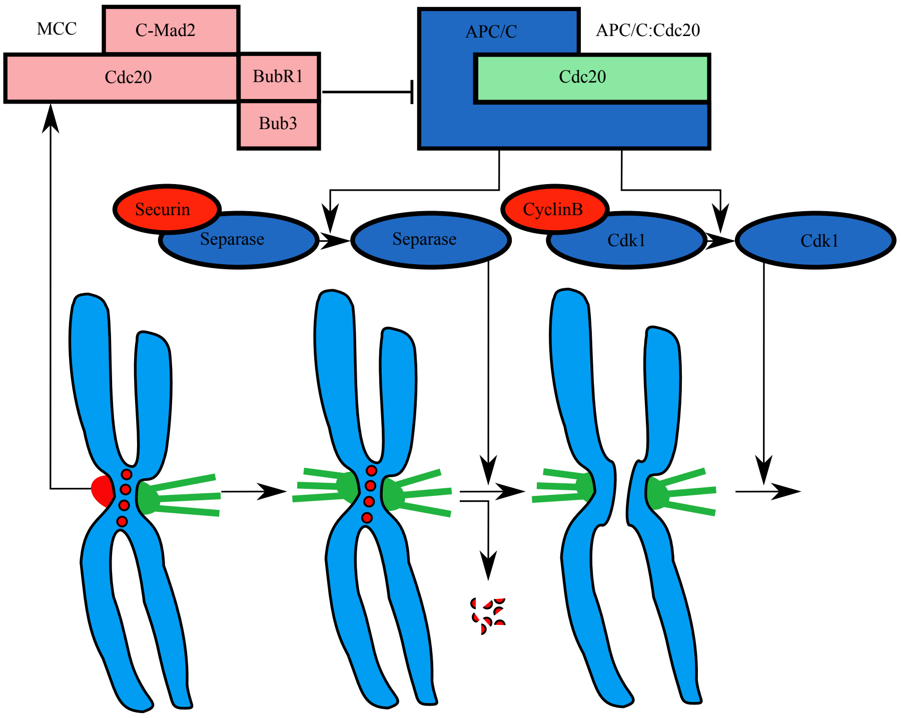


Figure 3.2: Depicted is the general functioning of the spindle assembly checkpoint. Unattached kinetochores (red part of chromosome) catalyze the MCC, which inhibits the APC/C. Once all kinetochores are properly attached, MCC decays, APC/C gets activated, and Separase and Cdk1 get freed, initiating anaphase.

cially, the SAC stays active and can block the function of the APC/C for several hours [Mus15].

After a proper attachment of the final spindle the MCC decays quickly, APC/C is activated and Separase and Cdk1 freed. Reddy et al. demonstrated that UbcH10 targets and ubiquitinates Cdc20 located in an APC/C:MCC complex [Red+07], decreasing its affinity to Mad2, and resulting in the repulsion of Mad2 from the MCC. The remaining complex BCC still inhibits the APC/C but is less stable than the MCC and dissociates faster (cf. App. B.1). Dynein, located at attached kinetochores, releases p31^{comet} [Hag+11; Wes+11], which lowers the concentration threshold necessary for UbcH10 to become active [Red+07]. In this way, p31^{comet} and UbcH10 collaborate to silence the SAC and allow a fast transition to anaphase, once all kinetochores are proper attached.

3.1.2 *Computational SAC Models*

While the principal of the SAC is mostly investigated and well understood, the exact mechanism are still lacking some explanation, e.g. how a single kinetochore generates a global signal and the sensitiveness of the switch and what the switch relies rely on. As these kind of underlying mechanics are difficult to investigate in a wet-lab, in the past decades a number of computational biologists have studied the SAC *in-silico* and tried to explain its behavior by modeling. Some of the many models which mimic the reported behavior of the SAC have emerged and will be introduced briefly below.

Doncic, Sear and colleagues analyzed abstact spatial models of potential checkpoint mechanisms with a focus on yeast and animals [DBJB05; SH06]. Mistry et al. provided a framework accounting for the correction mechanism for improper chromosome attachment [Mis+08]. Lohel et al. considered models that take into account species localization and realistic kinetochore-binding kinetics, focusing on diffusion effects [Loh+09]. Models by Ibrahim and colleagues provide detailed descriptions of human SAC activation but lack a realistic explanation for the switch with respect to the reaction network [Ibr+08a; Ibr+08b]. They use a manually-activated switch that enables or disables certain reactions according to one of the two phases (active and silenced SAC).

None of the models mentioned so far contains a reliable mechanism for silencing the checkpoint, and all lack realistic spatial properties. Furthermore, to capture the rheostat behavior reported by [Col+13; DG13], the respective SAC output signals, Securin and CyclinB, need to be included.

Chen et al. provided a compartmentalized model that gives a spatial explanation for the silencing of the SAC [CL14]. They account for the rheostat switch through active protein transport and their model gives a reliable explanation how SAC proteins are transported from the kinetochore region to the centrosomes. However, they emphasize that they did not focus on the underlying biochemical pathways for SAC activation.

No model until now includes the drag of microtubuli that is necessary to check for a proper alignment. Most model use a boolean to toggle between attached and unattached.

In the scope of this thesis, the first detailed mathematical model that considers all 92 human kinetochores and all major proteins involved in SAC activation and silencing is proposed. Deterministic and spatially-stochastic simulations are performed to find that certain spatial properties do not play a significant role. Thus this model developed here can be regarded as a completion of the model proposed by Chen et al., in that a realistic biochemical pathway is provided, while active transport is not taken into account (opposing the approach of Chen et. al.). Furthermore, it is shown that the model is consistent with the recently suggested rheostat switch behavior, measured by Securin or CyclinB concentration.

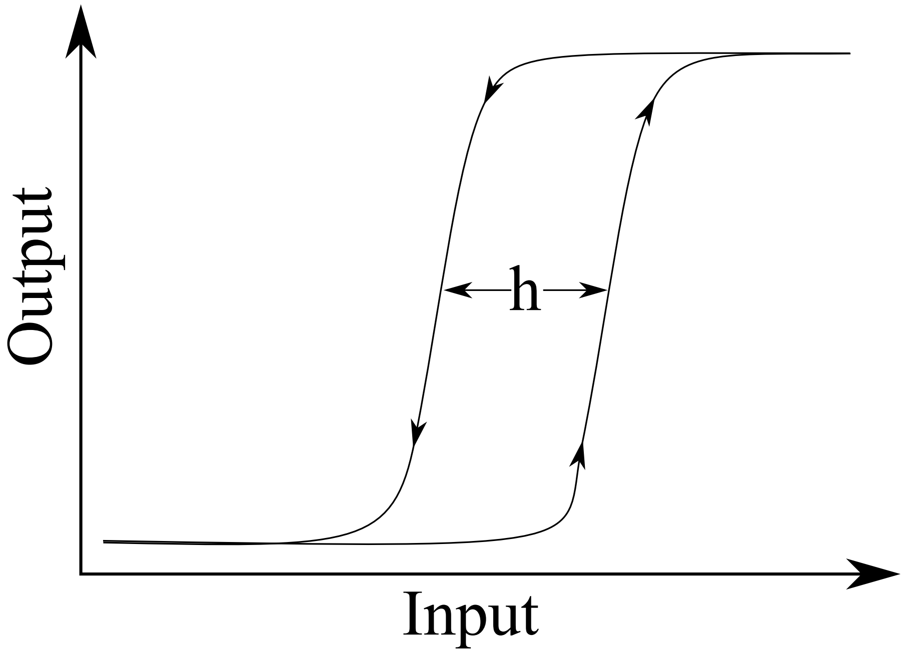


Figure 3.3: Shown is the dependency of a system of an input. The two stable steady states in this example could be ON and OFF, between which can be switched if a certain input is reached. Note, the system first needs to overcome a certain threshold to switch back, which is depicted by the hysteresis between the two lines. If the hysteresis equals 0 the system is reversible; an infinite one corresponds to an irreversible switch.

3.1.3 *Switches in Biological Systems*

A system with multiple reachable and stable steady-states has the ability to switch between them, dependent on some input, whereas switches generally can be reversible or irreversible. The reversibility of a switch is seen on the hysteresis, which is explained graphically in Figure 3.3. In a Boolean network switches are realized as a toggle between 0 and 1, because all nodes can only take two states and change them regarding to their inputs immediately. In real biological systems it is more complicated to realize switches [Tys+08], and in the following, techniques will be discussed to build models that yield a switching behavior similar to the one of the SAC.

Input could be the concentration of other species, or the appearance of an event i.e.

Bistability

Bi- or multistability describes a property of a dynamic system to have multiple stable equilibrium points. In mathematical terms, an equilibrium point is a local maximum in the force field, whereby minima correspond to the stable points. One of the most prominent and easiest systems that yields a bistability is the following one dimensional system:

$$\frac{dX}{dt} = X(1-X)(1+X). \tag{3.1}$$

It can be easily seen that the system has three steady states, namely -1, 0 and 1, whereby 0 is unstable (local maximum) and $1, -1$ are stable (local minima). Bistability always implies hysteresis (cf. Fig. 3.3), determining which state to go in based on the history of the input values. Creating bistability in a complex system requires feedback-loops as well as an ultrasensitive response [Nov+07].

Detecting a bistability in an existing system requires a bifurcation analysis, which basically simulates the system with every possible input history to every direction. The eigenvalues of the Jacobian matrix [HKW81] provide stability properties of the regarding system, which is used to find local bifurcation points, but usually is insufficient to detect global bifurcations. For that reason bifurcation points often are estimated numerically, with the disadvantage that rounding errors can lead to instabilities and makes the case for larger systems, where rounding errors are inevitably not reliable. The key is to combine both approaches, because pre-estimated analytical properties simplify the bifurcation analysis.

Every direction means that the parameter under study is increased and decreased from every starting point.

A successful bifurcation analysis detects the switching mechanism in the system, that generally can be classified into rheostat or toggle switches.

Rheostat Switches

The simple approach of creating a switch would be the activation of a gen B that firstly requires the activation of another gen A:

$$A \xrightarrow{E} A^* \tag{3.2}$$

$$B \xrightarrow{A^*} B^* \tag{3.3}$$

Enzyme E activates A once it reached a certain threshold, and then activated A is able to activate B. Input is the concentration of enzyme E, output the one of gen B. This simple activation is insufficient as a switch because the response is rather slow and the idea behind a switch is to have an immediate response.

To enhance or slow down the response it is possible to construct feedback loops [Alo06]. These are constructs that use the output to influence the input, resulting in a chain reaction. For the example given above a positive feedback loop would be the activation of A via B; a negative feedback would be the inhibition of A via B. Negative feedback loops can yield an oscillating behavior, as the species are rebalanced. Positive feedback loops amplify the output and thus switch between the states faster. Nevertheless, more desired is an all-or-nothing response, giving a small change of input, generating a huge effect on the output.

Toggle Switches

Achieving an ultrasensitive response is modeled with the Hill equation [Hil10]:

$$\frac{dB}{dt} = k_{on} \frac{A^n}{K_A^n + A^n} \tag{3.4}$$

Mass action kinetics assume that the rate of a reaction is proportional to the amount of its educts. Hill-kinetics on the other hand take cooperativity into account. K_A refers to half the concentration of the binding sites and thus determines the threshold of the reaction. The Hill coefficient n describes the strength of cooperativity and thus defines the steepness of the response curve. A theoretical value of ∞ for n would result in a Boolean response and toggle immediately.

Already bound ligands A enhance the binding of other ligands, which is similar to a positive feedback loop.

3.1.4 *Switching the SAC*

Switching the SAC from active to its inactivated form needs to be a bistable switch that is temporarily reversible. Once all kinetochores are attached the APC/C has to be activated rapidly to start to degrade Securin and CyclinB. This switching has to be reversible for some time, so that somehow disconnected kinetochores can re-activate the SAC. Nevertheless, approximately five minutes after the last attachment the process becomes irreversible [DG13]. This behavior suggests that the SAC is a bistable system (SAC active and inactive), whereby the input signal correspond to the amount of unattached kinetochores. The bistability of the SAC suggests that a feedback-loop is included in the SAC network, which will be investigated in further chapters.

This switch from inactive to active APC/C was believed to be an all-or-nothing response, meaning that once all kinetochores are attached APC/C starts degrading Securin and CyclinB. However, Collin et al. have recently shown that this is not the case [Col+13]. They postulate that the APC/C inhibitor MCC is active at different levels, depending on the number of unattached kinetochores, resulting in an active APC/C during metaphase. This hypothesis was underpinned by the concentration curve of CyclinB, which decreased continuously depending on the MCC level. Dick et al. make the same claim based on Securin experiments [DG13].

Bistability necessarily requires a feedback loop (cf. Sec. 3.1.3), which are not easy to detect in wet-labs, and for this reason the mechanism, leading to the silencing of the SAC is not well studied qualitatively in literature. Incorporating the loop into the full model can be achieved by adding a single reaction, but finding the right one is challenging. Based on this observation several coarse-grainings are applied on the model, boiling it down to its essential parts. This essential model then is analyzed analytically and gives a mathematical proof of the switch‘s nature.

3.2 DERIVING A FULL FRAMEWORK FOR SAC FUNCTIONING

Transiting from metaphase to anaphase can be split up into two parts: activating the SAC, and silencing the SAC. While the model connects those two parts and all reactions run simultaneously, they will be introduced separately for the sake of overview. In total the model consists of 17 biochemical reactions undergoing mass action kinetics, some of which are reversible, describing the interaction of 16 species (cf. Fig. 3.4).

Underlying the SAC is the transition from unattached to attached kinetochores:

$$\mathrm{Kin}_U \xrightarrow{k_{attach}} \mathrm{Kin}_A \tag{3.5}$$

whereby Kin_U and Kin_A denote the numbers of unattached and attached kinetochores, respectively. This process is rather complex and involves the interplay of plenty of proteins [SM09]. For simplicity it is only treated as one reaction of two species as this is sufficient to investigate the functioning of the SAC.

Reaction 3.5 has a stochastic treatment, meaning that kinetochores are discrete species changing their state randomly based on the rate k_{attach}.

3.2.1 *Inhibiting APC/Cs Ability to Ubiquitinate*

The transition from meta- to anaphase is carried out by Separase, which cleaves the sister chromatid, and Cdk1 initiating the mitotic exit, which is controlled and kept inactive by Securin and CyclinB (cf. Fig 3.2), respectively. By Cdc20 activated APC/C ubiquitinates Securin and CyclinB, resulting in the degradation of the latter. Thus, Securin and CyclinB can be seen as output signal of the SAC, because they correspond to a silenced SAC signal. Nevertheless, in this model only Securin is considered as output, as both Securin and CyclinB degrade in the same way (cf. Eq. 3.7 and 3.8), and without further

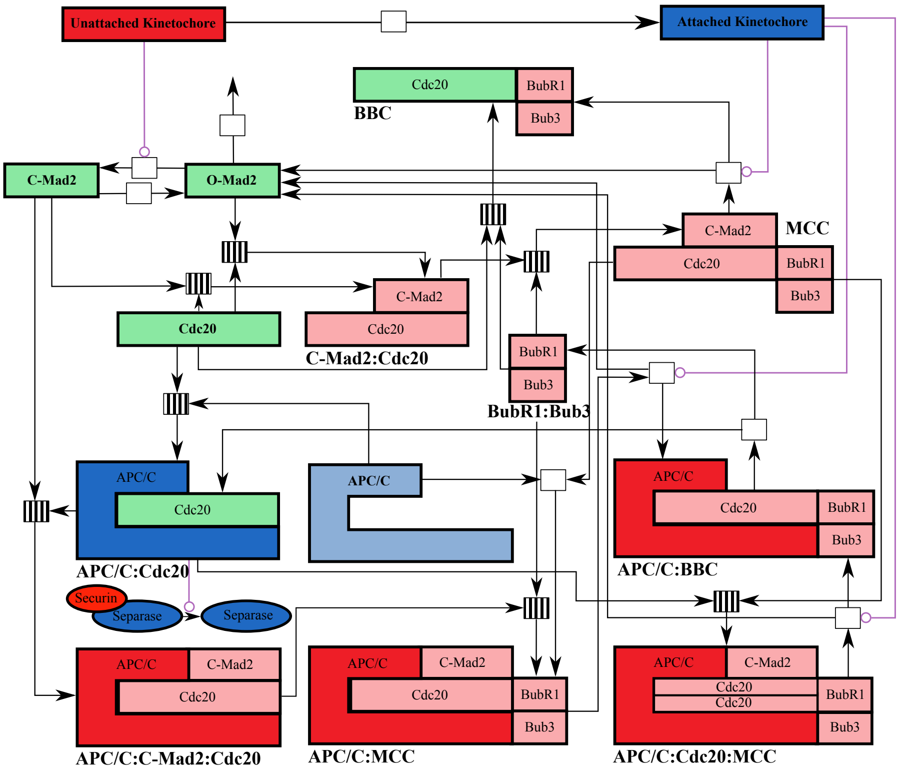


Figure 3.4: Shown is the biochemical pathway to activate and silence the SAC. **Species:** All 16 species are depicted and labeled, in which one bold font represents the species, and normal font the building blocks. **Colors:** Red species are inhibiting, blue ones supporting the transition to anaphase and green ones can have both functions, depending on how they bind. **Reactions:** All 17 biochemical reactions (cf. Eq. 3.5 - 3.26) are visualized as boxes, in which striped ones are reversible reactions and white ones are irreversible. Black arrows indicate the forward reaction and purple ones indicate an enzymatic influence.

information, both would have the same qualitative behavior in the simulation.

$$\mathrm{APC/C} + \mathrm{Cdc20} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{APC/C{:}Cdc20} \tag{3.6}$$

$$\mathrm{Separase{:}Securin} \xrightarrow{k_D \times [\mathrm{APC/C{:}Cdc20}]} \mathrm{Separase} \tag{3.7}$$

$$\mathrm{Cdk1{:}CyclinB} \xrightarrow{k_D \times [\mathrm{APC/C{:}Cdc20}]} \mathrm{Cdk1} \tag{3.8}$$

Activating APC/C (cf. Eq. 3.6) unhindered leads to the immediate decay of Securin and CyclinB and thus the transition into anaphase. For that reason, the aim of the SAC is to control activated APC/C and disable its ability to ubiquitinate. This is achieved mainly through the mitotic checkpoint complex MCC. Izawa et al. [IP12] has shown that a subunit of the MCC, namely Mad2, competes with APC/C for the same binding site in Cdc20. Capturing free Cdc20 by Mad2 prevents this Cdc20 unit from binding APC/C but not APC/Cs ability

MCC, the mitotic checkpoint complex consist of the building block C-Mad2, Cdc20 and BubR1:Bub3.

to bind a different Cdc20 unit. Another part of the MCC, namely the BubR1:Bub3 complex, completely disables the APC/C ability to bind any Cdc20. In conclusion, the MCC disables Cdc20 and APC/C functioning but keeps them at the same time close to each other, so that once all kinetochores are attached and the MCC is being disabled they can quickly interact and start ubiquitinating Securin and CyclinB.

$$\mathrm{APC/C} + \mathrm{MCC} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \mathrm{APC/C{:}MCC} \tag{3.9}$$

As mentioned above, the MCC is composed of Cdc20, Mad2 and the BubR1:Bub3 complex. Forming the MCC is a stepwise process, whereby it is not entirely known where the assembling process takes place [Cha+12]. Initially, O-Mad2 is turned to its closed conformation C-Mad2 which is catalyzed by the Mad1:Mad2 binding site located on unattached kinetochores [How+00], making it the most crucial reaction of the MCC pathway.

$$\mathrm{Mad1{:}Mad2} + \mathrm{O\text{-}Mad2} \underset{k_{-T1}}{\overset{k_{T1}}{\rightleftharpoons}} \mathrm{Mad1{:}Mad2{:}C\text{-}Mad2} \tag{3.10}$$

$$\mathrm{Mad1{:}Mad2{:}C\text{-}Mad2} + \mathrm{Cdc20} \xrightarrow{k_{T2}} \mathrm{Mad1{:}Mad2} + \mathrm{Cdc20{:}C\text{-}Mad2} \tag{3.11}$$

Once Mad2 is converted it binds a Cdc20 and is then released back to the cytosol. The model proposed here simplifies this mechanism to a kinetochore driven Mad2 turnover, which is releases immediately.

$$\mathrm{O\text{-}Mad2} \underset{k_{-3}}{\overset{k3\times[\mathrm{KinU}]}{\rightleftharpoons}} \mathrm{C\text{-}Mad2} \tag{3.12}$$

$$\mathrm{Cdc20} + \mathrm{C\text{-}Mad2} \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} \mathrm{Cdc20{:}C\text{-}Mad2} \tag{3.13}$$

$$\mathrm{C\text{-}Mad2{:}Cdc20} + \mathrm{BubR1{:}Bub3} \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} \mathrm{MCC} \tag{3.14}$$

After this crucial conversion the BubR1:Bub3 complex attaches and forms the MCC.

DeAntoni et al. [DSM05] suggested a mechanism that amplifies the closing of Mad2 into the cytosol (Eq. 3.15 and Eq. 3.16). This so called "Template Model" allows Cdc20:C-Mad2 to turnover Mad2 as well as the kinetochore-bound Mad1:Mad2 complex.

$$\mathrm{Cdc20{:}C\text{-}Mad2} + \mathrm{O\text{-}Mad2} \underset{k_{-T1}}{\overset{k_{T1}}{\rightleftharpoons}} \mathrm{Cdc20{:}C\text{-}Mad2{:}C\text{-}Mad2} \tag{3.15}$$

$$\mathrm{Cdc20{:}C\text{-}Mad2{:}C\text{-}Mad2} + \mathrm{Cdc20} \xrightarrow{k_{T2}} 2\mathrm{Cdc20{:}C\text{-}Mad2} \tag{3.16}$$

In the full model these template reactions are omitted, as previous studies have shown that they have a minor effect with conceivable rates T1 and T2 [Ibr+08b]. Next to the catalyzed formation of Cdc20:C-Mad, Mad2 in its open conformation is also able to bind Cdc20 directly, although at a significantly slower rate (cf. App. B.1).

$$\text{O-Mad2} + \text{Cdc20} \underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}} \text{C-Mad2:Cdc20} \tag{3.17}$$

Howell et al. stated that Mad2 moves along the microtubules [How+00], making the amount of Mad2 a limiting factor in the inhibition process during advanced metaphase. This theory has been modeled and studied extensively by Chen et al. [CL14] and is omitted mostly here for the sake of simplicity, as the focus of this study lies on the biochemical pathway governing the SAC. Introducing solely the decay, corresponding to the moving away from the kinetochore region, is sufficient for studying the switch. In the following expression, ∅ refers to the empty set.

$$\text{Mad2} \xrightarrow{k_{attach}} \varnothing \tag{3.18}$$

Modeling the vanishing of Mad2 along attached microtubules, is done by choosing the decay rate equivalent to the attachment rate of kinetochores. Unfortunately, Mad2 plays a crucial role in inhibiting APC/C, implicating that its loss needs to be compensated to keep the SAC intact.

To overcome this limitation in the inhibition process, Han and colleagues suggested an additional pathway for deactivating APC/C [Han+13]. They introduced the formation of MCC as a two-step catalytic pathway directly at the APC/C by C-Mad2 targeting activated APC/C:Cdc20, which allows Cdc20 to bind the inhibiting BubR1:Bub3 complex and form APC/C:MCC.

$$\text{APC/C:Cdc20} + \text{C-Mad2} \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} \text{APC/C:Cdc20:C-Mad2} \tag{3.19}$$

$$\text{APC/C:Cdc20:C-Mad2} + \text{BubR1:Bub3} \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} \text{APC/C:MCC} \tag{3.20}$$

Once BubR1:Bub3 is recruited to the APC/C, Mad2 dissociates from the complex and is able to form another MCC (originally every Mad2 molecule was only able to inhibit exactly one APC/C as part of the MCC). The remaining BubR1:Bub3:Cdc20 complex forms the so

BCC, the Bub checkpoint complex is able to disable APC/C without Mad2, but is less stable than MCC.

called BCC [Wes+11].

$$\text{APC/C:MCC} \xrightarrow{k_7 \times [\text{Kin}_A]} \text{APC/C:BBC} + \text{O-Mad2} \tag{3.21}$$

The inhibiting level of the APC/C:BCC complex is not as high as that of APC/C:MCC, as MCC is more stable than BCC (cf. App. B.1 and [Col+13; Han+13]). Additionally, Cdc20 can bind BubR1:Bub3 directly to form the BCC in cytosol.

$$\text{Cdc20} + \text{BubR1:Bub3} \underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}} \text{BCC} \tag{3.22}$$

Cdc20 is the most abundant species in the system (cf. App. B.2) and for that reason cannot be entirely captured by MCC, which is not a desirable behavior because free Cdc20 can immediately activate freed APC/C. Recently, Izawa et al. found that the MCC is able to bind a second Cdc20 molecule that has already activated an APC/C [IP15].

$$\text{APC/C:Cdc20} + \text{MCC} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \text{APC/C:Cdc20:MCC} \tag{3.23}$$

Tight inhibition means that most APC/C are bound to MCC and no free Cdc20 is available.

With this reaction, the concentration of free Cdc20 is close to zero during metaphase (shown later in the result section). Reactions introduced in this section (3.14 - 3.23) guarantee a tight inhibition while kinetochores continue attaching.

3.2.2 *Silencing the SAC to Reactivate APC/C*

The mechanisms governing SAC silencing are still not completely understood [MS07]. However, they must have the following properties to reactivate APC/C quickly:

(1) Extinction of the MCC fabric, namely the Mad1:Mad2 template located on unattached kinetochores.

(2) Disassembly of the APC/C:MCC.

(3) Fast targeting of APC/C by freed Cdc20.

Property 1

It has been postulated that dynein promotes pulling RZZ-Spindly-Mad1:Mad2 complexes away from outer kinetochores along microtubules [How+01]. Thus, once a kinetochore is attached, it is unable to turn over Mad2 from its open to closed conformation.

Property 2

Released p31$^{\text{comet}}$ is the crucial silencing factor and its amount is proportional to the number of attached kinetochores as it is part of the Mad1:Mad2 binding site. Although UbcH10 works on the rejection of Mad2 from APC/C, the strength of this process is dependent on the amount of p31$^{\text{comet}}$ as it lowers the threshold of UbcH10 to become active. Therefore, attached kinetochores catalyze the removal of Mad2 from APC/C:MCC and APC/C:Cdc20:MCC in our model.

Notably, MCC already contains the activator Cdc20 and this means that dissolved MCC results in the direct activation of APC/C. *Property 3*

$$\text{APC/C:MCC} \xrightarrow{k_7 \times [\text{KinA}]} \text{APC/C:BBC} + \text{O-Mad2} \quad (3.24)$$

$$\text{APC/C:Cdc20:MCC} \xrightarrow{k_9 \times [\text{KinA}]} \text{APC/C:BBC} + \text{O-Mad2} \quad (3.25)$$

$$\text{APC/C:BBC} \xrightarrow{k_{10}} \text{APC/C:Cdc20} + \text{BubR1:Bub3} \quad (3.26)$$

The silencing process starts with the first attached kinetochore, counteracting the formation of MCC. After its formation in an initial burst, MCC degrades continuously during metaphase (cf. Fig. 3.5), resulting in the formation of a small fraction of APC/C:Cdc20. However, activated APC/C starts degrading Securin, which explains the rheostat nature of the switch. The Securin level decreases continuously once the attachment process started, eliminating the theory of an all-or-nothing response. Summarizing the following events occur:

(1) In an early state of metaphase, most APC/Cs are inhibited by the MCC.

(2) Attached kinetochores start dissembling the stable APC/C:MCC and APC/C:Cdc20:MCC complexes by stripping away Mad2 via p31$^{\text{comet}}$ and UbcH10 (controlled by rate constants k_7 and k_9).

(3) During advanced metaphase, APC/C:MCC yields the less stable APC/C:BBC complex.

(4) After the last attachment, APC/C is fast-reactivated as APC/C:BBC decays quickly and is no longer being formed.

Reaction (3.6 and 3.26) guarantee the three properties and realize the silencing of the SAC. The full SAC framework is depicted in Fig. 3.4, which includes all 16 species introduced above, and their 17 reactions and has been published previously [HDI17].

3.2.3 *Simulating the ODE Model*

Now the introduced model will be simulated as a system of ODEs. The species are given naturally and the reactions have to be translated into ODEs, with their respective reaction rates. As far as possible, reaction parameters from literature and previously performed *in-vitro* parameter studies have been used (cf. App. B.1). Newly introduced reactions underwent a parameter study (cf. App. C.1). The most crucial parameter turns out to be the kinetochore-allocated turnover from O-Mad2 to C-Mad2. Considering cell size and diffusion parameters, this rate was determined to be a maximal $0.016\mathrm{s}^{-1}$ on every single kinetochore (cf. App. A.4) corresponding to a half-time of around 30 seconds. This is consistent with the experimental findings of Howell et al. [How+00]. The other crucial parameter is the kinetochores counterpart, namely stripping Mad from APC/C:MCC complexes. Here, parameters have been estimated to not exceed $0.015\mathrm{s}^{-1}$ to guarantee a functional SAC. This result will be verified later using the coarse-grained SAC model (cf. Sec. 3.3.4). All other parameters can be varied over a wide range (see the references for the relevant parameter studies), which is consistent with Gutenkunst et al. [Gut+07], who stated that in most biological systems, qualitative behavior is not governed by parameters but by the structure of the relevant network, meaning that the switch is a consequence of the network topology rather than parameter values.

cf. Fig. 3.5 Left, whereby only interesting species are shown and not all.

Simulating the ODE model shows that APC/C:MCC and APC/C:Cdc20:MCC are produced quickly and disable the ability of APC/C to ubiquitinate. Both species decrease steadily and yield the less stable APC/C:BCC complex. MCC and Cdc20:C-Mad2 levels decrease permanently, strengthening the hypothesis of Collin et al. that the SAC works like a rheostat switch. The concentration of free Cdc20 during the attachment process drops close to zero. Contradicting the all-or-nothing response, the level of APC/C:Cdc20 rises steadily until the final attachment, degrading Securin continuously.

Nevertheless, fast switching between disabled and activated APC/C can be seen, taking place a few minutes after the last proper attachment. This results in a peak of Securin degradation, as measured by Dick et al. (cf. Fig. 3.7 and [DG13]). In conclusion, the simulation of the ODE model depicts the switch realistically and is qualitatively as well as quantitatively in line with experimental findings.

3.2.4 *Simulating the Particle Model*

Realizing the full SAC as a particle model requires methods from Section 2 which are explained in detail in Appendix A.5. Results from the particle simulation look qualitatively similar to the outcome of the ODE simulation (cf. Fig. 3.5). The anticipated switching behavior can be seen, although the transition from silenced to activated APC/C is not as sharp as the one found in the ODE simulation. Because attached kinetochores emit $\mathrm{p31}^{\mathrm{comet}}$, which has a high diffusion coefficient, the silencing ought to be quite fast. $\mathrm{p31}^{\mathrm{comet}}$ molecules

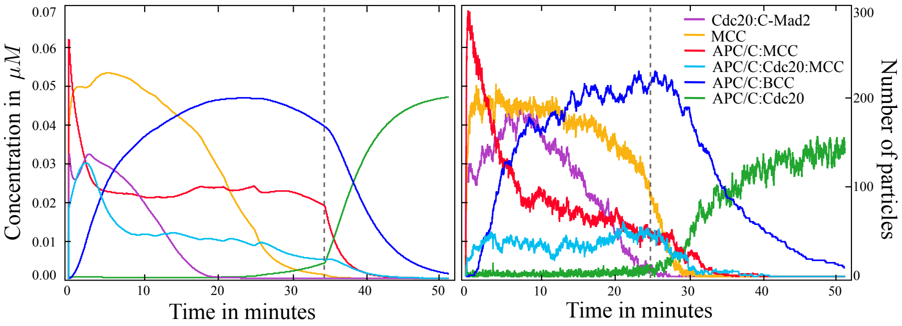


Figure 3.5: Comparison of the outcome from ODE (Left) and particle model (Right). Shown are the concentration curves over time for the labeled species. The dashed line indicates the attachment of the last kinetochore. **Left:** Cdc20:C-Mad2 and MCC are produced quickly and decay continuously. APC/C:MCC and APC/C:Cdc20:MCC rapidly capture and disable all APC/C molecules. Both complexes yield less stable APC/C:BCC. APC/C:Cdc20 is present during the whole process and reaches its full activity in the period directly following the last attachment. **Right:** The general behavior is the same as the one of the ODE curves, with the single difference that the reactivation of APC/C:Cdc20 is slower which is discussed in Section 3.2.4.

are not modeled explicitly, only as increased interaction radius of attached kinetochores, for performance reasons, as the concentration of p31^{comet} is high [Map+06] and its simulation would require a lot of computational power. Nevertheless, this simplification most likely accounts for the slow reactivation of APC/C:Cdc20 after the last attachment in the particle simulation.

An increased interaction radius means that particles do not have to hit the surface of the attached kinetochore to interact with it, which emulates the behavior of a p31 cloud around them.

APC/C:Cdc20 is present and active during metaphase, supporting the rheostat switch in Securin concentration. Aside from fluctuations, which result from the stochasticity of the simulation, all species concentrations are similar to those obtained in the ODE simulation. The setup of the simulation, described in the methods section (cf. Sec. 2.1), leads to the hypothesis that diffusion and reaction rates, as well as the extension of all species, do not play major roles in the functioning of the SAC. As pointed out in the previous section, the kinetochore-allocated reaction does play a major role. In this model, any O-Mad2 particle that enters a kinetochore region is converted into C-Mad2, which makes this reaction only space- and diffusion-dependent. Accordingly, proper SAC functioning requires that proteins are concentrated around the kinetochore plate (cf. Fig. 3.6). Other spatial properties, such as diffusion and reaction rates, as well as the extensions of the proteins, only play minor roles.

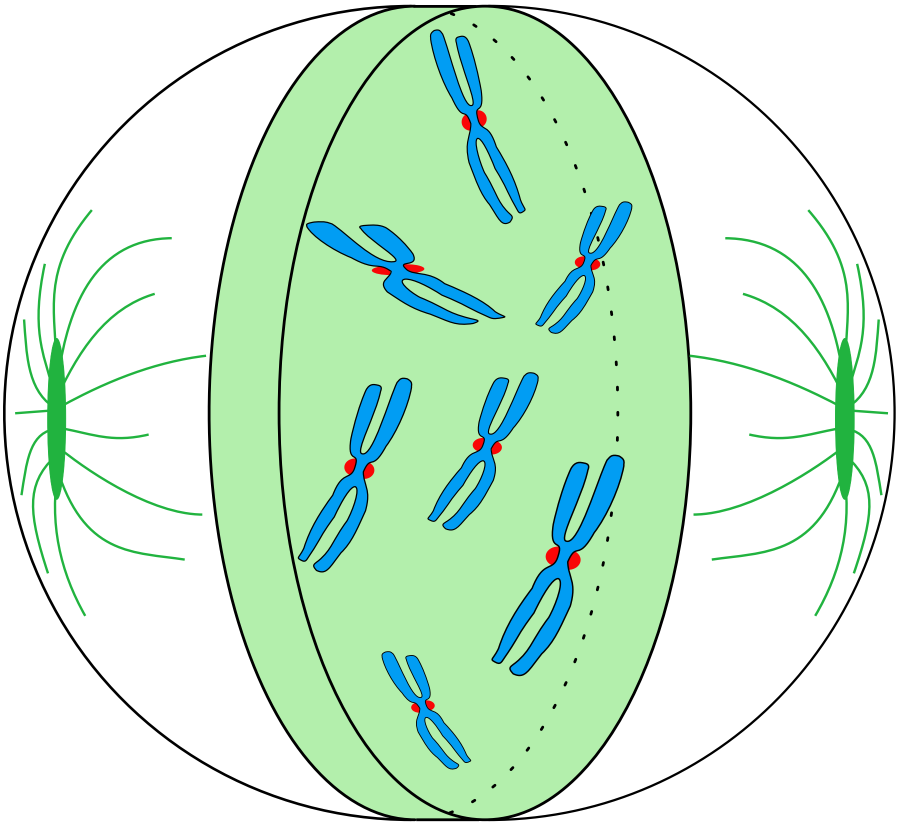


Figure 3.6: Shown is the reaction volume of a cell during mitosis, used in the spatially stochastic particle simulation. Left and right dark green areas symbolize the centromeres, from where microtubules spawn. Blue is the chromosome organized DNA with its kinetochores in red. The whole nuclear space with radius 6μm is modeled, but all species are hold in the light green kinetochore area. This space exclusion is necessary to guarantee fast turnover from O-Mad2 to C-Mad2. All reactions of the full model only take place in the light green area, which can be seen as a well-mixed soup.

3.2.5 *Switching Behavior of the Full Model*

Based on the level of APC/C:Cdc20, Securin is degraded continuously (cf. Eq. 3.7) which was incorporated into the full model to investigate the switching behavior. Furthermore, the influence of different treatments was simulated, namely nocodazole, which destroys the spindle and thus delays kinetochores‘ attachment rate, and Mad targeting siRNA, that blocks Mad2s ability to form MCCs and thus accelerates the decay of Securin as APC/C:Cdc20 is not inhibited. Nocodazole effects have been modeled by defining a fractions of kinetochores unable to attach, dependent on the related nocodazole concentration. SiRNA effects have been realized by decreasing the initial concentration of O-Mad2.

It was found that the full model responds precisely in the same way as the experimental data suggests (cf. Fig. 3.7). The wild type behavior shows a steady degradation of Securin and thus supports the rheostat switch hypothesis, over the all-or-nothing toggle switch. Nocodazole and siRNA influence the degradation rate and the time of anaphase onset (cf. Fig. 3.7C) in the expected manner.

Generating bistability in the system requires necessarily a feedback loop (cf. Sec. 3.1.3). In the full model, this feedback loop could be the dissembling of APC/C:MCC by APC/C, in which APC/C ubiquitinates Cdc20, decreasing its affinity to Mad2, and frees itself from the MCC. However, the incorporation of this feedback loop into the full model would be speculative as there are no kinetic data on this mechanism. While the simulation of the full model required automatable coarse-grainings, introduced in Section 2.2, the following section uses manual methods to simplify the model.

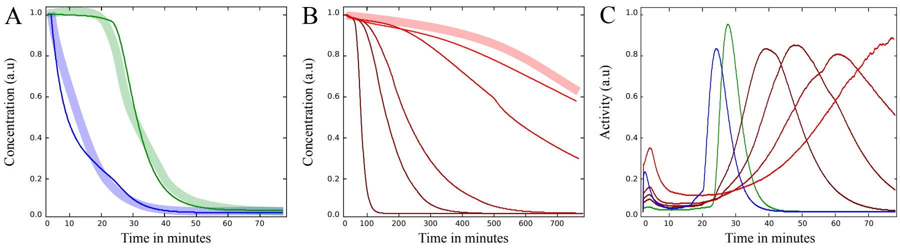


Figure 3.7: Securin behavior in the full model under different treatments. Thin lines denote data from the models' simulations here introduced. Thick lines represent data points from Dick et. al Fig. 5 [DG13]. **A:** Total concentration of Securin, normalized at prometaphase onset (Time $= 0$). The green curve shows the unperturbed system, in which Securin starts degrading at a significantly higher rate once kinetochores start attaching. The blue curve shows the influence of Mad2 targeting siRNA, simulated through the reduction of Mad2 initial concentration. Here, APC/C:Cdc20 is not captured by MCC and Securin starts degrading immediately. **B:** The same as in A but under the influence of different concentrations of nocodazole, modeled by differing numbers of kinetochores that are unable to attach. Curves are shown for $1-80$ permanently unattached kinetochores (shown by different shades of red). It is seen that the increasing of nocodazole directly prolonges the metaphase. **C:** The degradation rate of Securin (control in green, targeting siRNA in blue and nocodazole in red). As the rate peaks briefly before anaphase onset, these curves demonstrate how different drugs can extend metaphase and delay anaphase.

3.3 deriving coarse grained models

In order to demonstrate the effects of a coarse-grained model the following sections derive those models and will compare them to the full model. Furthermore, those coarse grained variations allow the performance of a bifurcation analysis (cf. Sec. 3.1.3).

Comparison included the calculation time and qualitative behavior of the concentration curves, as well as the switching mechanism.

Table 3.1: **Summary of the simple reduced model**

Abstract species	**Detailed species**	**Remark**
Kin_A	Kin_U	Kinetochore attached
Kin_U	Kin_U	Kinetochore unattached
Promotor	APC/C	Promotor
Activator	Cdc20	Activator of the Promotor
Promotor_A	APC/C:Cdc20	Active Promotor
Inhibitor	O-Mad2, C-Mad2, MCC, C-Mad2:Cdc20, BubR1:Bub3	Inhibitor of the Promotor
Promotor_I	APC/C:MCC, APC/C:BCC, APC/C:C-Mad2:Cdc20, APC/C:Cdc20:MCC	Inhibited Promotor

3.3.1 *Simple Reduction by Lumping Species*

Typically those species are lumped that have a similar function or a redundant.

Reducing the full SAC model is achieved by lumping several species manually into abstract species like "Promotor", "Activator", and "Inhibitor". As a result, the model reduces to 7 species and 9 reactions, making it more comprehensible, but still leaving a relatively large state space with complex dynamics. The relation between species of the two models can be obtained by Table 3.1, mapping detailed species of the full model to abstract species of the reduced model. As well as species, the reactions collapse to the following:

$$\text{Kin}_\text{U} \xrightarrow{k_{attach}} \text{Kin}_\text{A} \tag{3.27}$$

$$\text{Activator} \underset{k_{-3}}{\overset{k3\times[KinU]}{\rightleftharpoons}} \text{Inhibitor} \tag{3.28}$$

$$\text{Inhibitor} + \text{Promotor} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \text{Promotor}_\text{I} \tag{3.29}$$

$$\text{Activator} + \text{Promotor} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \text{Promotor}_\text{A} \tag{3.30}$$

The reactions above represent the basic attachment (cf. Eq. 3.27), as well as the production of the Inhibitor (cf. Eq. 3.28, corresponds to the MCC pathway), the inhibiting of the Activator (cf. Eq. 3.29, corresponds to the formation of APC/C:MCC) and the activation (cf. Eq. 3.30, formation of APC/C:Cdc20).

As well as in the full model, the attached kinetochores dissemble the inhibited Promotor:

$$\text{Promotor}_\text{I} \xrightarrow{k_7 \times [\text{KinA}]} \text{Promotor}_\text{A} \qquad (3.31)$$

Furthermore, Inhibitors increase the decay rate of activated Promotors, corresponding to the binding of MCC to an already activated APC/C:Cdc20.

$$\text{Promotor}_\text{A} \xrightarrow{k_{-2} \times [\text{Inhibitor}]} \text{Promotor} + \text{Activator} \qquad (3.32)$$

Rates are the ones from the full model, where the slowest one was always chosen if a reaction corresponds to multiple original ones. The framework for the simple reduced model is shown in Figure 3.8. Note that the model reduction was done manually, and is not possible automatably, as information about the system is inserted (e.g. which species are lumped as an Inhibitor).

3.3.2 *Simulation and Switch Behavior of the Simple Reduced Model*

The model was simulated as ODE as well as a particle model, and the same realization was used as for the full model (cf. App. A.5). As the results were qualitatively the same, only the outcome of the particle model is shown in Figure 3.8Right.

Securin curves are not shown here, as they look perfectly the same as in Figure 3.7, which is not surprising as it only depends on APC/C:Cdc20 corresponding to activated Promotor.

While the results of the reduced model did not vary a lot, the computational time was accelerated sparsely, which is not significant, considering that the graining is not automatable. Despite the drastic reduction of the model, a bifurcation analysis is still not possible (cf. Sec. 3.1.3). Additionally, a graining always comes along with a loss of information about the system, because species are removed and cannot be studied individually.

3.3.3 *Essential Reduction for Bifurcation Analysis*

Reducing the previous model by another species and four reactions leads to the essential SAC model, small enough to allow a bifurcation analysis (cf. Sec. 3.1.3). Species names are the ones from the full model, and it can be seen as reducing it down to the core mechanism

The essential model is similar to the one proposed by Verdugo et al. [Ver+13].

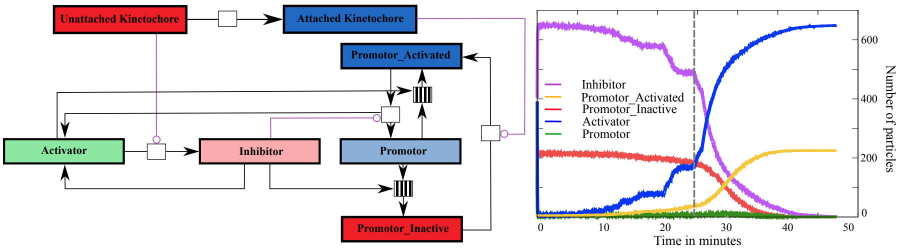


Figure 3.8: **Left:** Framework of the simple reduced model with the meaning of the symbols the same as those of the full model (cf. Fig. 3.4). **Right:** Particle number of time curve of the realized reduced model. Its qualitative behavior is the same as the one of the full model, after the last proper attachment (dashed line) the Promotor is quickly activated.

essential for explaining the SAC switch (cf. Fig. 3.9). Apparently, there exist even smaller models, showing a switching behavior, namely the ones proposed by Doncic et. al. [DBJB05]. These models are minimal as they only consist of four species and three reactions. Nevertheless, they are too grained and do not allow deeper analysis. This is why the model treated here model is the smallest one with essential properties.

Essential properties mean that a rheostat switch can be seen during the attachment process, while the majority of the inhibiting species is present during the attachment.

$$\mathrm{Kin_U} \xrightarrow{k_{attach}} \mathrm{Kin_A} \tag{3.33}$$

$$\mathrm{Mad2} \xrightarrow{[\mathrm{Kin_U}]\times k_3} \mathrm{MCC} \tag{3.34}$$

$$\mathrm{MCC} + \mathrm{APC/C} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \mathrm{APC/C{:}MCC} \tag{3.35}$$

$$\mathrm{APC/C{:}MCC} \xrightarrow{[\mathrm{APC/C}][\mathrm{Kin_A}]\times k_7} \mathrm{APC/C} + \mathrm{Mad2} \tag{3.36}$$

$$\mathrm{APC/C{:}Cdc20} + \mathrm{Securin} \xrightarrow{k_D} \mathrm{APC/C{:}Cdc20} \tag{3.37}$$

Equation 3.37 only serves to generate the output signal, as done in the full model (cf. Eq. 3.7). All initial concentrations and reaction rates are identical with the full model where possible (cf. App. B.1). If reactions in the reduced model differ from these in the full model, new rates have been introduced.

3.3.4 *Simulation and Switch Behavior of the Essential Model*

The outcome of the simulated essential model in Figure 3.9 shows the switch between active and inactive APC/C as well as the steady degradation of MCC. Even this small model explains the SAC that locks the cell in metaphase until all chromosomes are attached. Concentration plots of Securin under different treatments are shown in

Figure 3.10Left, and these coincide with the response found in the detailed model and with experimental findings (cf. Fig. 3.7 and [DG13]).

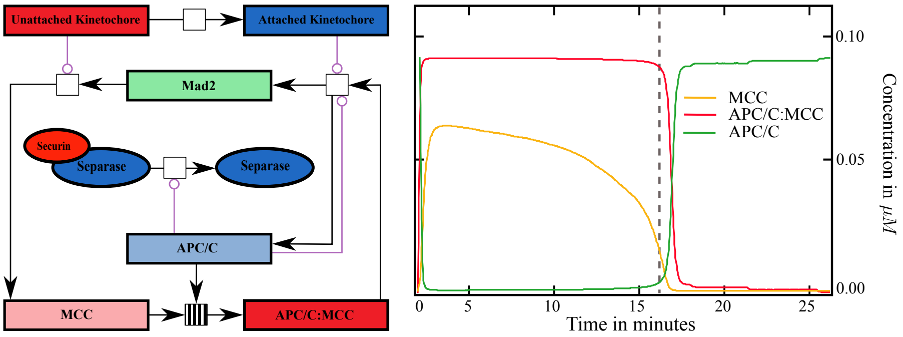


Figure 3.9: **Left:** Framework of the essential model, with the meaning of the symbols is the same as those in the full model (cf. Fig. 3.4). **Right:** Concentration of time curve of the ODE simulated essential model. Its qualitative behavior is the same as in the full model, after the last proper attachment (dashed line) the APC/C is quickly produced.

Eventually the bistability of the essential model can be demonstrated by performing a one parameter bifurcation analysis. The total concentration [MCCT] is under study, dependent on the number of unattached kinetochores Kin_U in the range from 0 to 92, with aim to investigate the steady state concentration of [MCCT].

Evaluating the quantitative behavior requires the total concentrations of Mad2, MCC and APC/C, which are derived from Figure 3.9. The total concentration of APC/C, written as [APCT] = [APC/C] + [APC/C:MCC], is constant; likewise, the total amount of C-Mad2, [MadT] = [C-Mad2] + [MCC] + [APC/C:MCC]. MCC‘s concentration is given by [MCCT] = [MCC] + [APC/C:MCC] and is time dependent according to:

$$\frac{d[\mathrm{MCCT}]}{dt} = k_3 \mathrm{Kin}_U([\mathrm{MadT}] - [\mathrm{MCCT}]) - k_8 \mathrm{Kin}_A [\mathrm{APC/C}][\mathrm{APC/C{:}MCC}] \tag{3.38}$$

Substituting the constant concentrations [APCT] and [MadT] into the steady state condition for APC/C:MCC gives the following

$$k_2[\mathrm{MCC}][\mathrm{APC}] = k_{-2}[\mathrm{APC/C{:}MCC}] + k_8[\mathrm{APC/C{:}MCC}] \tag{3.39}$$

which results in an equation for APC/C:MCC in the steady state, only depending on the total concentrations:

$$B = k_m + [\mathrm{APCT}] + [\mathrm{MCCT}] \tag{3.40}$$

$$[\mathrm{APC/C{:}MCC}] = \frac{B - \sqrt{B - 4[\mathrm{MCCT}][\mathrm{APCT}]}}{2} \tag{3.41}$$

whereby k_m is the relation $(k_{-2} + k_8)/k_2$. This analytical pre-calculation allows the generation of the bifurcation diagram, shown in Figure 3.10Right, which has the typical s-shaped curve of a bistable switch. It demonstrates clearly that all kinetochores have to be attached to disengage the SAC, while at least one must reattach to reengage the SAC. Using Securin instead of [MCCT] as steady state concentration would result in a 0-line, indicating that Securin will decay entirely, independent of the amount of kinetochores.

The Securin decay can take several hours, which is not depicted in the bifurcation diagram, but only the steady state.

The model is most sensitive to k_{M1} as this can lead to a defect in SAC functioning. If the kinetochore driven Mad2 turnover is too slow, the inhibition of APC/C cannot take place and the cell moves to anaphase prematurely. Assuming no active transport of Mad2, the described turnover solely depends on the diffusion rate, kinetochore and cell size, limiting the rate. Experimentally, the rate k_{M1} has been estimated to be $0.016s^{-1}$ [How+00; Fae+17]. Figure 3.10, shows clearly that the SAC does not work properly if this rate is decreased, as the last kinetochore could not maintain the inhibition, resulting in a premature exit, which confirms the wet-lab results impressively.

Theoretical calculations can be found in Appendix A.4.

This finding shows that it is possible to observe a continuous degradation of one species (Securin), leading to the assumption of a rheostat switch, while the underlying core components (MCCT) behave like a toggle switch.

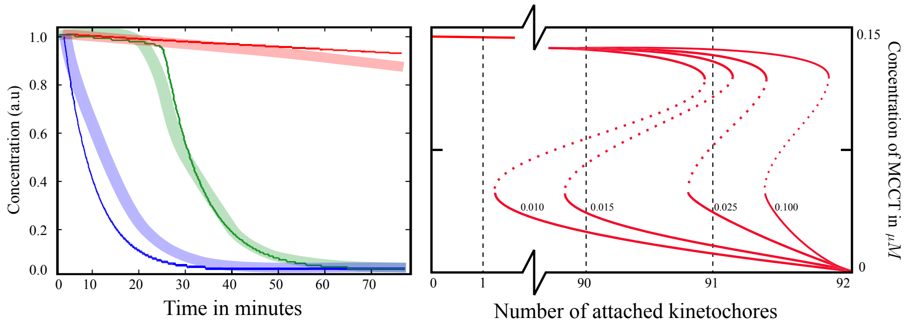


Figure 3.10: **Left:** Securin concentration of the essential model under different treatments for in-vitro (thick lines) and in-silico (thin lines). The green curve shows the wild type, blue shows the influence of Mad2 targeting siRNA and red shows the influence of nocodazole. In all situations, Securin degrades steadily, validating the rheostat switch hypothesis. **Right:** One-parameter bifurcation diagrams showing the activity of the MCC against the number of attached kinetochores using different rates kM1 (labeled next to the curves). With a properly chosen rate, the dashed red lines clearly show that all kinetochores have to be attached to completely silence the MCC. To re-engage the SAC, kinetochores have to reattach. If the rate drops below the estimated threshold, the SAC is silenced before all kinetochores are attached.

Table 3.2: **Four State Model.** Initially all kinetochores are "Unattached" and only Inhibitor species are present. Immediately after the transition into "Attachment" only Inhibitors are present still. After a while in this state, Inhibitor are transformed to Promotors and the kinetochore goes into "About to exit" state. If all Inhibitors vanished and only Promotors are left, the kinetochore is in "Exit" state.

State	KinU	KinA	Promotor	Inhibitor
Unattached	x			x
Attached		x		x
About to exit		x	x	x
Exit		x	x	

3.3.5 *Transition to Exact Stochastic Markov Model*

Previously introduced models and their simulations always contain rounding errors, as they are solved numerically. One method to get an exact result is the application of a Markov model (cf. Sec. 1.2.1). The downside of this method is the limitation in species and their amount due to state space explosion. For that reason a model is derived small enough to be evaluated exactly with the PRISM software [KNP11] that explores the state space of a system based on the Markovian state transition probabilities.

Analytical solutions are practically impossible, cf. App. A.1.

A drastic reduction of the state space is achieved through the removal of all species, except kinetochores, which now can be in four different states, namely "Unattached", "Attached", "About to exit" and "Exit." Each of those states represents a combination of species that are present nearby a kinetochore. Table 3.2 introduces the four states a kinetochore can be in and which species are present in its surrounding, whereby Promotor refers to activated APC/C and Inhibitor to MCC, APC/C:MCC, and APC/C:BCC.

With the states defined only the transitional probabilities are missing (cf. Fig. 3.11). Transitions from “Unattached” to “Attached” state and to “About to exit” happen spontaneously (rates k_A and k_{AE}, respectively), as the mikrotubuli attach randomly and kinetochores start immediately to dissemble APC/C:MCC. Changing to the “Exit”

k_A and k_{AE} are probabilities that are applied at every time-step.

state requires a spatial feature, namely that all neighboring kinetochores of a certain one are "About to exit" as well. If the neighborhood of a certain kinetochore contains at least one "Unattached" or "Attached", its state transits form "About to exit" to "Attached".

$$k_E(K) = \left(|\Omega(K)| = \sum_{c \text{ in } \Omega(K)} c \text{ in state "About to exit" or "Exit"} \right) \tag{3.42}$$

Equation 3.42 defines a Boolean variable that determines if a kinetochore K transits from "About to exit" to "Exit", based on its neighboring kinetochores $\Omega(K)$. This models the property that a single kinetochore is able to withhold premature transition to anaphase.

Analyzing the four state model is done in two different ways, firstly numerically by sampling over multiple simulated time-series, and secondly, exactly with the PRISM model checker.

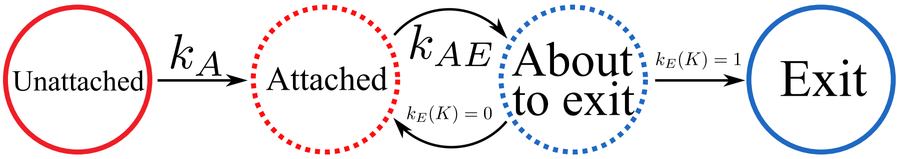


Figure 3.11: Markov chain for each kinetochore shows the transitional probabilities between the four states. From "unattached" to "attached" to "about to exit" has a fixed probability, while the "exit" state only is reached once all neighbors $\Omega(K)$ are also "about to exit".

3.3.6 *Simulating the Four State Model Numerically*

Probability $k_A := 0.005$, which correlates with the duration of metaphase.

The SAC failed if kinetochores in state "Unattached" and "Exit" occur simultaneously.

To simulate the 4-state model a kinetochore is defined as a point, randomly positioned in a circle of radius R, corresponding to the diameter of the nucleus. The initial state of all 92 kinetochores is "Unattached", which switches spontaneously to "Attached". The parameter k_{AE} describes the quantitatively unknown transition from "Attached" to "About to exit" and is varied within $[0, 1]$. The second parameter of the study is the neighborhood relationship, which is a variable radius $r \in [0, R]$ within all kinetochores count as neighbors. Outcome of this setup is the probability of a functioning SAC. This, in the following described algorithm, returns *True* if the SAC was functioning in every time-step and *False* otherwise.

(1) **Input:** Rate k_A, k_{AE} and neighbor radius r; fixed stepsize Δt; simulation time T_{end}.

(2) **Initialize:** Set state vector S with 92 entries corresponding to kinetochores, initially to "Unattached"; set for each kinetochore a random placement in circle with radius R; generate for each kinetochore a list of neighbors N, dependent on r; set time $t = 0$; set $t_{step} = 0$, set validity counter $\nu = 0$.

(3) **Iterate:** Repeat while $t < T_{end}$:

- (3.1) **Iterate over each entry of the state vector S:**
  - (3.1.1) **State = "Unattached":** Draw uniform random variable, compare with k_A and in case set state to "Attached".
  - (3.1.2) **State = "Attached":** Draw uniform random variable, compare with k_{AE} and in case set state to "About to exit".
  - (3.1.3) **State = "About to exit": If** all neighbors of the regarding entry $N(S)$ are "About to exit" or "Exit", set state to "Exit", **else** set state to "Attached".
- (3.2) **Check for validity: If** not any state ="Unattached" and any state ="Exit", $\nu \leftarrow \nu + 1$.
- (3.3) **Increase time:** $t \leftarrow t + \Delta t$; $t_{step} \leftarrow t_{step} + 1$.

(4) **Return:** If $t > T_{end}$ return $\nu / t_{step} = 1$.

Averaging over multiple numerical simulations results in a 3D plot, shown in Figure 3.12. While the abscissa and the ordinate hold the parameters under study (k_{AE} and r, respectively), the applicate presents the probability of a functioning SAC. It is shown that the spontaneous rate to transit to "About to exit" does not matter, as long it is greater than 0. The radius on the other hand plays a major role, as it influences the outcome directly, and a 100% functioning SAC only is realized when r is maximal, meaning that every other kinetochore is considered a neighbor. This result underpins the finding that even a single kinetochore can withhold anaphase onset. Note that the spatial distribution of kinetochores does not matter, only the radius, determining the fraction of kinetochores considered a neighbor.

3.3.7 *Exact Analysis of the Four State Model*

Using PRISM now to validate the result exactly is possible with all 92 kinetochores, as a state space of up to 10^{10} can be handled by the software. It is not possible to include space in PRISM, so instead of a radius r the amount of kinetochores is varied, corresponding to

The state space of PRISM using 92 kinetochores is 146,877 between which 560,655 transitions exist.

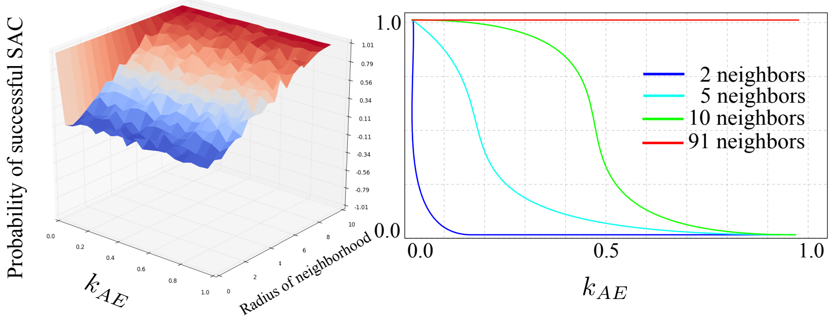


Figure 3.12: Analyzing the results of the four state model, with the probability of a successful SAC as output function (y-axis). **Left:** Outcome of the numerical simulation depicted as 3D plot of the time averaged results. It shows clearly that only the neighborhood relationship r influences the outcome and the rate k_{AE} does not play a major role. Furthermore it supports the theory of a fully connected graph of kinetochore, meaning that a proper cell division only is guaranteed if the SAC signal is global and not only local. **Right:** Shown are the exact PRISM probabilities of a functioning SAC with various transition probability k_{AE}, for four different spatial configurations, expressed through the amount of kinetochores that are considered a neighbor. Apparently, the SAC depends on this neighbor relationship, as an increased amount yields a higher chance for a successful SAC. Nevertheless, while the rates are increasing steadily, they only reach 100% if all kinetochores are treated as neighbors.

the neighborhood radius. The box below shows the PRISM input file with S the amount of kinetochores, n the number of neighbors and k_A and k_{AE} the regarding probabilities:

Listing 3.1: KinetochoreN.pm

```
ctmc

const double p;
const int n=92;
const int k=4;

module kinetocore

dis:[0..n] init n;
con:[0..n] init 0;
set:[0..n] init 0;
ext:[0..n] init 0;
win:[0..1] init 0;
los:[0..1] init 0;

        [] dis>0 & con<n-> 0.005*dis:(dis’=dis-1)&(con’=con+1);
        [] con>0 & set<n -> p*con:(con’=con-1)&(set’=set+1);
        [] set>0 & set+ext<n & con<n -> set*(1-pow((set+ext-1)/(n
            -1),k)):(set’=set-1)&(con’=con+1);
        [] set>0 & set+ext>1 & dis=0 & ext<n -> set*pow((set+ext
            -1)/(n-1),k):(win’=1)&(set’=set-1)&(ext’=ext+1);
        [] set>0 & set+ext>1 & dis>0 & ext<n -> set*pow((set+ext
            -1)/(n-1),k):(los’=1)&(set’=set-1)&(ext’=ext+1);
endmodule

rewards
        true:1;
endrewards
```

with the properties

Listing 3.2: Properties.pctl

```
P=? [los=0 U win=1]
```

The outcome of this exact stochastic analysis coincides with the numerical simulation, namely that the amount of neighbors determines a successful SAC. The difference between the two approaches is that the probability k_{AE} has a reversed effect than the radius to a functioning SAC, meaning the smaller the value k_{AE} the higher the likelihood of a functioning SAC which could not be seen in the numerical simulation. This may be due to the lack of spatial properties, namely that the numerical simulation always considers the same neighbors (neighbor list N) and PRISM checks the neighbor relationship by taking any kinetochores.

3.4 CONCLUSION

The spindle assembly checkpoint is biologically a highly investigated, but computationally a still challenging system. While the basic understanding is well known (cf. Fig. 3.2), with plenty of models emerging during the last decade, no biochemically reliable model exists yet. In this chapter, the SAC switch was explored extensively and based on recent findings embedded into a full biochemical model (cf. Fig. 3.4). It has been shown in detail how the methods introduced in the previous section (cf. Sec. 2.1) are applied to realize a particle simulation that is qualitatively and quantitatively in accordance with wet-lab measured data, as well as the ODE simulation, as the particle simulation did not reveal a major role of the included spatial properties (cf. Fig. 3.5). Proceeding from this model, different manual coarse-grainings have been accomplished to not only reduce the simulation, but also the model's complexity (cf. Sec. 3.3.4), resulting in a stochastic compartmentalized model that is exactly evaluated. Despite the drastic reductions, all emerged SAC variations behave qualitatively similar and show the desired switching behavior, while their computational time is reduced immensely.

Recent studies proposed that the SAC is rather a rheostat than a toggle switch [Col+13; DG13], based on the behavior of Securin and CyclinB. The bifurcation analysis of the essential model supports this theory solely on the Securin level (cf. Fig. 3.10), but also reveals that some other components, namely the MCC, toggle. This finding clarifies that the SAC switch cannot be reduced to a single concentration curve but is most likely an interplay of multiple mechanisms, which demand further investigation.

In conclusion, this chapter has shown how an ODE model is translated into a particle based model and how coarse-graining methods can be applied to simulate complex system in a feasible amount of time. Furthermore, multiple approaches are performed to reduce complex models to their essential core. This allows the study of, among others, biological systems and to reveal their mechanisms, which are challenging to observe in the wet-lab.

The weakness of the above proposed methods stems from the manual-grainings, as further understanding is required of the system's function. Future work would be to develop a general language that allows the consistent formulation of systems and their function, which would allow the reduction automatically , which currently is only viable for a limited set of systems.

4 COARSE-GRAINING THE FORMATION OF PML NUCLEAR BODIES

This chapter focuses on the self-assembly process of PML nuclear bodies. While the set of rules leading to their formation is small, the state-space is quasi infinite, benefiting tremendously from a rule-based approach to simulate the formation process. In close co-operation and cross-feedback with biologists the first spatial model emerged, explaining critical features of the life cycle of PML bodies in space. Comparing spatial structures with microscopic data is not simple, calling for the development of several tools to analyze and compare data of different sources.

4.1 SELF-ASSEMBLY AND PML NUCLEAR BODIES

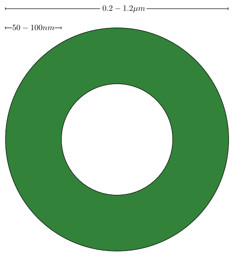
4.1.1 *Introduction*

PML nuclear bodies are dot-like protein-complexes located in the nuclei of most mammalian cells in the interchromatin space between chromosomes territories [Bri+98]. In the 1960 they were identified by electron microscopy using anti-PML protein antibodies [RB+60]. The main protein that organizes the bodies was found to be the PML protein. In the absence of PML, no nuclear dots are formed [LB+01]. PML is expressed in six different isoforms, all of which localize in nuclear bodies as well as the surrounding nucleoplasm [WP+08]. It appears that the majority PML molecules is not bound in nuclear bodies [LB+01]. PML bodies recruit more than 100 different proteins, with Sp100, SUMO and DAXX being the most prominent and best described ones [Szo+90; NM01]. All these PML body components have in common the ability to become SUMOylated [BP07]. SUMO, an ubiquitin-like protein, plays an important role in nuclear body assembly, as its isoforms SUMO1, SUMO2 and SUMO3 are necessary to recruit the other proteins.

Function While the function of individual PML body proteins is mostly known (e.g. PML protein inhibits the infection of certain types of viruses [CA+98], Sp100 has been shown to increase the transcription of certain genes and is responsive to interferons[Was+02]), the precise biochemical or biophysical function of the assembled nuclear dots is still unclear. Several studies suggest that PML bodies are involved in basic cellular functions, such as DNA repair, cellular senescence, stem cell renewal, protein modification, cell proliferation and apoptosis [DBJ04; SP02; BPP08]. Nevertheless, the underlying biochemical details regarding the involvement of the bodies in these processes is still elusive [LB+01]. Furthermore, it has been shown, that PML-/- mice, which are unable to assemble nuclear dots, develop well and live normally, although their chance of acquiring cancer and developing tumors is increased, compared to wild-type mice [SP02].

PML-/- or PML +/+ refers to the mutation of PML protein, positive or negative, respectively.

Structure Nuclear bodies vary in diameter between $0.2 - 1.2\mu m$, with a $50 - 100nm$ thick outer shell under normal growth conditions, independent of the radius [Lan+10]. Note that the shape depends on the size and particular situation, usually being a sphere, but also as horseshoe-like shapes. The shell is mainly formed by PML and SUMO molecules, as their depletion results in disperse distribution of the other components. The inside of the shell was found to either contain SUMO isoforms, DNA/RNA or to be entirely free of any protein [Dyc+94; Lan+10].



Schematic illustration of PML nuclear body cross-section.

During the cell cycle, PML nuclear bodies alternate between a state with full incorporation of all PML nuclear body constituents in interphase (in this thesis referred to as active state) and an accumulation state in which only PML molecules are present during cell division (in this thesis referred to as resting state). Entering mitosis, PML nuclear bodies become insolubly aggregated into so called “mitotic accumulations of PML protein” (MAPPs). During prometaphase, nuclear bodies are phosphorylated, resulting in desumoylation. PML molecules in MAPPs do not exchange, proving their stability during mitosis.

It is still largely unclear how nuclear bodies are assembled and stay localized during G1, S, and G2-Phase. It is possibly a self-organized assembling process as proposed by [CLR11]. In this chapter, the first spatial model is introduced that shows how the assembly process may occur, taking into account all available data. The SRSim tool is used to simulate the basic model and to study overexpression and depletions as well as the insertion of artificial PML molecules and to investigate the minimal requirements of the formation.

More elaborated details on the function and structure of PMl nuclear bodies can be found in [Hoi+18].

4.1.2 *Structural Properties of Important PML Nuclear Body Proteins*

The model developed here considers only the core players in the formation of PML nuclear bodies, namely PML protein, Sp100, DAXX,

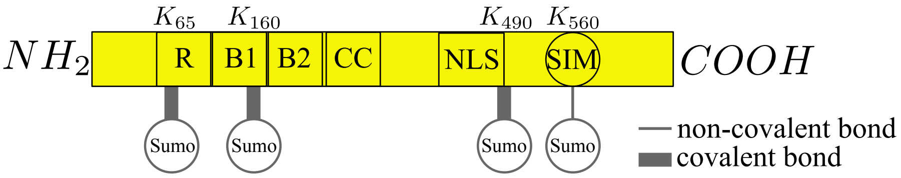


Figure 4.1: Schematic illustration of PML protein structure. The RB1B2CC TRIpartite motif, consists of the zinc RING (R), the B1 and B2 boxes and an alpha helical coiled coil domain (CC). Residues K_{60}, K_{160} and K_{490} are targets for covalent SUMO modification (also PML in the resting state, indicated by the thick bond). PML isoforms I - V additionally have a SUMO interactive motif (SIM), which is able to bind SUMO non-covalently (indicated by the thin bond). Note, that the sizes of the boxes do not correlate with the length of the regarding motif.

SUMO1, SUMO2 and SUMO3. Properties and interaction boxes of all proteins are introduced in the following.

PML Promyelocytic leukemia protein (PML) is the product of the PML gene, located on chromosome 15, with a length of ≈ 700 amino acids. Due to splicing alternatives, the PML gene yields more than 15 isoforms, varying in their C-terminal domain. Nevertheless, all isoforms contain the TRIpartite motif (TRIM/RBCC), which is essential and required for PML nuclear body assembly [PP01]. More specifically, the RBB part is involved in the initial formation of nuclear body seeds, supported by the coiled coil domain which mediates further PML homo-dimerization and thus contributes to the spatial stability in the resting state. PML isoforms I - VI are found in nuclear bodies and contain (except PML VI) additionally a nuclear localization signal (NLS), able to form another covalent bond at the Lys-Gly at K_{490}, and a SUMO interacting motif (SIM) at K_{560} [Kha+01]. Due to their structural similarity, similar biophysical parameters and molecular mass all PML isoforms are treated the same in this thesis, hence referred to as PML protein (cf. Tab. B.4) A structural overview of a prototype PML protein is shown in Figure 4.1.

DAXX and Sp100 DAXX, the death-associated protein 6, is a 740 amino acid long protein, which is encoded by the DAXX gene. It contains three coiled coil domains (at K_{180}, K_{358} and K_{430}) and two NLSs (at K_{391} and K_{628}), whereby the latter one can bind SUMO1 covalently at Lys-Gly at K_{630}. DAXX contains two SIM motifs one each at the N- and C-terminal end (K_1 and K_{733}) where it can be SUMO associated (cf. Fig 4.2, [EC+10]). At K_{142} SUMO2 can be linked covalently, but less likely than the SUMO1 interaction at K_{630} for which reason it is omitted in the model.

Sp100 is a nuclear antigen with a length of 878 amino acids. It contains two HMG boxes (in the region between K_{677} and K_{837}) that

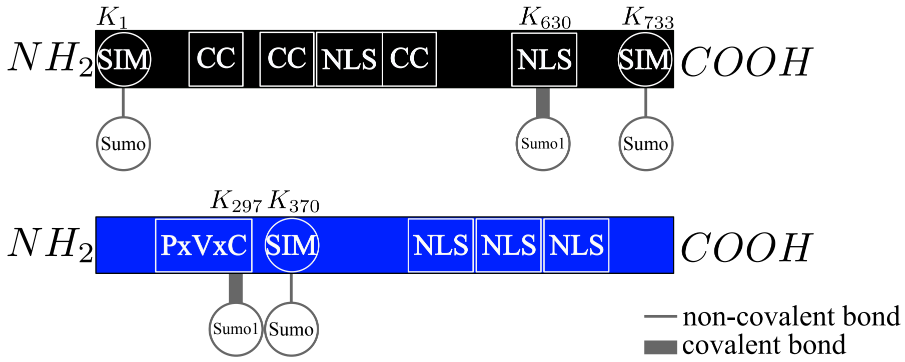


Figure 4.2: Schematic illustration of Sp100 and DAXX proteins. **Top:** Structure of a Daxx protein, with its three coiled coils (CC) and the two nuclear localization signals (NLS), whereby one can bind SUMO1 covalently. The two SIMs are located at the N- and C-Terminal, respectively. **Bottom:** Structure of Sp100 protein, with its triple NLS motif. The denoted PxVxC motif is able to bind SUMO1 covalently, at the Lys-Gly at K297. Its SIM is positioned at K_{370}. Note that the sizes of the boxes' motifs do not correlate with the length of the regarding motif.

are able to bind DNA, and three NLSs (at K536, K568 and K717) that are able to transport the protein into the nucleus. The Lys-Gly domain at K297 is part of a PxVxL motif and able to bind covalently to SUMO1 [SJW97]. Its single SIM is located at K366 [Kni+08]. A structural overview of DAXX and Sp100 is provided in Figure 4.2.

SUMO1 and SUMO2/3 Small Ubiquitin-like modifier (SUMO) proteins attach covalently to proteins, to vary their function, but unlike ubiquitin, SUMO is not a signal for their degradation. This so called SUMOylation takes place in multiple cellular process, e.g. apoptosis, response to stress, or progression through the cell cycle [Joh04]. On their C-terminal Lys-Gly residue SUMO proteins can form covalent bonds with the acceptor of target protein. SUMO interactive motifs of other proteins probably bind through a β-strand to the β2 -strand of SUMO proteins non-covalently [Hec+06].

SUMO2/3 are 95 and 103 amino acids long, respectively, and ~ 96% identical. Both have a Lys-Gly domain at K11 where they can bind covalently other SUMOs and thus form chains. SUMO1, with a length of 101 amino acids, is only ~ 45% identical with SUMO2/3. In particular SUMO1 does not posses the K11 site and is therefore unable to form chains via this residue. It was therefore proposed that SUMO1 terminates SUMO2/3 chain elongations. Recent studies by the Hemmerich lab found that SUMO also has its own SIM, which probably is responsible for the SUMO-SUMO interaction explained later (cf. Sec. 4.4.3; unpublished data).

Figure 4.3 shows a structural overview of the PML body relevant SUMO proteins.

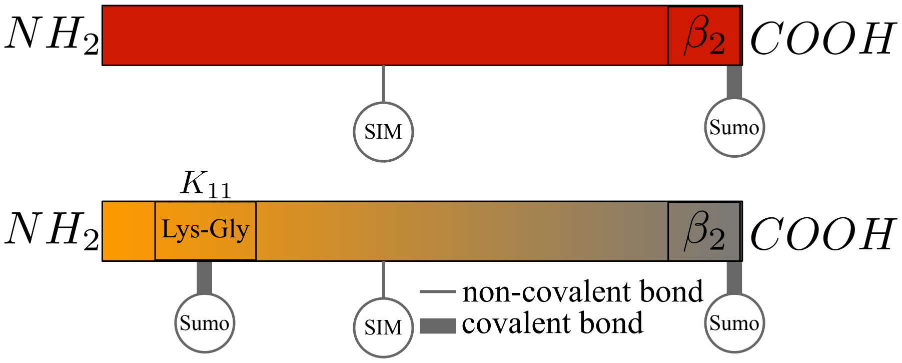


Figure 4.3: Schematic illustration of SUMO proteins. As the own SIM is a hypothesis, it is not entirely sure where exactly it is located. **Up:** Structure of a SUMO1 molecule, with its Lys-Gly residue at the C-Terminal and the β strand to attach at SIMs. Note that it is missing the K_{11} site and thus terminates SUMO chains. **Down:** Structure of SUMO2/3 protein, with its C-Terminal able to form covalent bonds at the β strand. Furthermore, it has a second Lys-Gly domain to bind other SUMO proteins able to form SUMO chains.

4.2 ANALYZING METHODS

Analyzing the simulation outcome of a model is a crucial step, as it provides evidence that the model is mimicking the desired behavior correctly. Usually, ODE models are easily evaluated by analyzing the concentration-over-time plot, as done in the previous section (cf. Sec. 3.1), because they allow quantitative as well as qualitative statements. Evaluating self-assembly simulations is more challenging, because 3D-structures need to be compared which is not easy, and often content sensitive. For this purpose the main structural criteria of PML bodies were purified and appropriate methods developed to determine them in the simulation:

(P1) Shell-like structure, with a nearly empty inside and a constant thickness

(P2) Non-homogeneous surface with patches, discrete holes and clustered proteins

4.2.1 *Relative Radial Density*

One method to validate if a structure has a shell is to analyze the spherical distribution of all molecules. Beginning with the center point of the structure proteins are clustered by their distance to the defined center, which then is displayed as a concentration-over-distance histogram. Note that the concentration is weighted with the distance,

This method assumes that the center of the structure is known. Here analyzed structures have their center in the origin of the reactor.

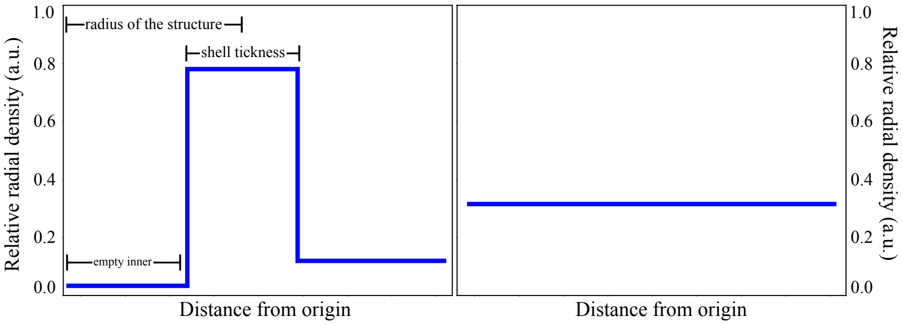


Figure 4.4: Shown are two prototypes for the outcome of the spherical distribution analysis. **Left:** If the structure shows a shell, the radial density should yield a single peak, corresponding to the thickness of the shell and the radius of the structure. In case of the PML bodies the inner part of the structure (distances smaller than the radius of the structure) should be nearly empty. **Right:** Uniformly distributed particles yield a constant line, as the relative density is equal over the whole volume.

as the area in which a molecule is present increases with an increasing distance from the center. If this plot (cf. Fig. 4.4) shows one significant peak it is good evidence that at this distance relatively more molecules are gathered, which is exactly the case for a shell-like structure. This method is used to verify property (P1).

4.2.2 *Radial Distribution Function*

Property (P2) is fulfilled if the particles are clustered, which can be validated with the radial distribution function (RDF). This method is a common statistical analysis that compares the frequency of distances between a set of particles with the one in an ideal gas. In fact, the histogram of pairwise particle distances is normalized with the homogeneous particle density in an ideal gas $\rho = [\text{Amount of particles}]/[\text{Volume}]$ (cf. Fig. 4.5). If the particles form a chain-like structure, the RDF shows discrete spikes, with zero-lines in between them, whereby the distance between two maxima correspond to the distance between two particles. If the RDF graph contains a single dominant spike with a continuous tail (not only discrete distances) it provides evidence that the majority of the proteins is clustered. The RDF of an ideal gas shows a constant line, indicating the even distribution of all particles.

4.2.3 *Virtual Microscopy*

One novel visual method to obtain both properties (P1) and (P2) is virtual microscopy, developed in the scope of this thesis and publicly

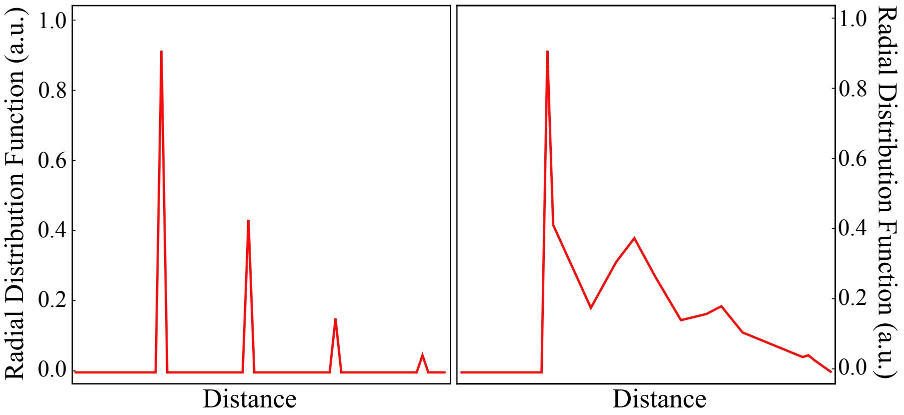


Figure 4.5: Shown are two prototypes for the outcome of the radial distribution function. **Left:** If the radial distribution function shows discrete spikes, the most likely structure is a chain polymer, where the distance to the regarding spikes correlates with multiples of the monomer's diameter (which usually is the interaction distance). **Right:** If the structure is clustered, the RDF yields one prominent peak and a continuous tail. Next to the direct neighborhood, most distances occur through the clustering of molecules.

available on GitHub [Hen18b]. Visualizing structures *in-vitro* or *in-vivo* is done by tagging certain molecules with a fluorophore and observing them under the microscope. This method is transferred to *in-silico* models by virtual microscopy. The outline of the algorithm is given below.

Also multiple proteins can be visualized at the same time, which is challenging in wet-lab experiments.

(1) **Input:** Particle trajectory with (x, y, z) positions of all particles; size of the reactor; number of particles.

(2) **Initialize:** Set: type IDs and radius of particles of interest, time-step of the record, height of the layer, width of the layer, angle of the layer (default $= 0°$), resolution of output image.

(3) **Separate Time-Step:** Isolate the particles position during the set time-step.

(4) **Iterate over Particles:** Check whether particle lies within the chosen layer and bin it into the two-dimensional output image.

(5) **Combine Pixels:** Calculate the brightness of every pixel of the output image, by combining the amount of particles linearly.

(6) **Visualize:** Visualize the generated image.

In conclusion, the input of the algorithm is a particle trajectory and a freely chosen layer and time-step, output is a weighted image of the tagged particles within the layer. This resulting image provides a visual validation of the model, as its is easy to compare to microscope-generated images.

4.3 MODEL

4.3.1 *Interaction Framework of PML Nuclear Bodies*

Using all PML body contributors as building blocks, their interacting motifs were used to establish an interaction framework. Those interactions between all proteins are determined by their structure, namely their known domains and motifs as described in Section 4.1.2. In the active state of the PML life cycle, the seed of the self-assembling bodies are the PML proteins, as they have three covalent binding sites (K_{65}, K_{160} and K_{490}) that are the base for the formation of SUMO chains.

PML	(K65, K160, K490, SIM, bound ~ yes)
DAXX	(K630, SIM1, SIM2, bound ~ yes)
Sp100	(K297, SIM, bound ~ yes)
SUMO1	(C, beta, bound ~ no)
SUMO23	(C, K11, beta, bound ~ no)

Any of these three covalent binding sites can bind any of the three SUMO variations covalently at their C-terminus. Once attached, SUMO2 and SUMO3 can covalently attach any other SUMO on their K_{11} site, resulting eventually in a chain. If SUMO1 attaches to PML or an existing SUMO2/3 chain, it terminates the chain, making it unable to bind other SUMOs as it is lacking K_{11} (cf. Sec. 4.1.2). Simulating this process can result in SUMO chains of infinite length.

Infinite, as a figure of speech, refers to the fact that the length of the chains is not determined a-priori.

The average length of SUMO chains can be estimated from Western blots and is known to contain in average $15-20$ SUMO molecules. As SUMOylation is an enzymatic, hence reversible process [Yeh09], the k_{on} and k_{off} values in the model can be chosen in a way that the average length of SUMO chains corresponds to the values measured in wet-lab measured (cf. Fig. C.2).

Note that SUMO chains not necessarily have to be found covalently, for which reason the theoretical calculation can differ from the practical one.

Sp100 and DAXX are somewhat similar to PML protein with respect to SUMOylation and SUMO binding via a SIM. While both proteins have not been reported to carry SUMO2/3 chains, their modification with SUMO1 is documented [See+01; JRK02]. Since PML,

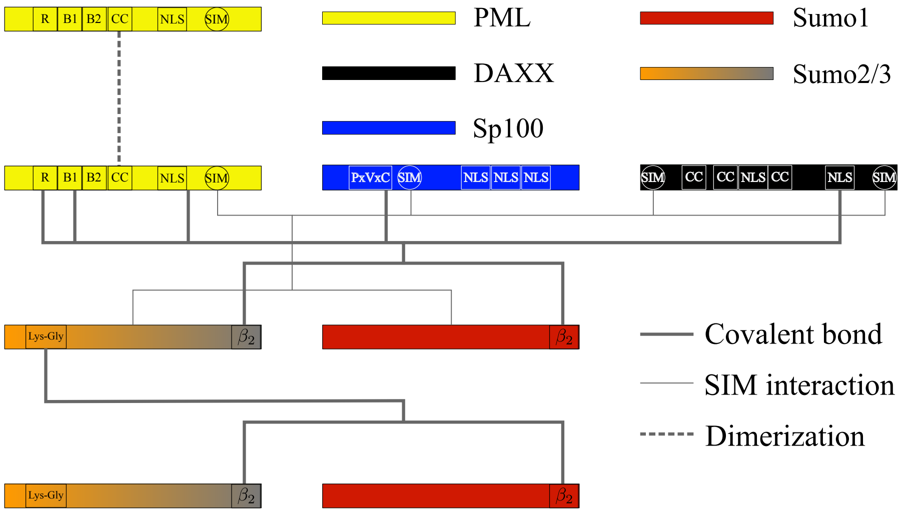


Figure 4.6: Shown is the interaction framework of the PML body components, labeled in the upper right corner. Lines denote between which binding sites bonds can established, following the molecular structure introduced in Section 4.1.2 and shown in Figures 4.1, 4.2 and 4.3. The kind of bond is labeled in the lower right corner, where those differences all are treated the same in the simulation.

DAXX and Sp100 also contain SIM motifs, they can bind to any moiety on the SUMO chains. This mechanism is the fundamental of PML nuclear bodies formation as it allows the occurrence of more complex structures and not only chains.

Additionally, all three seed species (PML, Sp100 and DAXX) have SIMs (cf. Sec. 4.1.2). Any part of a SUMO chain (including the end of non-terminated ones) can bind to the SIM of those species. The whole framework to describe the formation of nuclear bodies is depicted in Figure 4.6.

During the resting state of the PML bodies, additional reactions are the direct binding of PML proteins to each other. In this state PML bodies do not have their proper functionality, as only PML proteins remain in the cell. Nevertheless, they cluster and form insoluble complexes that are locally fixed in the cell, which explains how nuclear dots are locally stable over the cell cycle. Later in mitosis SUMO, DAXX and Sp100 are added to form mature PML nuclear bodies. The resting state contains only one type of molecule, for what reason the dimerization of PML protein is omitted in the model and the following sections focus on the formation of the active bodies and only show the resting state in Figure 4.8A.

Table 4.1: **Spatial properties of PML molecules** The molecular mass is taken from protein databases, the radius and diffusion coefficient is calculated using the methods introduced in Section 2.1.2.

Species	Mass	Radius	Diffusion coefficient
PML	76.36kDa	2.80nm	$11.63um^2/s$
Sp100	60.00kDa	2.58nm	$12.62um^2/s$
DAXX	83.00kDa	2.88nm	$11.30um^2/s$
SUMO1	12.00kDa	1.42nm	$22.93um^2/s$
SUMO2	12.00kDa	1.42nm	$22.93um^2/s$
SUMO3	12.00kDa	1.42nm	$22.93um^2/s$

4.3.2 *Setting up the Model*

Approximating protein complexes as spheres is usually done for the sake of simplicity.

Current peak resolution: 2.4nm.

Translating the framework into a rule-based spatial model requires the definition of particles, representing PML species, and the formulation of rules between them, corresponding to the interacting motifs. As the focus of interest lies on the structure of the forming bodies, the geometry plays an essential role. For this purpose, all proteins have been modeled as homogeneous spheres, with their radii corresponding to their molecular mass (cf. Tab. 4.1), following the methods from Section 2.1.2. Next to these spatial specifications it is necessary to know the initial particle number of every species. It is known that only 6.5% of all PML proteins form nuclear bodies in the nucleus. Unfortunately, there is a major disagreement in the literature on the number of PML body components in one nucleus. Table 4.2 summarizes these different values. The most reliable number is the one for the amount of PML protein in the nucleus estimated by the Hemmerich lab, based on quantitative Western blotting measurements. Based on this anchor number it appears that the numbers calculated by Beck and Schwanhäuser are far too low for PML protein. Assigning this assumption for the numbers of the other proteins, the difference-factor is applied and results in the numbers listed in the row "Sim" in Table 4.2. The sum of all species would be around 10^6 which is too large to simulate in SRSim in a feasible amount of time. For that reason future simulations only run with 10% of the estimated particles. While the shell thickness is unaffected, the radius of the simulated body is $\approx \sqrt[3]{0.1}$ of the original value.

In contrast to the SAC model from Section 3.1, reaction rules are now bond forming rules, meaning that chemical transformations do

Table 4.2: **Particle numbers in different studies** Beck refers to the work from Beck et. al [Bec+11], Schw to the work from Schwanhäusser et al [Sch+11], Hemm to the findings of Hemmerich (*unpublished data)* and Sim are the values used in these simulations.

Study	Location	PML	Sp100	DAXX	SUMO1	SUMO2/3
Beck	Nucleus	6.2×10^4	3.1×10^4	2.6×10^4	3.5×10^6	1.4×10^7
	All bodies	4.0×10^3	2.0×10^3	1.7×10^3	2.3×10^4	9.0×10^5
	One body	2.1×10^2	1.0×10^2	0.9×10^2	1.2×10^3	4.7×10^4
Schw	Nucleus	9.5×10^4	-	-	5.1×10^5	6.8×10^6
	All bodies	6.2×10^3	-	-	3.3×10^4	4.4×10^5
	One body	3.3×10^2	-	-	1.7×10^3	2.3×10^4
Hemm	Nucleus	7.5×10^5	-	-	-	-
	All bodies	4.9×10^4	-	-	-	-
	One body	2.6×10^3	-	-	-	-
Sim	One body	2.6×10^3	1.3×10^3	1.1×10^3	1.3×10^4	1.8×10^5

not occur but only the agglomeration of particles. To define these rules between species, they must be equipped with binding sites that are located on the surface of the regarding particle. Binding sites allow the formation of bonds between molecules, following a set of predefined rules (cf. App. D.1). Here, all binding sites have been defined consistent with the motif configuration of the regarding molecule. Additionaly, a species "Repulsor" is introduced, which is unable to react and located on the surface of the reactive volume to generate a heterogeneous surface (this models the chromosomes, as PML nuclear bodies are located in the interchromosomales space).

PML	(Sumo1, Sumo2, Sumo3, SIM, bound ~ yes)
DAXX	(Sumo, SIM1, Sim2, bound ~ yes)
Sp100	(Sumo, SIM, bound ~ yes)
SUMO1	(Sumo, PML, bound ~ no)
SUMO2	(Sumo1, Sumo2, PML, bound ~ no)
SUMO3	(Sumo1, Sumo2, PML, bound ~ no)
Repulsor	(bound ~ no)

The brackets of each protein denote its binding sites as used in the .bngl file, whereby the "bound" property is a Boolean value, indicating if the individual protein is already part of a complex. Initially only PML, Sp100 and DAXX can form bonds (rate k_3), once a SUMO is attached its "bound" flag is set to "yes" (rate k_4), meaning that SUMO only can bind another SUMO if either of them is already part of a complex. Both of these reactions are reversible with rate k_5 and k_6, respectively. The model uses six reaction rates, which are listed below and chosen randomly, as no kinetic data is available.

This hinders the free formation of SUMO chains in cytosol.

k_1	1.5×10^{-2}
k_2	1.0×10^{0}
k_3	1.5×10^{-2}
k_4	1.5×10^{0}
k_5	1.5×10^{-7}
k_6	1.5×10^{-2}

These rates are the micro-rates, corresponding to a reaction probability (cf. Sec.2.1.3). Rates k_1 and k_2 are not used in this model, but describe the PML homo-dimerization.

To avoid the bulking of SUMO chains, an angular force is applied between three individual SUMO molecules in a chain. The angle between three SUMO particles is always kept at a minimum of 100° giving the chains a certain flexibility (cf. App. D.1). Eventually, the cubical reactor is set up with reflective boundary conditions, so that its size roughly corresponds to the available space of one PML nuclear body in the nucleus. The files of the model description are shown in Appendix D.1 and attached in the digital supplement material.

4.4 SIMULATION RESULTS

With the derived building block's geometry and their interaction framework the aim of the following sections is to find requirements that lead to the self-assembly of PML nuclear bodies. Initially, all particles are distributed equally throughout the reactor. Besides the space-exclusion and the binding potential no forces are added. Later, the initial distribution is varied, as well as the force field between SUMO molecules.

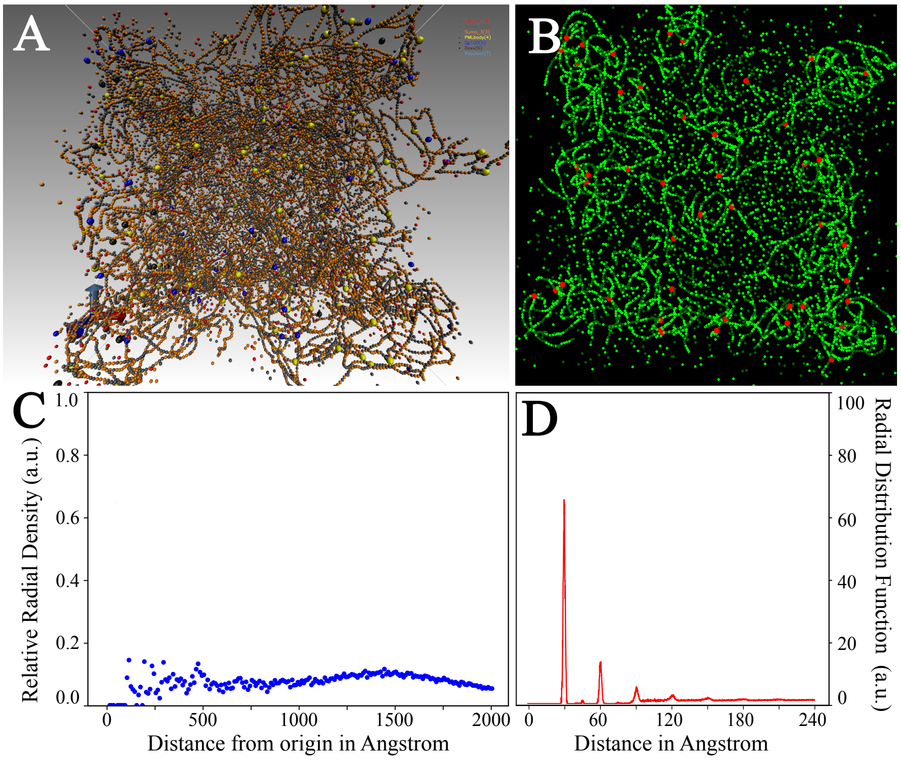


Figure 4.7: Results of the basic particle simulation of PML bodies. **A:** Shown is the 3D visualization of the final state, no PML body is formed, only a net of PML molecules and SUMO chains. **B:** Virtual microscopy shows the position of PML molecules (red) and SUMO2/3 (green) in an origin centered layer. Obviously, they are distributed homogeneously and do not form a shell. **C:** The molecule concentration over distance-from-origin plot shows clearly the even distribution and the lacking shell. **D:** The radial distribution function shows discrete peaks, indicating a kind of clustering. Nevertheless, the zero line between two peaks is evidence that the clustering is not like a bulk, but just a chain, which also can be seen in panels A and B, where the peaks are exactly at multiples of the radii of SUMO molecules.

4.4.1 *Basic Particle Simulation: Intuitive Approach*

The basic simulation simply spawns all particles randomly over the whole reactor, neglecting the fact that in human nuclei an insoluble PML bulk is present. It has all above described rules enabled and no further force-fields applied.

The outcome of the simulation is presented in Figure 4.7 and shows clearly that SUMO chains are formed between the seed particles as desired and form a loose net that covers the whole reaction vessel. As classified above the two properties of PML nuclear bodies are missing: firstly (P1), the nature of a shell with a nearly empty core, which is seen in the analysis of the radial distribution that does not show a well-defined peak, and secondly (P2), the net is not clustered, but fine-grained, which is seen by the RDF that shows the prototypic behavior of a chain structure (cf. Fig. 4.7D).

4.4.2 *Concentrating PML Molecules Initially: Seed PML Simulation*

This PML bulk is the resting state of PML bodies.

The following simulation incorporates the fact that PML proteins form an insoluble seed during M-Phase, consequently property (P1) can be observed. PML proteins are forming multimers via the described coiled-coil motif (CC, cf. Sec. 4.1.2). *In-silico* this is achieved by defining a sub-volume in the center of the reactor where all PML proteins spawn initially, without the inclusion of the regarding interaction. Panel A of Figure 4.8 shows what this seed could look like. Using this structure as the origin of the simulation mimics the effect that PML bodies change from their resting state to an active one, which

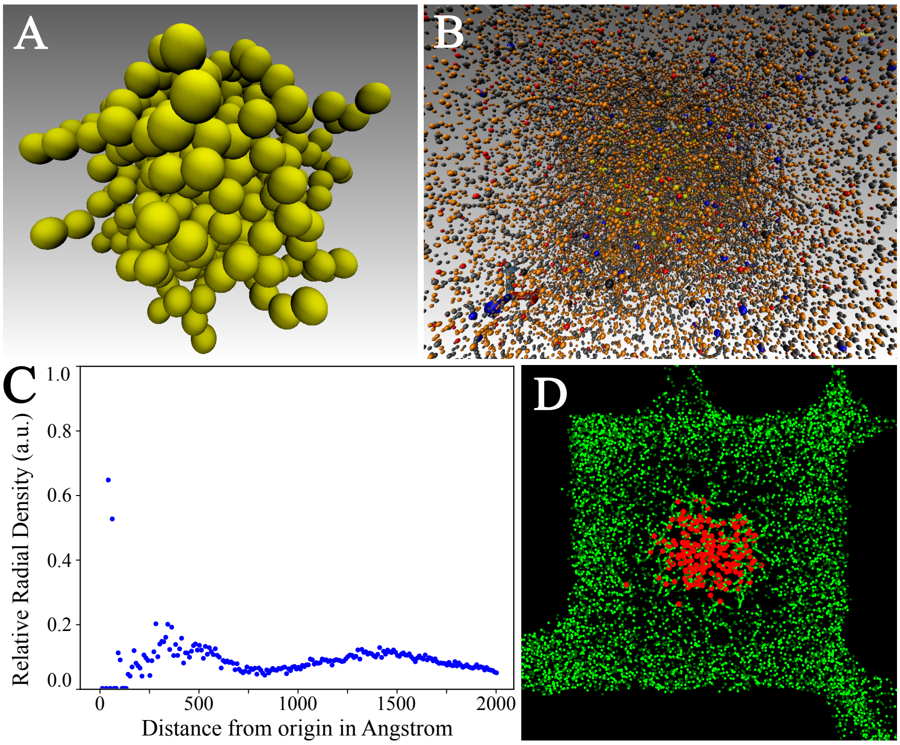

A
B
C
D
Relative Radial Density (a.u.)
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
Distance from origin in Angstrom

Figure 4.8: Results of the simulation with an initial bulk consisting of PML molecules. **A:** Shown is the 3D visualization of the PML bulk, representing the resting state of PML bodies and origin for the further simulation. **B:** 3D image of the final simulation state shows that the structure is more centered but not clustered. **C:** The molecule concentration over distance-from-origin plot shows the slight formation of some peaks, indicating a weak shell at a radius of 300 – 400 nm. **D:** Virtual microscopy shows the position of PML molecules (red) and SUMO2/3 (green) in an origin centered layer. The initial clustering of PML molecules at the center is dissolved by SUMO, while later ones are still distributed equally over the whole surface.

is achieved through the addition of the remaining proteins, namely Sp100, DAXX and the SUMO variations.

Figure 4.8 shows that with this assumption (altered initial distribution) fewer chains are formed which might be due to the uneven distribution of seed particles and SUMO variations. Nevertheless, the radial distribution now shows a slight peak, indicating the typical shell structure of PML nuclear bodies starts to form (property (P1)). The outcome of the RDF is not shown as it did not change, compared to the Basic simulation, because the formed network between PML, Sp100 and DAXX particles is still not clustered and looks quite fragile, as seen in the virtual microscopy image.

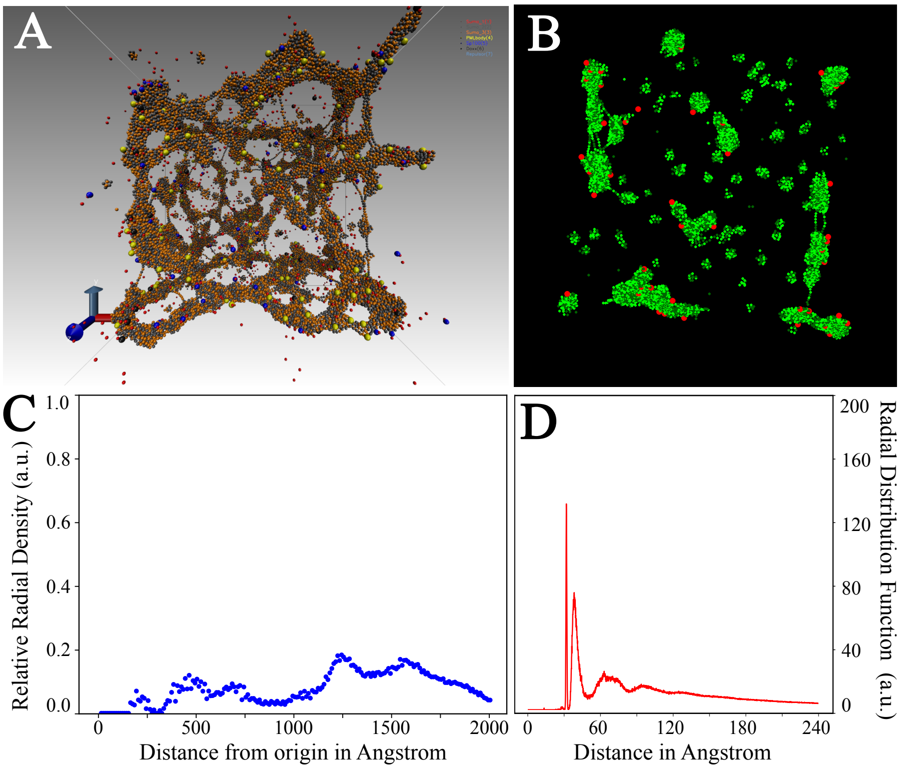


Figure 4.9: Outcome of the simulation using no PML seed but a SUMO-SUMO potential. **A:** Shown is the 3D visualization of the final state, SUMOchains form and generate clustered strands, resulting in one discrete structure. **B:** Virtual microscopy shows the position of PML molecules (red) and SUMO2/3 (green) in an origin centered layer. The clustering is visually recognizable, but spread over the surface as PML is not bulk seeded initially. **C:** The molecule concentration over distance-from-origin plot shows an accumulation of molecules at ~ 1,250 nm, which could be due to the large structure that expands into the corners of the reactor. **D:** The radial distribution function shows discrete peaks, indicating a kind of clustering. Here, the peak is more marked, compared to the one in Figure 4.7D and also does not show zero lines, indicating that the molecules are not only clustered in a chain, but in a bulk.

4.4.3 *Adding SUMO-SUMO Interaction: Attraction of SUMO Chains*

The preliminary formed bodies are not clustered as expected. Their surface is more like a net of individual SUMO chains, rather than forming a shell with discrete holes as seen in the RDF 4.7D.

SUMOs are slightly polar, suggesting the possibility that chains enhance this property. Huang et. al found that based on this polarity SUMO chains of opposing orientation attract each other [Hua+04]. These potentials only occur once the chain has a certain length. Experimental findings of the Hemmerich lab support this hypothesis *(unpublished data)*.

Incorporating this behavior into the model is done via a Leonard-Jones-like potential, namely the Leonard-Jones(2,1)-Potential:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^2 - \left(\frac{\sigma}{r} \right) \right] \tag{4.1}$$

with ϵ and σ being the regular LJ parameters and r the distance between two SUMO particles. Activating the potential once the chains reached a certain length is rather difficult to implement and basically done time-based. After a sufficient time (until most possible bonds are formed) it is assumed that SUMO chains reached a certain length and the potential is then activated. Here, the simulation runs for $1,000,000$ time-steps, which is sufficient to form most bonds, then the Leonard-Jones(2,1) is activated and the simulation runs for another $3,000,000$ time-steps.

Note that PML protein is not seeded any more in this case.

With the inclusion of this SUMO potential the outcome looks already more like a discrete structure, as huge clusters are formed that are connected with thicker strands (P2). Nevertheless, the entity looks rather porous and not very compact and does not form a shell (P1), which can be seen in Figure 4.9C. Additionally, the "center of the body" is not well-defined and still contains particles.

4.4.4 *Taking All Parts Together: Realistic Simulation*

The key to getting realistic bodies that fulfill property (P1) and (P2) is the combination of the two previously tested approaches: seeding PML in a small sub-volume and the attraction of SUMO chains after a certain time.

Simulating this setup results in the formation of proper PML nuclear bodies, presented in Figure 4.10. Only one discrete structure is formed, containing most of the PML molecules and mostly empty in the inside. A thick-walled shell is formed that has some discrete holes in it but is not porous. Note that the thickness of the shell was found to be similar for all PML bodies, independent of their size, and for that reason the shell in the simulation has already a realistic thickness (cf. Fig. 4.10C). The clustering, represented by the RDF, does not change compared to the one in Figure 4.9 and thus is not shown here. Comparing the image of the virtual microscopy with the wet-lab image shows the similarities, as most particles are bound in one body, while there exist some small, mostly SUMO protein, patches (cf. Fig. 4.10B and 4.10D). In conclusion, with this model approach it is possible to explain the requirements of the PML nuclear body as a

This is mentioned here, as only 10% of the actual particle numbers are simulated.

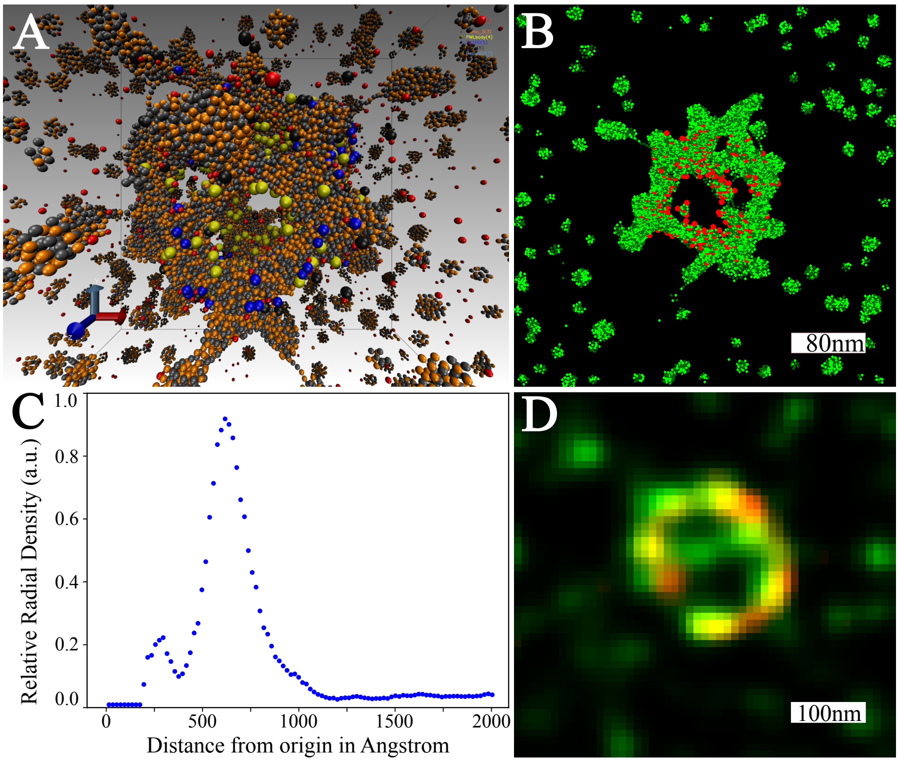


Figure 4.10: Outcome of the simulation with a SUMO-SUMO potential and a PML seed. **A:** 3D visualization of the final state, showing the PML body closely resembling nuclear body structure derived from super-resolution microscopy. **B:** Virtual microscopy shows the position of PML molecules (red) and SUMO2/3 (green) in an origin centered layer. It shows nicely the co-localization of the depicted proteins and some SUMO-patches outside the body, while the inside is nearly empty. The layer is centered with a thickness of ~ 50nm. **C:** The molecule concentration over distance-from-origin plot shows an accumulation of molecules at ~ 600 nm, which represent nicely the size of a PML nuclear body that only has 10% of the actual particles. The red curve presents the graph of a perfect sphere with a thickness, corresponding to the one from PML bodies. It shows remarkably that the simulated curve (blue) depicts the shell quite similar. **D:** STED super-resolution microscopy shows the position of PML molecules (red) and SUMO2/3 (green) in an origin-centered layer. It shows the co-localization of the depicted proteins and some patches outside of the body (Figure kindly provided by P. Hemmerich, FLI Jena).

self-assembly process. The next chapter will focus on the behavior of mutations to validate the model against the *in-vitro* behavior.

4.4.5 *Mutations*

After the visual comparison of *in-vivo* with *in-silico* studies, perturbation experiments were used to challenge the model and to validate model predictions. Mutations include depletion of all species individually down to 10% of their original amount, and the individual overexpression to different levels up to 2000%.

While those experiments are often challenging and time-consuming in wet-labs, their *in-silico* simulation is simple, as only the initial amount specific model members has to be changed. To make a qualitative statement about the effects of a modulation, three classifiers are introduced:

(**RNB**): Regular NB form, including a reliable shell thickness and diameter (P1) as well as the patchiness (P2).

(**NNB**): No proper NB is formed, including no clustering, or no discrete structure.

(**CNB**±): Changed NB are formed, including increased or decreased diameter, and different shell thickness, compared to the wild-type.

The above properties allow the classification of the nuclear bodies undergoing mutations, which is summarized in Table 4.3. The comparison shows clearly that the model behaves precisely as predicted by published wet-lab results (*unpublished data, Hemmerich lab*). Varying the amount of PML molecules influences directly the size of the body, where smaller bodies do not form a shell anymore but just a single aggregate. Changing the initial amount of SUMO1 should change the size of nuclear bodies as well, because SUMO1 is able to terminate SUMO chains. Lower or higher amounts of SUMO1 should therefore increase or, respectively, decrease the size of PML nuclear bodies, correlating with the length of SUMO chains.

Mechanistically, DAXX and Sp100 could have an influence on the formation of PML bodies, at least *in-silico*, because both carry SUMOylation sites and SIMs, by which they might be critically involved in modulating the molecular fine structure of the complex protein-protein interaction network within the nuclear bodies. However, in Sp100 or DAXX depleted cells, PML body formation is not affected, supporting the notion that they do not play a major role in the assembly process [Ish+99; CC03]. The depletion of SUMO2 or

Table 4.3: **Comparison of *in-vivo* and *in-silico* mutation experiments**

Mutated Molecule	Particle Mutation	Observations	
		In-Vivo	**In-Silico**
PML	1000%	CNB+	CNB+
	2000%	CNB++	CNB++
	10%	CNB–	CNB–
Sp100	1000%	RNB	RNB
	10%	RNB	RNB
DAXX	1000%	RNB	RNB
	10%	RNB	RNB
SUMO1	1000%	CNB-	CNB-
	10%	CNB+	CNB+
SUMO2	1000%	-	RNB
	10%	RNB	RNB
SUMO3	1000%	-	RNB
	10%	RNB	RNB
SUMO2/3	1000%	-	RNB
	10%	NNB	NNB

SUMO3 individually has no substantial effect, because they are proposed to be mutually replaceable, based on their $\sim$ 96% sequence identity. The overexpression of SUMO2 or SUMO3 could not be done *in-silico*, because the amount of particles would exceed the capacity of the software (the simulation of a 10—fold increased SUMO particle count takes multiple months and thus is unfit for a study). Qualitative curves of the mutation experiments can be found in the appendix (cf. App. C.3 - C.7).

4.4.6 *Artificial PML Proteins*

One interesting question that arises is the necessity for three covalent SUMO binding sites and the SIM of PML. Figuring out the minimal requirements for the formation of PML nuclear bodies is equally interesting and challenging. To tackle this problem the focus is on the structure of the PML proteins. PML as the essential building block of

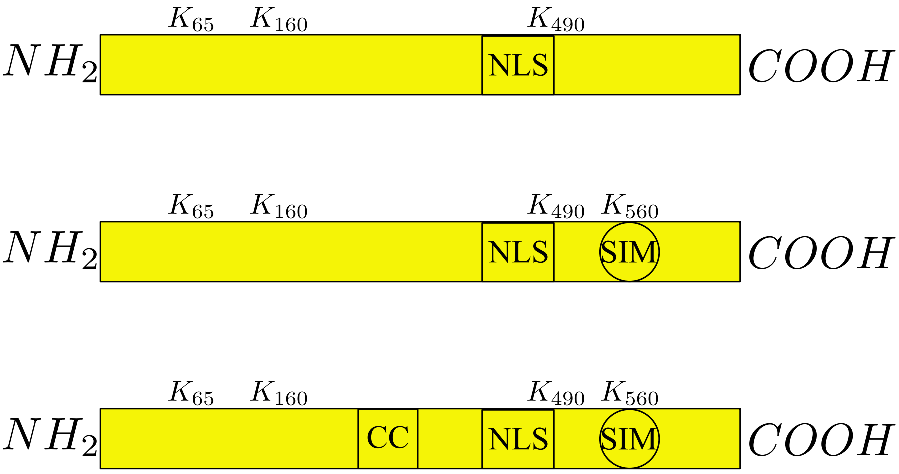


Figure 4.11: Depicted are the three artificial PMl molecules that were designed and tested by the Hemmerich lab. The first variant only contains the three covalent binding sites K65, K160 and K490, further ones have an additional SIM and coiled coil.

the bodies has four interaction motifs, namely three covalent SUMO binding sites and one SIM. The aim of the investigation is to check the need for all four sites. For that purpose different combinations of binding sites and motifs have been created and analyzed.

This method can be seen as extrapolation of the problem, as little to no wet-lab studies exist.

The *in-silico* realization again is quickly done by removing individual motifs in the definition file of the species. *In-vivo*, specific constructs need to be designed, which also has been done by the Hemmerich lab. The variants tested here are shown in Figure 4.11. The first step was to check whether those artificial PML proteins bind to an existing nuclear body, and surprisingly it was found that none of the variants bind, as they were found equally distributed throughout the whole space. This could not be confirmed *in-silico*, as all binding sites have the same weight and no special functions. As long as one covalent site remains the artificial protein at least will bind to an existing body.

As suggested by Boddy et. al [Bod+97] a small region of the RING finger motif of PML is crucial for the formation of bodies and cannot be compensated by the other binding sites, which has been confirmed by the Hemmerich lab. This motif is not included in the model for what reason the formation of PML bodies works properly, as long as binding sites are available.

Two sites only allow the generation of chains; one binding site would not form anything significant.

These results are sobering as the necessity for multiple binding sites in PML protein could not be investigated, neither *in-silico* nor *in-vitro*. The box motif is a functionality that has more than a structural property, which currently is not captured by the here proposed model of the PML body formation.

4.5 CONCLUSION

PML nuclear bodies present an interesting system for the study of self-assembly processes that is difficult to describe by conventional modeling approaches. For that reason a spatial rule-based model is proposed that is solely based on the few interactions between protein motifs (cf. Fig. 4.6). Starting from a simple approach, structures formed according to the defined rules, but these did not reflect the structures of PML nuclear bodies as known from super-resolution and electron microscopy studies (cf. Fig. 4.7). This indicates that the initial assumptions were too weak and some mechanisms are missing.

Validation via the properties (P1) and (P2).

Using this observation as input for further cell culture experiments, the Hemmerich lab found that SUMO proteins may contain a SIM, responsible for a SUMO-SUMO oligomerization potential, which was then validated by them (*unpublished data, Hemmerich lab*). This finding, taken together with the previously made observation that PML molecules form insoluble MAPPs in mitosis (resting state), was added to the model as additional assumption. This led to a qualitative and quantitative PML nuclear body, the structure of which was remarkable similar to the ultrastructure of PML nuclear bodies in cell culture cells (cf. Fig. 4.10). Methods to analyze the *in-silico* structure were developed and the new observations were compared with microscopic data. While the shell thickness is nearly identical, the diameter of the bodies was decreased, which results from the software restriction that allowed only the simulation of 10% of the actual amount of particles. Once the wild-type structure was generated *in-silico*, further validation experiments were performed through modulating the initial number of molecules. With respect to structural details the similarity between the model predictions and the living system was again remarkable (cf. Tab. 4.3). These observations suggest that the initial literature knowledge on molecular interactions between PML nuclear bodies components (cf. Fig. 4.6) together with two additional assumptions derived from the iterative (small-scale) bioinformatics approach, namely PML seed formation and SUMO chain dimerization, are minimally essential to assemble a PML nuclear body *in-silico*. This is consistent with a model in which mass-law action is sufficient to drive PML nuclear body assembly. Apparently, functional effects could not be observed as they are not included in the model, yet.

Finally, to further reduce the requirements for the formation of nuclear bodies, artificial PML molecules were designed and added to the existing nuclear body. With this kind of experiments the model was stretched to its limit, as it could not reproduce the data from the wet-lab. While the model's binding sites work, independent of anything around it, reality is way more complicated in terms of dependencies with protein structures.

Nevertheless, the model is the first one that explains the spatial self-assembly process of a PML nuclear bodies at molecular detail. Structural properties are well mimicked but it lacks functionality, meaning that molecules are only spatial building blocks. With the increasing calculation power and the rise of novel simulation method, it should be the aim of future approaches to increase the particle count to be biologically more relevant, and to refine the model by including binding site dependencies and adding structural functionality. Binding dependencies can already be included in the SRSim software, but usually lack wet-lab investigations; realizing a function, based on certain structural properties is a feature has to be included in the software itself.

5 CONCLUSION AND OUTLOOK

A procedure has been developed that translates ODE models to particle based models, which has been successfully applied to biological systems. Two novel models have been proposed that explain the switching nature of the SAC and the formation of PML nuclear bodies during cell cycle stages. While the SAC model could verify multiple existing hypotheses and validate wet-lab measured interaction rates, the model of PML bodies highlighted two assumptions that could later be verified in wet-lab experiments.

This thesis focuses on the study through simulation of complex systems. While a range of approaches and methods has emerged during past decades, most of them treat species as concentrations, which is easier to simulate and often sufficient. Nevertheless, usually space is omitted but it has a significant role in self-assembling processes and plenty of other biological processes such as the SAC.

See Section 2

Here a pathway is described that takes an ODE model as input and generates a corresponding particle model, based on spatial features that need to be added. However, usual biological systems contain billions of particles, and have to be simulated over a long time period compared to the possible time-steps. These different time-scales demand methodology that combines them in the way that the time it takes to simulate them is decreased, while the modeled system still mimics the original one. In Section 2 automatable coarse-grainings have been developed that reduce the amount of particles and dilate the system, so it becomes realizable within a feasible amount of time, but still delivers qualitatively and quantitatively correct results. These coarse-grainings have been applied successfully in Section 3.1 to simulate and investigate the spindle assembly checkpoint. A novel approach was realized in the DiCoSAD software package that combines spatial rule-based modeling with an event driven algorithm, allowing the study of self-assembling processes on a large time-scale. DiCoSAD unites the benefits of SRSim and GFRD and already demonstrated impressively the value of this approach, whereby the software still has to be extended by multiple features, e.g entire rule-system, unimolecular reactions, different boundaries, membranes.

See Section 3.1

Refining existing SAC models has led to a biochemically correct, spatial model that explains the natural switch from inhibited to activated APC/C, solely based on the amount of attached kinetochores. Including Securin and CyclinB in the model verified currently proposed hypotheses [Col+13; DG13] that the switch does not toggle but degrades steadily. The wet-lab measurements to accelerate or delay the switch can be shown precisely with the model. While automatable coarse-grainings from Section 2 were applied to generate a feasible particle simulation, showing realistic appearances of events, the model was manually reduced to its essential core. Performing a one-parameter bifurcation analysis on the minimal SAC model revealed the most crucial rate for a proper functioning switch. This rate precisely coincides with the one measured in wet-lab, demonstrating the strength and accuracy of the proposed model. Ultimately, an exact stochastic model was derived that underlined the hypothesis of a global SAC signal over the one of a local signal. Next step to enhance the SAC model would be to embed it into a compartmentalized system, like the one developed by [CL14]. While they focus on the microtubule attachment and Mad2 transport, biochemical correctness is neglected. Combining their approach with the one proposed in this thesis would lead to a complete whole cell level model that could provide an entire system understanding of the spindle assembly checkpoint.

See Section 4

PML nuclear bodies vary their structure during the cell cycle, making them an interesting self-assembling process. In section 4 the very first spatial model was developed, incorporating realistic particle relationships, that were simulated using a rule-based approach. The aim of the study was to find the necessary assumptions for the self-organized assembling of shell-like structures. Starting from an arbitrary particle distribution with no physical potentials led to some structural clusters, but did not match the specifications of nuclear bodies. The addition of an initial PML seed (corresponding to the resting state) and the attraction of SUMO chains, justified through their polarity, resulted in PML nuclear bodies that coincide precisely in wet-lab measured properties such as radius or shell thickness. Once the model was further validated by mutation experiments, that were all matching, artificial PML proteins were designed to investigate the minimal requirements for the assembling process. Here the model hit its limits, as it could not verify wet-lab results, because *in-silico* protein do not have a function yet. Future work should focus on the embedding of behavior into particles. This is partially done by the rule-based approach, but cannot cover the functionality required for the artificial PML bodies. Furthermore, the full particle account has to be realized, as in this thesis only 10% of the realistic value was used for performance issues.

In conclusion, this thesis has developed novel methodologies that help to provide a deeper insight into complex systems that have spatial features influencing their function. Furthermore, two models of concrete biological systems have been established, demonstrating how the previously introduced methods can be applied. Most importantly, the power of *in-silico* modeling has been shown, and hypotheses proposed that could be verified later *in-vivo*, thus embracing the interdisciplinary nature of theoretical biology.

A APPENDIX ALGORITHMS AND CALCULATIONS

A.1 DERIVING THE STEADY STATE OF A FUSION PROCESS ANALYTICALLY

Assuming the reversible reaction of two species A and B fusing into a third one C:

$$A + B \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} C. \tag{A.1}$$

The exact concentration of all species in the steady state (A_s, B_s, C_s) is of interest and can be calculated analytically. Considering this reaction irreversible leads to a trivial solution. The reaction is described with a set of ODEs under the asumption of mass-action-kinetics:

$A + B \rightarrow C \Rightarrow C_s = \min(A_0, B_0)$

$$\frac{d[A]}{dt} = -k_{on}[A][B] + k_{off}[C] \tag{A.2}$$

$$\frac{d[B]}{dt} = -k_{on}[A][B] + k_{off}[C] \tag{A.3}$$

$$\frac{d[C]}{dt} = k_{on}[A][B] - k_{off}[C]. \tag{A.4}$$

In steady state it is $\frac{d[A]}{dt} = \frac{d[B]}{dt} = \frac{d[C]}{dt} = 0$, which leads to the following relationship:

$$\frac{k_{on}}{k_{off}} = \frac{[C]}{[A][B]}. \tag{A.5}$$

With given initial concentrations A_0, B_0 and C_0 and reaction rates k_{on} and k_{off} the mass conservation during the reaction is described by the following equations:

$$[A] + [C] = A_0 + C_0 \tag{A.6}$$

$$[B] + [C] = B_0 + C_0. \tag{A.7}$$

These two conditions plugged into Equation A.5 results in a quadratic equation in [C]

$$[C]^2 - X[C] + Y = 0 \tag{A.8}$$

with

$$X = A_0 + B_0 + 2C_0 + \frac{k_{off}}{k_{on}} \tag{A.9}$$

$$Y = A_0B_0 + A_0C_0 + B_0C_0 + C_0^2 \tag{A.10}$$

which usually can be solved analytically and gives up to two solutions for C_s:

$$[C]_{s1/s2} = \frac{X}{2} \pm \sqrt{\left(\frac{X}{2}\right)^2 - Y}. \tag{A.11}$$

It is shown easily that both solutions are positive as all initial concentrations and reaction rates are positive. This states that theoretically two steady state concentrations of species C are possible, which is not possible in a one-dimensional deterministic system. With Equation A.6 and A.7, respectively, A and B can be determined with their initial concentration and C_s. Obviously, there are also two solutions for the concentration of A_s and B_s. It has to be shown that one of these solutions always is negative (it follows from Equation A.5 that the concentrations of A and B have the same sign), so there remains only one valid concentration for A, B and C.

Lemma 1. *Given Equations A.6, A.7 and A.11 there exists exactly one index* i *with* $[A]_i < 0 \wedge [B]_i < 0$.

Proof. A_s is defined as function of C_s:

$$\begin{aligned}
[A]_{s1/s2} &= A_0 + C_0 - C_{s1/s2} \\
&= A_0 + C_0 - \left(\frac{X}{2} \pm \sqrt{\left(\frac{X}{2}\right)^2 - Y}\right) \\
&= A_0 + C_0 - \frac{X}{2} \mp \sqrt{\left(\frac{X}{2}\right)^2 - Y} \\
&< A_0 + C_0 - \frac{X}{2} \mp \sqrt{\left(\frac{X}{2}\right)^2} \\
&= A_0 + C_0 - \frac{X}{2} \mp \frac{X}{2} \\
[A]_{s1} &< A_0 + C_0 - X = -B_0 - C_0 - \frac{k_{off}}{k_{on}} < 0 \\
[A]_{s2} &< A_0 + C_0
\end{aligned}$$

It is shown that $[A]_{s1}$ and thus $[B]_{s1}$ always is smaller than zero. □

With this proof the corresponding solution $[C]_{s1} = \frac{X}{2} + \sqrt{\left(\frac{X}{2}\right)^2 - Y}$ is not a valid one. The single solutions for the steady concentration of all species is:

$$[A]_S = A_0 + C_0 - \frac{X}{2} + \sqrt{\frac{X^2}{4} - Y} \tag{A.12}$$

$$[B]_S = B_0 + C_0 - \frac{X}{2} + \sqrt{\frac{X^2}{4} - Y} \tag{A.13}$$

$$[C]_S = \frac{X}{2} - \sqrt{\frac{X^2}{4} - Y} \tag{A.14}$$

which, extended and simplified results in:

$$[A]_S = \frac{1}{2}\left[A_0 - B_0 - \frac{k_{off}}{k_{on}} \right. \tag{A.15}$$

$$\left. + \frac{\sqrt{(A_0 k_{on} - B_0 k_{on} - k_{off})^2 + 4k_{on}k_{off}(A_0 + C_0)}}{k_{on}}\right] \tag{A.16}$$

$$[B]_S = \frac{1}{2}\left[-A_0 + B_0 - \frac{k_{off}}{k_{on}} \right. \tag{A.17}$$

$$\left. + \frac{\sqrt{(A_0 k_{on} - B_0 k_{on} - k_{off})^2 + 4k_{on}k_{off}(A_0 + C_0)}}{k_{on}}\right] \tag{A.18}$$

$$[C]_S = \frac{1}{2}\left[A_0 + B_0 + 2C_0 + \frac{k_{off}}{k_{on}} \right. \tag{A.19}$$

$$\left. - \frac{\sqrt{(A_0 k_{on} - B_0 k_{on} - k_{off})^2 + 4k_{on}k_{off}(A_0 + C_0)}}{k_{on}}\right]. \tag{A.20}$$

A.2 TRANSFORMING MARKOV CHAINS TO MASTER EQUATIONS

General form of a time discrete Markov Chain is:

$$\Phi(t+1) = \mathbf{M}\Phi(t). \tag{A.21}$$

Master equations usually are time continuous and are described with the following ODE in matrix notation:

$$\frac{d\Phi}{dt} = \mathbf{X}\Phi. \tag{A.22}$$

The aim is now to find the relationship between matrices **M** and **X**. Discretization of Equation A.22 using an Euler scheme and $dt = 1$ gives:

$$\Phi(t+1) = \Phi(t) + \mathbf{X}\Phi(t). \tag{A.23}$$

Plugging this result into Equation A.21 results in the relation

J *denotes the matrix of ones.*

$$\mathbf{X} = \mathbf{M} - \mathbf{J} \tag{A.24}$$

A.3 PROOF THAT LINEAR SCALING IS LENGTH CONSERVING

Lemma 2. *Given an euclidean space that undergoes a linear scaling* $f(x)$*, meaning that all points in that space are mapped linear by factor* c_s*. Then one has that the distance between two points in the original system is scaled by the same factor in the scaled system.*

Proof. The euclidean distance between two points $P = (p_x, p_y, p_z)$ and $Q = (q_x, q_y, q_z)$ is given by:

$$d(P, Q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}. \tag{A.25}$$

The system undergoes a dilation by factor c_s, whereby each point is transformed according to the following mapping:

$$f(x, y, z) = \left(\frac{x}{c_s}, \frac{y}{c_s}, \frac{z}{c_s}\right). \tag{A.26}$$

The distance between points P and Q then is transformed to:

$$\begin{aligned} d(f(P), f(Q)) &= \sqrt{\left(\tfrac{p_x}{c_s} - \tfrac{q_x}{c_s}\right)^2 + \left(\tfrac{p_y}{c_s} - \tfrac{q_y}{c_s}\right)^2 + \left(\tfrac{p_z}{c_s} - \tfrac{q_z}{c_s}\right)^2} \\ &= \tfrac{1}{c_s}\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \\ &= \tfrac{1}{c_s} d(P, Q). \end{aligned} \tag{A.27}$$

□

A.4 THEORETICAL CALCULATION OF MAD2 TURNOVER

The aim is to calculate the theoretical $k_o n$ value of the reaction

$$\text{O-Mad2} \xrightarrow{k_{on} \times [KinU]} \text{C-Mad2} \tag{A.28}$$

based on spatial properties and the assumption that O-Mad2 particles diffuse freely with coefficient D, unattached kinetochore have a fixed position, and both interact on contact.

This is achieved by calculating the average time $\langle T \rangle$ that a particle needs in order to enter a sub-volume (kinetochore, radius r) in the reaction vessel (nucleus, radius R). Applying a spherical Taylor expansion to an average hitting time model, and applying Neumann boundary conditions, leads to the following formula:

$$\langle T \rangle \approx \frac{R^3}{3Dr}. \tag{A.29}$$

Equation A.29 determines the average hitting time of a single particle and and thus correlates with the half-time of the reaction ($\langle T \rangle = t_{1/2}$):

$$t_{1/2} = \frac{\ln(2)}{k_{on}} \tag{A.30}$$

Assuming 6μm as the radius of the nucleus, a kinetochore radius of $0.1\mu m$ and the Mad2 diffusion coefficient to be $16.61\mu m^2 s^{-1}$, a half-time of 43 s is found, which is consistent with experimental findings [How+00] and results in a turnover rate of $k_{on} = 0.016s^{-1}$ at each kinetochore.

A.5 APPLYING SIMULATION COARSE GRAINING METHODS TO THE SAC MODEL

Realizing the simulation of 20 real-life minutes requires the scaling of space and time as discussed in Section 2. Here these methods are applied to simulate the developed SAC models. One of the two dilation parameters can be chosen freely, whereby the spatial one is limited, as very small particles are effected massively by strong forces. Thus, space was reduced by a factor of $c_s = 150$, resulting in an appropriate time scaling of $150^2 = 22,500 = c_t$, meaning 0.08 seconds of simulation time:

$$\frac{1800}{150^2} = 0.08.$$

Following the procedure, all rates are increased by the time-factor c_t, while all particles and the reactor are decreased by the space-factor c_s (cf. Tab. B.3).

B

APPENDIX TABLES

Table B.1: **Kinetic Parameters of the full SAC model**

Classification	Parameter	Value	Remark
Attachment	k_{attach}	$0.0032 s^{-1}$	20min metaphase
SAC activation	k_2	$100 \mu M^{-1} s^{-1}$	[Ibr+08a]
	k_{-2}	$0.08 s^{-1}$	[Ibr+08a]
	k_3	$0.02 s^{-1}$	[How+00]
	k_{-3}	$0.2 s^{-1}$	[How+00]
	k_4	$10.0 \mu M^{-1} s^{-1}$	[Ibr+08b]
	k_{-4}	$0.01 s^{-1}$	[Ibr+08b]
	k_5	$10.0 \mu M^{-1} s^{-1}$	[Ibr+09]
	k_{-5}	$0.01 s^{-1}$	[Ibr+09]
	k_6	$0.001 \mu M^{-1} s^{-1}$	[MS07]
	k_{-6}	$0.01 s^{-1}$	[Ibr+08b]
	k_8	$0.01 \mu M^{-1} s^{-1}$	[Ibr+09]
	k_{-8}	$0.1 s^{-1}$	[Ibr+09]
	k_{T1}	$0.01 \mu M^{-1} s^{-1}$	[DSM05]
	k_{-T1}	$0.02 s^{-1}$	[DSM05]
	k_{T2}	$10 \mu M^{-1} s^{-1}$	[DSM05]
SAC silencing	k_1	$5 \mu M^{-1} s^{-1}$	[Ibr+08a]
	k_{-1}	$0.08 s^{-1}$	[Ibr+08a]
	k_7	$0.01 s^{-1}$	this study
	k_9	$0.01 s^{-1}$	this study
	k_{10}	$0.1 s^{-1}$	this study
	k_D	$0.05 s^{-1}$	this study

Table B.2: **Spatial Parameters of the SAC Model.** Kinetochores initial amount is 92 and they do not diffuse. Other species particles start from zero. Their mass and diffusion coefficient combine from the basic blocks.

Species	Mass	Diffusion	Initial Conc.	Source
O-Mad2	26.06kDa	$16.61\mu m^2 s^{-1}$	$0.15\mu M$	[Fan02; Tan+01; How+00]
C-Mad2	26.06kDa	$16.61\mu m^2 s^{-1}$	$0.02\mu M$	[Fan02; Tan+01; How+00]
Cdc20	54.72kDa	$12.97\mu m^2 s^{-1}$	$0.13\mu M$	[Fan02; Tan+01]
BubR1:Bub3	242.00kDa	$7.92\mu m^2 s^{-1}$	$0.22\mu M$	[Fan02; Tan+01]
APC/C	836.5kDa	$5.23\mu m^2 s^{-1}$	$0.09\mu M$	[Tan+01]

Table B.3: **Coarse Graining Values for Spatial SAC Models.**

Property	real Value	CG value
Timescale c_t	-	$22,500$
Duration of Metaphase	1800 seconds	0.08 seconds
Nanosecond time-steps	1.8×10^{11}	8.0×10^{7}
Rates	k_i	$c_t \times k_i$
Spacescale c_s	-	150
Radius of Nucleus	$10.00\mu m$	$66.6 \times 10^{-3}\mu m$
Radius of Kinetochore	$0.1\mu m$	$66.6 \times 10^{-5}\mu m$
Radii	r_i	r_i/c_s

Table B.4: **Properties and Roles of PML Isoforms.** All information taken from [Nis+13].

Isoform	Mass in Da	Radius	Diffusion	Role
PML I	97.55 kDa	3.04 nm	1.85 um^2/s	Stimulates myeloid cell differentiation
PML II	90.72 kDa	2.97 nm	1.03 um^2/s	Implicated in virus-induced PML NB disruption
PML III	70.37 kDa	2.72 nm	1.63 um^2/s	Controls centrosome duplication
PML IV	70.02 kDa	2.72 nm	1.04 um^2/s	Regulates apoptosis, senescence and DNA damage
PML V	67.47 kDa	2.69 nm	2.79 um^2/s	Forms NB and recruits Daxx and Sp100
PML VI	62.01 kDa	2.61 nm	1.49 um^2/s	Resists to As2O3 induced degradation
PML Average	76.36 kDa	2.80 nm	1.64 um^2/s	-

C

APPENDIX FIGURES

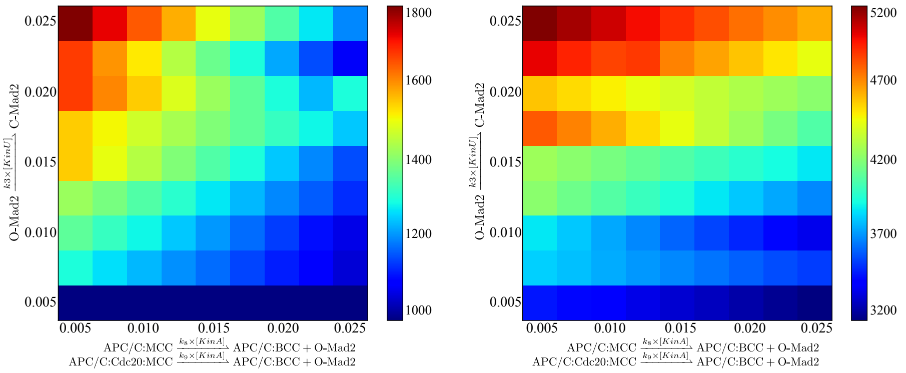


Figure C.1: Parameter study for the most crucial reaction rates, that are k_3 and k_8. The colorbars on the right of each panel present the half-time of Securin degradation in seconds. The x-axis varies the rate of the labeled reaction (k_3) while the y-axis alters reaction rate k_8. **Left:** Normal mitosis, where all kinetochores attach in an average time of $20 - 30$ minutes, corresponding to $\approx 1,500$ seconds. Looking at the presented heatmap this corresponds to values of $k_3 > 0.015$ and $k_8 < 0.015$. **Right:** Disturbed mitosis, where one chromosome is unable to attach. It shows that single kinetochores are able to extend metaphase for several hours (the shown values are only half-times of Securin). The rates for a reliable arrest in mitosis coincide with the ones in panel A. Both panels together suggest that k_3 value should be around 0.015 as it was theoretically determined to be a maximum of 0.016 (cf. App. A.4) and k_8 should not exceed 0.015. In other terms, both kinetochores have a strong influence to maintain the SAC as their reaction rates are similar.

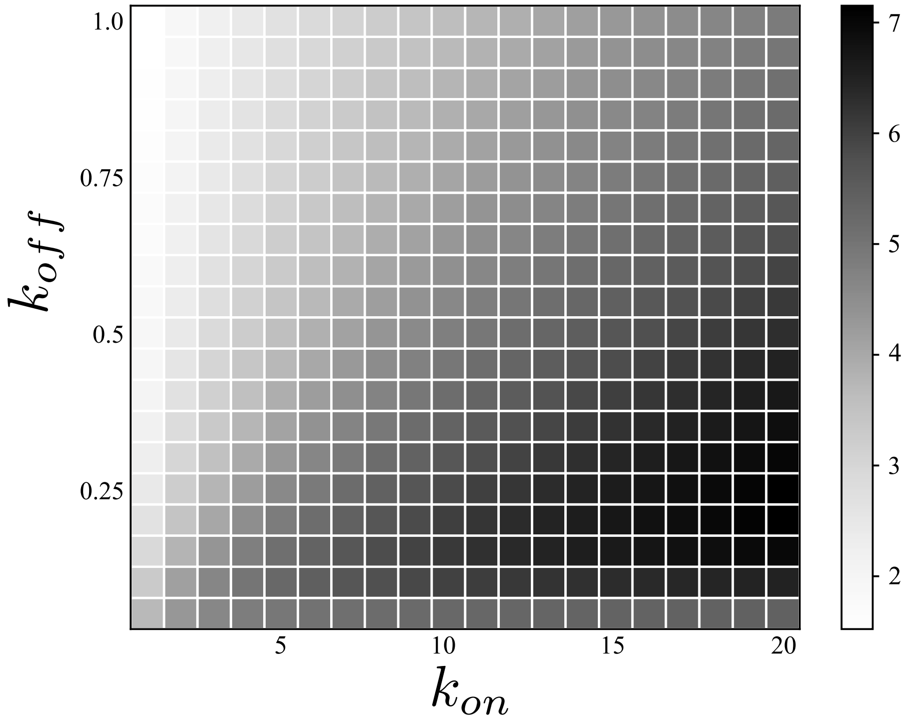


Figure C.2: Depicted is the heatmap of the average SUMO chain length at steady state, depending on varying k_{on} and k_{off} values. The x-axis corresponds to the k_{on} value, the y-axis to k_{off} of the SUMOylation. The 2D map shows color-coded (bar on the right) the average length of formed SUMO chains (in nm) at steady state. Western blots show PML protein based SUMO chains of $250 kDa$, corresponding to an amount of $10-15$ SUMO molecules. Reaching this length of a SUMO chain with mass-action kinetics is not reliable, but requires some enzyme kinetics. Another explanation is the SUMO - SUMO potential that could attract short length SUMO chains and thus form a non-covalent chain, which is not distinguishable in a western blot.

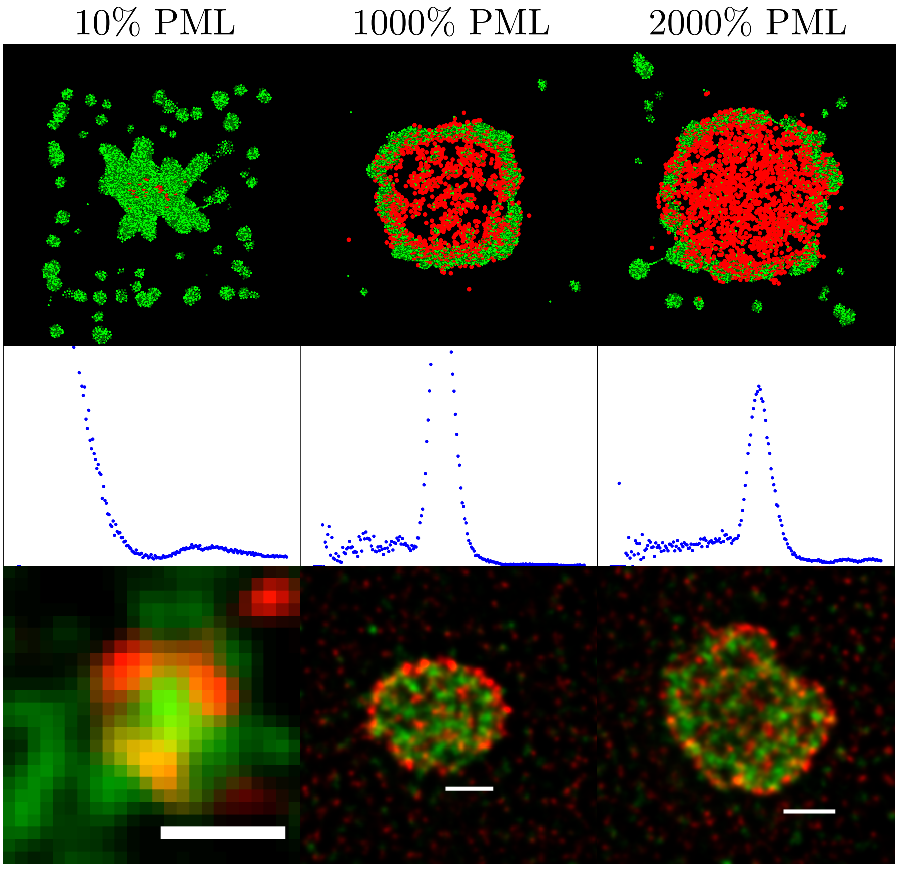


Figure C.3: Structure of PML nuclear bodies after modulation of PML molecule number. Comparison of the virtual microscopy and the STED super-resolution microscopy. Furthermore shown is the visualization of the spherical distribution of all molecules. **Left:** Reduction of initial PML molecule number to 10% of its actual value. The graph shows clearly that most of the proteins are clustered around the center, corresponding to a small PML nuclear body that does not develop a proper shell. The STED super-resolution image of a typical PML nuclear body after the knockdown is morphologically similar to the model nuclear body (100 nm bar). **Middle and Right:** 10/20-fold overexpression of PML leads to enlarged PML nuclear bodies, which clearly can be seen in the spherical distribution, as the main peak shift aways from the center. Typical PML nuclear bodies after the overexpression show some similar and dissimilar features when comparing the model result and the 'real' STED super-resolution images (500 nm bars). In the model, SUMO-2/3 is concentrated in the periphery of a more or less solid PML cloud. By STED, PML is also distributed in a large cloud but there is in addition some accumulation in the periphery, which might be due to the fact that the model overexpression is done locally restricted. SUMO-2/3 shares the core of these large PML bodies but does not strongly overlap with the PML core mesh. In addition, SUMO-2/3 does not form a shell-like structure as evident from the model.

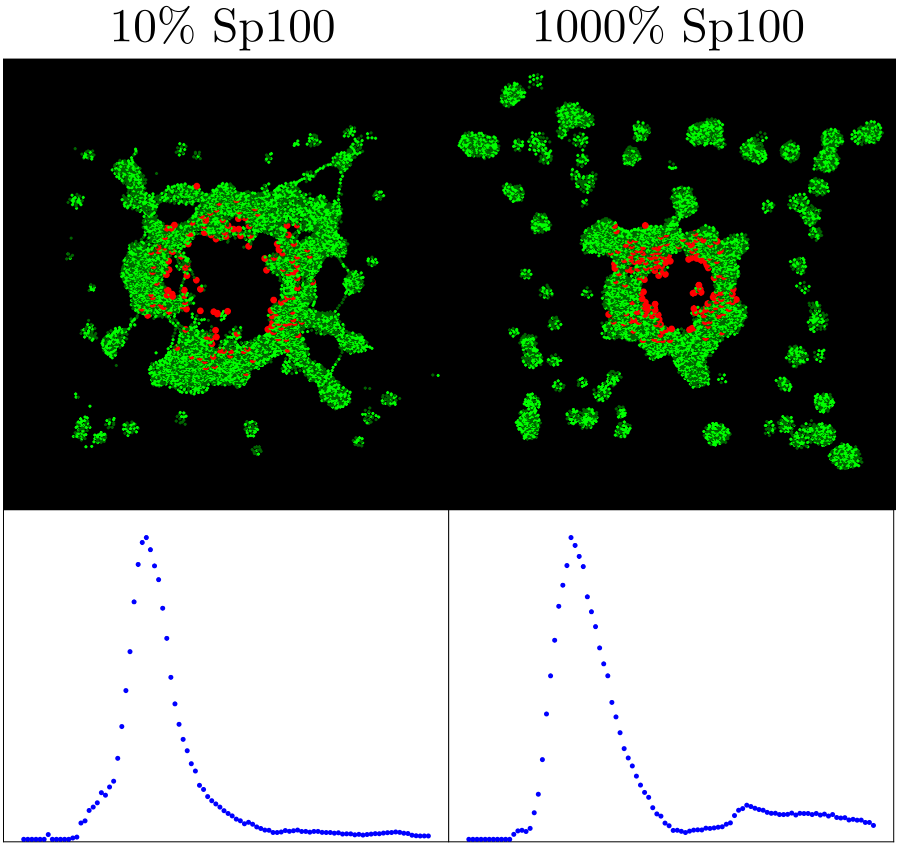


Figure C.4: Structure of PML nuclear bodies after modulation of Sp100 molecule number. Comparison of the virtual microscopy and the STED super-resolution microscopy, with 90% depleted (**Left**) and 10-fold overexpressed Sp100 (**Right**). Furthermore shown is the visualization of the spherical distribution of all molecules. As seen in the spherical distribution, Sp100 has a minor influence on the diameter of PML nuclear bodies, which might be due to the fact that only SUMO1 can bind, which terminates a chain and thus minimizes the structure. The *in-silico* model is consistent with published data showing no major influence of Sp100 on PML body structure (data not shown).

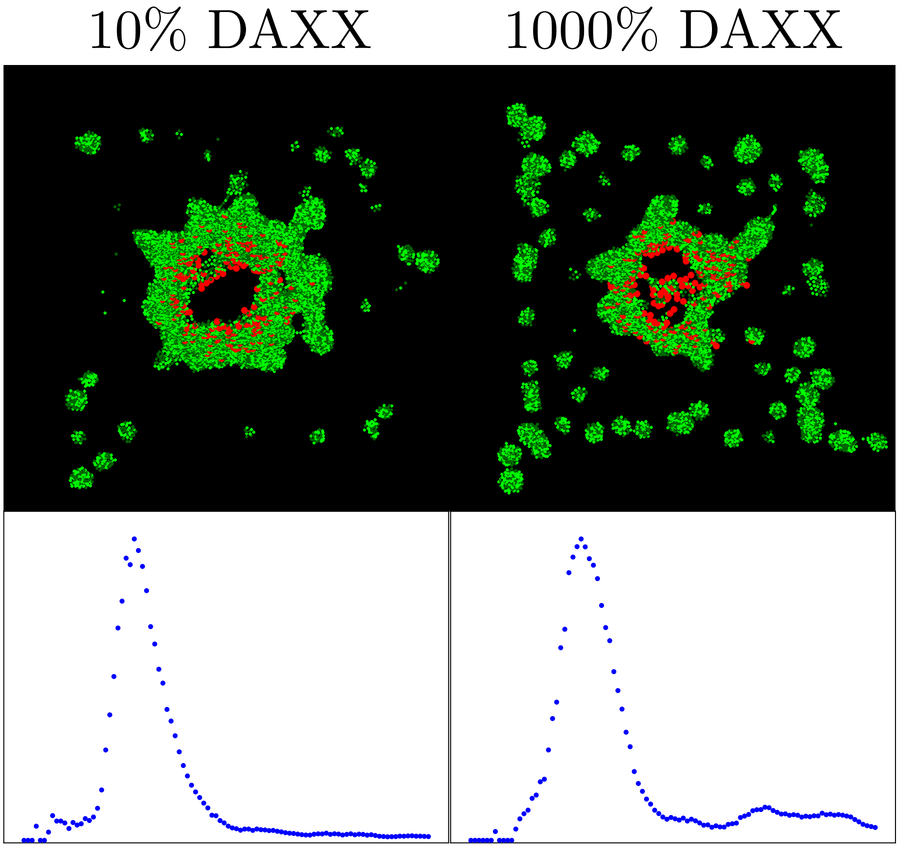


Figure C.5: Structure of PML nuclear bodies after modulation of DAXX molecule number. Comparison of the virtual microscopy and the STED super-resolution microscopy, with 90% depleted (**Left**) and 10-fold overexpressed DAXX (**Right**). Furthermore shown is the visualization of the spherical distribution of all molecules. As seen in the spherical distribution, DAXX has no great influence on the molecule density distribution of the PML nuclear body periphery. In the DAXX-depleted situation an accumulation of SUMO2/3 at the nuclear bodies can be observed, while DAXX overexpression does not alter the nuclear body structure. The *in-silico* model is consistent with published data showing no major influence of DAXX on PML body structure (data not shown).

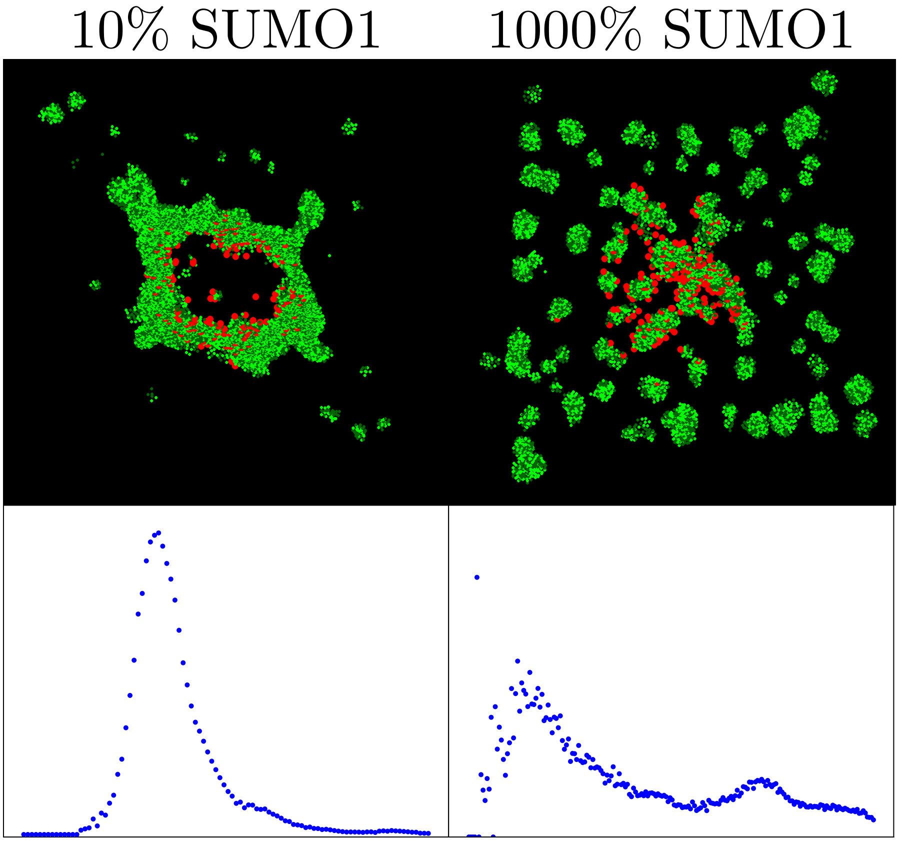


Figure C.6: Structure of PML nuclear bodes after modulation of SUMO1 molecule number. Comparison of 90% depleted (**Left**) with 10-fold overexpressed SUMO1 (**Right**). Removing SUMO1 results in enlarged chains, as SUMO1 terminates the SUMO chain formation, clustering most of the available material Without this feature, PML bodies increase in size as can be seen in the left column. Adding additional SUMO1 has the opposing effect, namely extremely shortened SUMO chains, which results in no formation of PML bodies, because the chain length is not sufficient to connect the PML proteins.

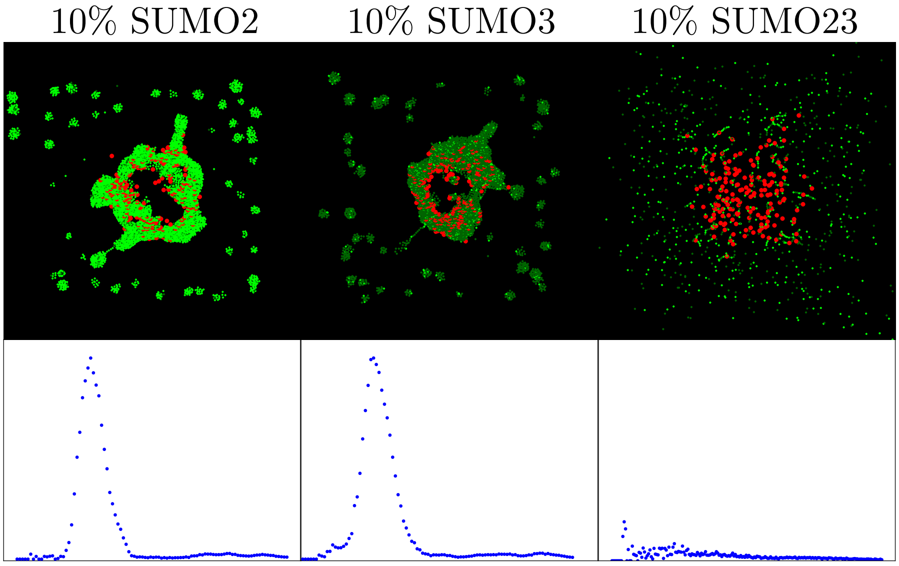


Figure C.7: Structure of PML nuclear bodies after modulation of SUMO2/3 molecule number. Comparison of 90% depleted SUMO2/3 individually (**Left** and **Middle**) and combined (**Right**). Removing only one of the two SUMO variants has nearly no effect on the formation of bodies, because SUMO2 and SUMO3 can replace each other. Neither the virtual microscopy, nor the spherical distribution show any anomalies compared to the wild-type. With both of them depleted simultaneously no structure is formed, as they are the main glue of the bodies.

D

APPENDIX SOURCE CODE OF THE SIMULATIONS

D.1 PML SIMULATION

The simulation of the PML bodies was performed using SRSim, which requires 4 input files. Essential part is the geometrical definition of all contributing particles, containing their name, mass and binding sites which is formulated in the .geo file. The same file furthermore holds general physical information, like force constants, and looks the following for the NB simulation:

Listing D.1: PMLBody.geo

```
<?xml version="1.0" ?>
<molecule-geometry-definition>
  <version value="1.01"/>
  <GeneralProperties>
    <property name="GPT_Devi_Dist" value="1.0"/>
    <property name="GPT_Devi_Angle" value="10"/>
    <property name="GPT_Mol_Mass" value="1"/>
    <property name="GPT_Mol_Rad" value="1"/>
    <property name="GPT_Site_Dist" value="1"/>
    <property name="GPT_Force_Repulsion" value="100.0"/>
    <property name="GPT_Force_Bond" value="1.0"/>
    <property name="GPT_Force_Angle" value="0"/>
    <property name="GPT_Force_Dihedral" value="300"/>
    <property name="GPT_Temperature" value="300"/>
    <property name="GPT_Refractory" value="50"/>
    <property name="GPT_Option_Dihedrals" value="0"/>
    <property name="GPT_Option_Impropers" value="0"/>
    <property name="GPT_Option_Rigid" value="1"/>
  </GeneralProperties>

  <molecule name="PMLbody">
    <property name="GPT_Mol_Mass" value="80.0"/>
    <property name="GPT_Mol_Rad" value="30.0"/>
    <property name="GPT_Force_Repulsion" value="2500.0"/>
    <property name="GPT_Force_Bond" value="10.0"/>
    <site dist="30.0" name="bound" phi="0" theta="0"/>
    <site dist="30.0" name="Sumo1" phi="0" theta="0"/>
    <site dist="30.0" name="Sumo2" phi="0" theta="90"/>
    <site dist="30.0" name="Sumo3" phi="135" theta="135"/>
    <site dist="30.0" name="Sim" phi="225" theta="135"/>
  </molecule>
```

```
<molecule name="Sumo_1">
  <property name="GPT_Mol_Mass" value="15"/>
  <property name="GPT_Mol_Rad" value="14.2"/>
  <property name="GPT_Devi_Angle" value="360"/>
  <property name="GPT_Force_Angle" value="100"/>
  <site dist="14.2" name="bound" phi="0" theta="0"/>
  <site dist="14.2" name="Sumo" phi="0" theta="90"/>
  <site dist="14.2" name="PML" phi="180" theta="90"/>
</molecule>

<molecule name="Sumo_2">
  <property name="GPT_Mol_Mass" value="15"/>
  <property name="GPT_Mol_Rad" value="14.2"/>
  <property name="GPT_Force_Angle" value="1.3"/>
  <property name="GPT_Devi_Angle" value="100"/>
  <site dist="14.2" name="bound" phi="0" theta="0"/>
  <site dist="14.2" name="Sumo_1" phi="0" theta="90"/>
  <site dist="14.2" name ="PML"   phi="90" theta="90"/>
  <site dist="14.2" name="Sumo_2" phi="180" theta="90"/>
</molecule>

<molecule name="Sumo_3">
  <property name="GPT_Mol_Mass" value="15"/>
  <property name="GPT_Mol_Rad" value="14.2"/>
  <property name="GPT_Force_Angle" value="1.3"/>
  <property name="GPT_Devi_Angle" value="100"/>
  <site dist="14.2" name="bound" phi="0" theta="0"/>
  <site dist="14.2" name="Sumo_1" phi="0" theta="90"/>
  <site dist="14.2" name ="PML"   phi="90" theta="90"/>
  <site dist="14.2" name="Sumo_2" phi="180" theta="90"/>
</molecule>

<molecule name="Sp100">
  <property name="GPT_Mol_Mass" value="60"/>
  <property name="GPT_Mol_Rad" value="25.8"/>
  <property name="GPT_Devi_Angle" value="360"/>
  <property name="GPT_Force_Angle" value="0"/>
  <site dist="25.8" name="bound" phi="0" theta="0"/>
  <site dist="25.8" name="Sim" phi="0" theta="90"/>
  <site dist="25.8" name="Sumo" phi="180" theta="90"/>
</molecule>

<molecule name="Daxx">
  <property name="GPT_Mol_Mass" value="83"/>
  <property name="GPT_Mol_Rad" value="28.8"/>
  <property name="GPT_Devi_Angle" value="360"/>
  <property name="GPT_Force_Angle" value="0"/>
  <site dist="28.8" name="bound" phi="0" theta="0"/>
```

```
    <site dist="28.8" name="Sim1" phi="0" theta="90"/>
    <site dist="28.8" name="Sim2" phi="180" theta="90"/>
    <site dist="28.8" name="Sumo" phi="90" theta="90"/>
  </molecule>

  <molecule name="Repulsor">
    <property name="GPT_Mol_Mass" value="300"/>
    <property name="GPT_Mol_Rad" value="200.0"/>
    <property name="GPT_Force_Repulsion" value="100.0"/>
    <property name="GPT_Refractory" value="50"/>
    <site dist="100.0" name="bound" phi="0" theta="0"/>
  </molecule>

</molecule-geometry-definition>
```

Another file contains the initial amount of all species, as well as their interactions formulated in the bngl language.

Listing D.2: PMLBody.bngl

```
begin species


     Sumo_1(Sumo,PML,bound~no)                  Sumo_10
     Sumo_2(Sumo_1,Sumo_2,PML,bound~no)         Sumo_20
     Sumo_3(Sumo_1,Sumo_2,PML,bound~no)         Sumo_30

     PMLbody(Sim,Sumo1,Sumo2,Sumo3,bound~yes)   PMLbody0
     Sp100(Sim,Sumo,bound~yes)                  Sp1000
     Daxx(Sim1,Sim2,Sumo,bound~yes)             Daxx0

     Repulsor(bound~yes)                        Repulsor0

end species

begin reaction rules

#PML Body Knauel
#1   PMLbody(Sumo) + PMLbody(Sumo1)     ->      PMLbody(Sumo!1).
    PMLbody(Sumo1!1)        k1
#2   PMLbody(Sumo) + PMLbody(Sumo2)     ->      PMLbody(Sumo!1).
    PMLbody(Sumo2!1)        k1
#3   PMLbody(Sumo) + PMLbody(Sumo3)     ->      PMLbody(Sumo!1).
    PMLbody(Sumo3!1)        k1

#4   PMLbody(Sumo!1,Sumo1!2).PMLbody(Sumo1!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo1).PMLbody(Sumo1!1,Sumo)     k2
#5   PMLbody(Sumo!1,Sumo1!2).PMLbody(Sumo2!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo1).PMLbody(Sumo2!1,Sumo)     k2
#6   PMLbody(Sumo!1,Sumo1!2).PMLbody(Sumo3!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo1).PMLbody(Sumo3!1,Sumo)     k2
```

```
#7   PMLbody(Sumo!1,Sumo2!2).PMLbody(Sumo1!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo2).PMLbody(Sumo1!1,Sumo)     k2
#8   PMLbody(Sumo!1,Sumo2!2).PMLbody(Sumo2!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo2).PMLbody(Sumo2!1,Sumo)     k2
#9   PMLbody(Sumo!1,Sumo2!2).PMLbody(Sumo3!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo2).PMLbody(Sumo3!1,Sumo)     k2

#10  PMLbody(Sumo!1,Sumo3!2).PMLbody(Sumo1!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo3).PMLbody(Sumo1!1,Sumo)     k2
#11  PMLbody(Sumo!1,Sumo3!2).PMLbody(Sumo2!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo3).PMLbody(Sumo2!1,Sumo)     k2
#12  PMLbody(Sumo!1,Sumo3!2).PMLbody(Sumo3!1,Sumo!2)    ->
    PMLbody(Sumo!1,Sumo3).PMLbody(Sumo3!1,Sumo)     k2

#Sumo1 an PML
13   Sumo_1(Sumo,bound~no) + PMLbody(Sumo1,bound~yes)   ->
    Sumo_1(Sumo!1,bound~no).PMLbody(Sumo1!1,bound~yes)      k3
14   Sumo_1(Sumo!1,bound~no).PMLbody(Sumo1!1,bound~yes)  ->
    Sumo_1(Sumo!1,bound~yes).PMLbody(Sumo1!1,bound~yes)        k4

15   Sumo_1(Sumo,bound~no) + PMLbody(Sumo2,bound~yes)   ->
    Sumo_1(Sumo!1,bound~no).PMLbody(Sumo2!1,bound~yes)      k3
16   Sumo_1(Sumo!1,bound~no).PMLbody(Sumo2!1,bound~yes)  ->
    Sumo_1(Sumo!1,bound~yes).PMLbody(Sumo2!1,bound~yes)        k4

17   Sumo_1(Sumo,bound~no) + PMLbody(Sumo3,bound~yes)   ->
    Sumo_1(Sumo!1,bound~no).PMLbody(Sumo3!1,bound~yes)      k3
18   Sumo_1(Sumo!1,bound~no).PMLbody(Sumo3!1,bound~yes)  ->
    Sumo_1(Sumo!1,bound~yes).PMLbody(Sumo3!1,bound~yes)        k4

#Sumo2 an PML
19   Sumo_2(Sumo_2,bound~no) + PMLbody(Sumo1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo1!1,bound~yes)       k3
20   Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo1!1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~yes).PMLbody(Sumo1!1,bound~yes)    k4

21   Sumo_2(Sumo_2,bound~no) + PMLbody(Sumo2,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo2!1,bound~yes)       k3
22   Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo2!1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~yes).PMLbody(Sumo2!1,bound~yes)    k4

23   Sumo_2(Sumo_2,bound~no) + PMLbody(Sumo3,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo3!1,bound~yes)       k3
24   Sumo_2(Sumo_2!1,bound~no).PMLbody(Sumo3!1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~yes).PMLbody(Sumo3!1,bound~yes)    k4

#Sumo3 an PML
```

```
25   Sumo_3(Sumo_2,bound~no) + PMLbody(Sumo1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo1!1,bound~yes)       k3
26   Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo1!1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~yes).PMLbody(Sumo1!1,bound~yes)    k4

27   Sumo_3(Sumo_2,bound~no) + PMLbody(Sumo2,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo2!1,bound~yes)       k3
28   Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo2!1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~yes).PMLbody(Sumo2!1,bound~yes)    k4

29   Sumo_3(Sumo_2,bound~no) + PMLbody(Sumo3,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo3!1,bound~yes)       k3
30   Sumo_3(Sumo_2!1,bound~no).PMLbody(Sumo3!1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~yes).PMLbody(Sumo3!1,bound~yes)    k4

#Sumo123, Ende der Kette, an neues PML
31   Sumo_1(PML,bound~yes) + PMLbody(Sim,bound~yes)  ->  Sumo_1(
    PML!1,bound~yes).PMLbody(Sim!1,bound~yes)              k3
32   Sumo_2(Sumo_1,bound~yes) + PMLbody(Sim,bound~yes)  ->
    Sumo_2(Sumo_1!1,bound~yes).PMLbody(Sim!1,bound~yes) k3
33   Sumo_3(Sumo_1,bound~yes) + PMLbody(Sim,bound~yes)  ->
    Sumo_3(Sumo_1!1,bound~yes).PMLbody(Sim!1,bound~yes) k3

#Sumo untereinander
34   Sumo_2(Sumo_1,bound~yes) + Sumo_2(Sumo_2,bound~no)  ->
    Sumo_2(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~no)       k3
35   Sumo_2(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~no)  ->
    Sumo_2(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~yes)    k4

36   Sumo_2(Sumo_1,bound~yes) + Sumo_3(Sumo_2,bound~no)  ->
    Sumo_2(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~no)       k3
37   Sumo_2(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~no)  ->
    Sumo_2(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~yes)    k4

38   Sumo_3(Sumo_1,bound~yes) + Sumo_2(Sumo_2,bound~no)  ->
    Sumo_3(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~no)       k3
39   Sumo_3(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~no)  ->
    Sumo_3(Sumo_1!1,bound~yes).Sumo_2(Sumo_2!1,bound~yes)    k4

40   Sumo_3(Sumo_1,bound~yes) + Sumo_3(Sumo_2,bound~no) ->
    Sumo_3(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~no)
    k3
41   Sumo_3(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~no)  ->
    Sumo_3(Sumo_1!1,bound~yes).Sumo_3(Sumo_2!1,bound~yes)    k4

42   Sumo_1(Sumo,bound~no) + Sumo_2(Sumo_1,bound~yes)  ->  Sumo_1
    (Sumo!1,bound~no).Sumo_2(Sumo_1!1,bound~yes)           k3
43   Sumo_1(Sumo!1,bound~no).Sumo_2(Sumo_1!1,bound~yes)  ->
    Sumo_1(Sumo!1,bound~yes).Sumo_2(Sumo_1!1,bound~yes)        k4
```

```
44   Sumo_1(Sumo,bound~no) + Sumo_3(Sumo_1,bound~yes)  ->  Sumo_1
    (Sumo!1,bound~no).Sumo_3(Sumo_1!1,bound~yes)           k3
45   Sumo_1(Sumo!1,bound~no).Sumo_3(Sumo_1!1,bound~yes)  ->
    Sumo_1(Sumo!1,bound~yes).Sumo_3(Sumo_1!1,bound~yes)        k4

#Mitte der Kette an neues PML
46   Sumo_2(PML,bound~yes) + PMLbody(Sim,bound~yes)  ->  Sumo_2(
    PML!1,bound~yes).PMLbody(Sim!1,bound~yes)               k3
47   Sumo_3(PML,bound~yes) + PMLbody(Sim,bound~yes)  ->  Sumo_3(
    PML!1,bound~yes).PMLbody(Sim!1,bound~yes)               k3

#Sumo1 an Sp100
57   Sumo_1(Sumo,bound~no) + Sp100(Sumo,bound~yes)      ->
    Sumo_1(Sumo!1,bound~no).Sp100(Sumo!1,bound~yes)         k3
58   Sumo_1(Sumo!1,bound~no).Sp100(Sumo!1,bound~yes)  ->  Sumo_1(
    Sumo!1,bound~yes).Sp100(Sumo!1,bound~yes)      k4

#Sumo2 an Sp100
#59   Sumo_2(Sumo_2,bound~no) + Sp100(Sumo,bound~yes)  ->  Sumo_2
    (Sumo_2!1,bound~no).Sp100(Sumo!1,bound~yes)    k3
#60   Sumo_2(Sumo_2!1,bound~no).Sp100(Sumo!1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~yes).Sp100(Sumo!1,bound~yes) k4

#Sumo3 an Sp100
#61   Sumo_3(Sumo_2,bound~no) + Sp100(Sumo,bound~yes)  ->  Sumo_3
    (Sumo_2!1,bound~no).Sp100(Sumo!1,bound~yes)    k3
#62   Sumo_3(Sumo_2!1,bound~no).Sp100(Sumo!1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~yes).Sp100(Sumo!1,bound~yes) k4

#Ende Kette an neues Sp100
#63   Sumo_1(PML,bound~yes) + Sp100(Sim,bound~yes)  ->  Sumo_1(
    PML!1,bound~yes).Sp100(Sim!1,bound~yes)          k3
#64   Sumo_2(Sumo_1,bound~yes) + Sp100(Sim,bound~yes)  ->  Sumo_2
    (Sumo_1!1,bound~yes).Sp100(Sim!1,bound~yes)    k3
#65   Sumo_3(Sumo_1,bound~yes) + Sp100(Sim,bound~yes)  ->  Sumo_3
    (Sumo_1!1,bound~yes).Sp100(Sim!1,bound~yes)    k3

#Mitte der Kette an neues Sp100
66   Sumo_2(PML,bound~yes) + Sp100(Sim,bound~yes)  ->  Sumo_2(PML
    !1,bound~yes).Sp100(Sim!1,bound~yes)           k3
67   Sumo_3(PML,bound~yes) + Sp100(Sim,bound~yes)  ->  Sumo_3(PML
    !1,bound~yes).Sp100(Sim!1,bound~yes)           k3

#Sumo1 an Daxx
57   Sumo_1(Sumo,bound~no) + Daxx(Sumo,bound~yes)       ->
    Sumo_1(Sumo!1,bound~no).Daxx(Sumo!1,bound~yes)          k3
58   Sumo_1(Sumo!1,bound~no).Daxx(Sumo!1,bound~yes)  ->  Sumo_1(
    Sumo!1,bound~yes).Daxx(Sumo!1,bound~yes)        k4
```

```
#Sumo2 an Daxx
#59   Sumo_2(Sumo_2,bound~no) + Daxx(Sumo,bound~yes)  ->  Sumo_2(
    Sumo_2!1,bound~no).Daxx(Sumo!1,bound~yes)      k3
#60   Sumo_2(Sumo_2!1,bound~no).Daxx(Sumo!1,bound~yes)  ->
    Sumo_2(Sumo_2!1,bound~yes).Daxx(Sumo1!1,bound~yes)  k4

#Sumo3 an Daxx
#61   Sumo_3(Sumo_2,bound~no) + Daxx(Sumo,bound~yes)  ->  Sumo_3(
    Sumo_2!1,bound~no).Daxx(Sumo!1,bound~yes)      k3
#62   Sumo_3(Sumo_2!1,bound~no).Daxx(Sumo!1,bound~yes)  ->
    Sumo_3(Sumo_2!1,bound~yes).Daxx(Sumo!1,bound~yes)   k4

#Sumo123, Ende der Kette, an neues Daxx
#63   Sumo_1(PML,bound~yes) + Daxx(Sim,bound~yes)  ->  Sumo_1(PML
    !1,bound~yes).Daxx(Sim!1,bound~yes)            k3
#64   Sumo_2(Sumo_1,bound~yes) + Daxx(Sim,bound~yes)  ->  Sumo_2(
    Sumo_1!1,bound~yes).Daxx(Sim!1,bound~yes)      k3
#65   Sumo_3(Sumo_1,bound~yes) + Daxx(Sim,bound~yes)  ->  Sumo_3(
    Sumo_1!1,bound~yes).Daxx(Sim!1,bound~yes)      k3

#Mitte der Kette an neues Daxx
66   Sumo_2(PML,bound~yes) + Daxx(Sim1,bound~yes)  ->  Sumo_2(PML
    !1,bound~yes).Daxx(Sim1!1,bound~yes)           k3
67   Sumo_3(PML,bound~yes) + Daxx(Sim1,bound~yes)  ->  Sumo_3(PML
    !1,bound~yes).Daxx(Sim1!1,bound~yes)           k3

68   Sumo_2(PML,bound~yes) + Daxx(Sim2,bound~yes)  ->  Sumo_2(PML
    !1,bound~yes).Daxx(Sim2!1,bound~yes)           k3
69   Sumo_3(PML,bound~yes) + Daxx(Sim2,bound~yes)  ->  Sumo_3(PML
    !1,bound~yes).Daxx(Sim2!1,bound~yes)           k3

#Bindestellen----------------
70   Sumo_1(Sumo!1,PML).Sumo_2(Sumo_1!1) -> Sumo_1(Sumo,PML) +
    Sumo_2(Sumo_1)           k5
71   Sumo_1(Sumo!1,PML).Sumo_3(Sumo_1!1) -> Sumo_1(Sumo,PML) +
    Sumo_3(Sumo_1)           k5
72   Sumo_1(Sumo!1,PML).PMLbody(Sumo1!1) -> Sumo_1(Sumo,PML) +
    PMLbody(Sumo1)           k5
73   Sumo_1(Sumo!1,PML).PMLbody(Sumo2!1) -> Sumo_1(Sumo,PML) +
    PMLbody(Sumo2)           k5
74   Sumo_1(Sumo!1,PML).PMLbody(Sumo3!1) -> Sumo_1(Sumo,PML) +
    PMLbody(Sumo3)           k5
75   Sumo_1(Sumo!1,PML).Daxx(Sumo!1) -> Sumo_1(Sumo,PML) + Daxx(
    Sumo)           k5
76   Sumo_1(Sumo!1,PML).Sp100(Sumo!1) -> Sumo_1(Sumo,PML) + Sp100
    (Sumo)          k5
77   Sumo_1(Sumo,PML,bound~yes)  ->  Sumo_1(Sumo,PML,bound~no)
            k6
```

```
#SIM------------------------
78   PMLbody(Sim!1).Sumo_1(PML!1) -> PMLbody(Sim) + Sumo_1(PML)
            k5
79   PMLbody(Sim!1).Sumo_2(Sumo_1!1) -> PMLbody(Sim) + Sumo_2(
    Sumo_1)           k5
80   PMLbody(Sim!1).Sumo_3(Sumo_1!1) -> PMLbody(Sim) + Sumo_3(
    Sumo_1)           k5
81   PMLbody(Sim!1).Sumo_2(PML!1) -> PMLbody(Sim) + Sumo_2(PML)
            k5
82   PMLbody(Sim!1).Sumo_3(PML!1) -> PMLbody(Sim) + Sumo_3(PML)
            k5
83   Daxx(Sim1!1).Sumo_2(PML!1) -> Daxx(Sim1) + Sumo_2(PML)
            k5
84   Daxx(Sim1!1).Sumo_3(PML!1) -> Daxx(Sim1) + Sumo_3(PML)
            k5
85   Daxx(Sim2!1).Sumo_2(PML!1) -> Daxx(Sim2) + Sumo_2(PML)
            k5
86   Daxx(Sim2!1).Sumo_3(PML!1) -> Daxx(Sim2) + Sumo_3(PML)
            k5
87   Sp100(Sim!1).Sumo_2(PML!1) -> Sp100(Sim) + Sumo_2(PML)
            k5
88   Sp100(Sim!1).Sumo_3(PML!1) -> Sp100(Sim) + Sumo_3(PML)
            k5
end reaction rules

begin parameters

     Sumo_10     1
     Sumo_20     18
     Sumo_30     18
     PMLbody0    0
     Sp1000      1
     Daxx0       1
     Repulsor0   0

     k1      1.5e-2
     k2      1e-00

     k3      1.5e-2
     k4      1.5e-00

     k5      1.5e-07
     k6      1.5e-00

end parameters
```

The other two files describe the simulation process and are not presented here, but in the digital appendix.

D.2 SAC SIMULATION

The simulation of the SAC models was done using the ReaDDy software, which requires multiple input files. Similar to the PML simulation, the particle simulation of the SAC switch requires the geometrical particle description. Note, some of the particles are only temporary pseudo-particles, introduced to achieve ternary reactions that are not included in ReaDDy.

Listing D.3: param_particles.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<param_particles version="1.1">
    <particle>
        <type>Kinetochore_U</type>
        <diffusionConstant>0.0e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.666</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.667</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>Kinetochore_A</type>
        <diffusionConstant>0.0e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.666</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.667</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>Cdc20</type>
        <diffusionConstant>13.0e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.5</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.51</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>O-Mad2</type>
        <diffusionConstant>17.0e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.39</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.4</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>C-Mad2</type>
        <diffusionConstant>17.0e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.390</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.40</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>APC</type>
        <diffusionConstant>5.2e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>1.26</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>1.27</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>Bub</type>
        <diffusionConstant>7.9e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.83</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.84</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>C-Mad2:Cdc20</type>
        <diffusionConstant>11.42e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.57</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.58</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>C-Mad2:Cdc20:C-Mad2</type>
        <diffusionConstant>10.0e+6</diffusionConstant>
        <collisionRadiusMap>
```

```
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.63</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.64</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>MCC</type>
            <diffusionConstant>7.19e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.91</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.92</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>APC:Cdc20</type>
            <diffusionConstant>5.13e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.28</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.29</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>APC:MCC</type>
            <diffusionConstant>4.7e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.40</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.41</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>APC:Cdc20:Bub</type>
            <diffusionConstant>4.7e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.39</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>1.40</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>Cdc20:Bub</type>
            <diffusionConstant>7.19e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.90</radius>
                </collisionRadius>
            </collisionRadiusMap>
            <reactionRadiusMap>
                <reactionRadius>
                    <partnerType>default</partnerType>
                    <radius>0.91</radius>
                </reactionRadius>
            </reactionRadiusMap>
        </particle>
        <particle>
            <type>APC:Cdc20:C-Mad2</type>
            <diffusionConstant>5.00e+6</diffusionConstant>
            <collisionRadiusMap>
                <collisionRadius>
                    <partnerType>default</partnerType>
```

```
                <radius>1.2</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>1.21</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>MK</type>
        <diffusionConstant>0.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>CK</type>
        <diffusionConstant>0.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>AMK</type>
        <diffusionConstant>0.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>ACK</type>
        <diffusionConstant>0.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>ACM</type>
        <diffusionConstant>5.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
    <particle>
        <type>CM</type>
        <diffusionConstant>5.00e+6</diffusionConstant>
        <collisionRadiusMap>
            <collisionRadius>
                <partnerType>default</partnerType>
                <radius>0.44</radius>
            </collisionRadius>
        </collisionRadiusMap>
        <reactionRadiusMap>
            <reactionRadius>
                <partnerType>default</partnerType>
                <radius>0.45</radius>
            </reactionRadius>
        </reactionRadiusMap>
    </particle>
</param_particles>
```

Here, in contrast to the rule-system of the PML bodies, reactions are biochemical which is seen easily by the definition of educts and products.

Listing D.4: param_reactions.xml

```
<param_reactions version="1.1">

    <reaction>
        <name>Attachment</name>
        <type>typeConversion</type>
        <educts>
            <educt      type="particle">Kinetochore_U</educt>
        </educts>
        <products>
            <product    type="particle">Kinetochore_A</product>
        </products>
        <k_forward>85.5</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Closing_Mad</name>
        <type>enzymatic</type>
        <educts>
            <educt      type="particle">Kinetochore_U</educt>
            <educt      type="particle">O-Mad2</educt>
        </educts>
        <products>
            <product    type="particle">Kinetochore_U</product>
            <product    type="particle">C-Mad2</product>
        </products>
        <k_forward>4500000000000</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Decay_Mad</name>
        <type>typeConversion</type>
        <educts>
            <educt      type="particle">C-Mad2</educt>
        </educts>
        <products>
            <product    type="particle">O-Mad2</product>
        </products>
        <k_forward>4500</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Inhibition_1.1</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">C-Mad2</educt>
            <educt      type="particle">Cdc20</educt>
        </educts>
        <products>
            <product    type="particle">C-Mad2:Cdc20</product>
        </products>
        <k_forward>2250000</k_forward>
        <k_backward>225</k_backward>
    </reaction>
    <reaction>
        <name>Inhibition_1.2</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">O-Mad2</educt>
            <educt      type="particle">Cdc20</educt>
        </educts>
        <products>
            <product    type="particle">C-Mad2:Cdc20</product>

        </products>
        <k_forward>22.5</k_forward>
        <k_backward>225</k_backward>
    </reaction>
    <reaction>
        <name>Inhibition_2</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">C-Mad2:Cdc20</educt>
            <educt      type="particle">Bub</educt>
        </educts>
        <products>
            <product    type="particle">MCC</product>
        </products>
        <k_forward>2250000</k_forward>
        <k_backward>225</k_backward>
    </reaction>
    <reaction>
        <name>Inhibition_4</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">APC</educt>
            <educt      type="particle">MCC</educt>
        </educts>
        <products>
            <product    type="particle">APC:MCC</product>
        </products>
        <k_forward>2250000</k_forward>
        <k_backward>1800</k_backward>
    </reaction>
    <reaction>
        <name>Activation</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">APC</educt>
            <educt      type="particle">Cdc20</educt>
        </educts>
        <products>
            <product    type="particle">APC:Cdc20</product>
        </products>
        <k_forward>112500</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Activation_Backwards_1</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">APC:Cdc20</educt>
            <educt      type="particle">MCC</educt>
        </educts>
        <products>
            <product    type="particle">ACM</product>
        </products>
        <k_forward>180000</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Activation_Backwards_2</name>
        <type>fission</type>
        <educts>
            <educt      type="particle">ACM</educt>
        </educts>
        <products>
            <product    type="particle">Cdc20</product>
            <product    type="particle">CM</product>
        </products>
        <k_forward>1000000000000000</k_forward>
        <k_backward>0</k_backward>
```

```
</reaction>
<reaction>
    <name>Activation_Backwards_3</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">CM</educt>
    </educts>
    <products>
        <product    type="particle">APC</product>
        <product    type="particle">MCC</product>
    </products>
    <k_forward>1000000000000000</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Autocat_1</name>
    <type>fusion</type>
    <educts>
        <educt      type="particle">C-Mad2:Cdc20</educt>
        <educt      type="particle">O-Mad2</educt>
    </educts>
    <products>
        <product    type="particle">C-Mad2:Cdc20:C-Mad2</
            product>

    </products>
    <k_forward>4500</k_forward>
    <k_backward>4500</k_backward>
</reaction>
<reaction>
    <name>Autocat_2</name>
    <type>doubleTypeConversion</type>
    <educts>
        <educt      type="particle">C-Mad2:Cdc20:C-Mad2</
            educt>
        <educt      type="particle">Cdc20</educt>
    </educts>
    <products>
        <product    type="particle">C-Mad2:Cdc20</product>
        <product    type="particle">C-Mad2:Cdc20</product>
    </products>
    <k_forward>225000</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Extra_1</name>
    <type>fusion</type>
    <educts>
        <educt      type="particle">Cdc20</educt>
        <educt      type="particle">Bub</educt>
    </educts>
    <products>
        <product    type="particle">Cdc20:Bub</product>

    </products>
    <k_forward>225</k_forward>
    <k_backward>2250</k_backward>
</reaction>
<!--Silencing Reaction, Inhibiton works fine until here -->
<reaction>
    <name>Catalyzed_Decay_2</name>
    <type>fusion</type>
    <educts>
        <educt      type="particle">APC:MCC</educt>
        <educt      type="particle">Kinetochore_A</educt>
    </educts>
    <products>
        <product    type="particle">AMK</product>
    </products>
    <k_forward>22500</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Catalyzed_Decay_3</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">AMK</educt>
    </educts>
    <products>
        <product    type="particle">ACK</product>
        <product    type="particle">O-Mad2</product>
    </products>
    <k_forward>1000000000000000</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Catalyzed_Decay_4</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">ACK</educt>
    </educts>
    <products>
        <product    type="particle">APC:Cdc20:Bub</product>
        <product    type="particle">Kinetochore_A</product>
    </products>
    <k_forward>1000000000000000</k_forward>
    <k_backward>0</k_backward>
</reaction>
```

```
<reaction>
    <name>Catalyzed_Decay_MCC_1</name>
    <type>fusion</type>
    <educts>
        <educt      type="particle">MCC</educt>
        <educt      type="particle">Kinetochore_A</educt>
    </educts>
    <products>
        <product    type="particle">MK</product>
    </products>
    <k_forward>22500</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Catalyzed_Decay_MCC_2</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">MK</educt>
    </educts>
    <products>
        <product    type="particle">O-Mad2</product>
        <product    type="particle">CK</product>
    </products>
    <k_forward>1000000000000000</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Catalyzed_Decay_MCC_3</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">CK</educt>
    </educts>
    <products>
        <product    type="particle">Cdc20:Bub</product>
        <product    type="particle">Kinetochore_A</product>
    </products>
    <k_forward>1000000000000000</k_forward>
    <k_backward>0</k_backward>
</reaction>
<reaction>
    <name>Catalyzed_Decay_5</name>
    <type>fission</type>
    <educts>
        <educt      type="particle">APC:Cdc20:Bub</educt>
    </educts>
    <products>
        <product    type="particle">APC:Cdc20</product>
        <product    type="particle">Bub</product>
    </products>
```

```
        <k_forward>2250</k_forward>
        <k_backward>0</k_backward>
    </reaction>
    <reaction>
        <name>Alternativ_1</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">APC:Cdc20</educt>
            <educt      type="particle">C-Mad2</educt>
        </educts>
        <products>
            <product    type="particle">APC:Cdc20:C-Mad2</product
                >

        </products>
        <k_forward>225000</k_forward>
        <k_backward>225</k_backward>
    </reaction>
    <reaction>
        <name>Alternativ_2</name>
        <type>fusion</type>
        <educts>
            <educt      type="particle">APC:Cdc20:C-Mad2</educt>
            <educt      type="particle">Bub</educt>
        </educts>
        <products>
            <product    type="particle">APC:MCC</product>

        </products>
        <k_forward>225000</k_forward>
        <k_backward>225</k_backward>
    </reaction>
</param_reactions>
```

Some other files describe the simulation process and can be found in the digital appendix.

E DIGITAL APPENDIX

E.1 DIGITAL CONTENT

All proposed models and novel developed analysis tools, as well as the thesis itself are presented on the accompanying data medium. The folder structure is shown in Figure E.1 and follows strictly the section

```
/
├─ 2 Novel methods to coarse-grain particle simulations (Chapter 2)
│  └─ DiCoSAD/ Python source files of the DiCoSAD software
├─ 3 Coarse-graining models of the spindle assembly checkpoint (Chapter 3)
│  ├─ FullModel/ Full SAC model
│  │  ├─ Copasi/ .xml file containing the ODE version
│  │  └─ Readdy/ Readdy files describing the particle based model
│  ├─ SimpleModel/ Simple reduced SAC model
│  │  ├─ Copasi/ .xml file containing the ODE version
│  │  └─ Readdy/ Readdy files describing the particle based model
│  ├─ EssentialModel/ Essential SAC model
│  │  ├─ Copasi/ .xml file containing the ODE version
│  │  ├─ Readdy/ Readdy files describing the particle based model
│  │  └─ XPPAUT/ Xppaut version to perform the bifurcation nalysis
│  └─ 4StateModel/ Thermodynamically inconsistent pathways
│     ├─ Python/ Python files for the numerical simulation
│     └─ PRISM/ PRISM files for the exact analysis
├─ 4 Coarse-graining the formation of PML nuclear bodies (Chapter 4)
│  ├─ Methods/ Analysis methods for comparing PML nuclear bodies in-silico
│  │  ├─ RRD/ Python script for evaluating the relative radial density
│  │  ├─ RDF/ Python script for evaluating the radial distribution function
│  │  └─ VirtualMicroscopy/ Python script for visualizing in-silico simulations
│  └─ SRSim/ SRSim files for the self-assembling formation of nuclear bodies
├─ README Conclusion of the data structure in the folder
└─ PHD_Thesis.pdf Digital version of the thesis
```

Figure E.1: **Directory structure of the digital appendix.** Every directory contains model data and here developed analysis tools of the regarding section. The ODE models are mostly in xml format, while the particle based models were realized with SRSim or ReaDDy and have the respective format.

structure of the thesis. Raw output of the particle simulations are not shown, as they would exceed the available data limit.

BIBLIOGRAPHY

[Alo06] U Alon. *An introduction to systems biology: design principles of biological circuits*. London: CRC press, 2006.

[And65] P Andrews. "The gel-filtration behaviour of proteins related to their molecular weights over a wide range". In: *Biochemical Journal* 96.3 (1965), p. 595.

[AB04] S S Andrews and D Bray. "Stochastic simulation of chemical reactions with spatial resolution and single molecule detail". In: *Physical Biology* 1.3-4 (2004), pp. 137 –151. DOI: 10.1088/1478-3967/1/3/001.

[And+10] S S Andrews, N J Addy, R Brent, and A P Arkin. "Detailed simulations of cell biology with Smoldyn 2.1". In: *PLOS Computational Biology* 6.3 (2010), e1000705. DOI: 10.1371/journal.pcbi.1000705.

[Avo11] A Avogadro. "Essai d'une manière de déterminer les masses relatives des molécules élémentaires des corps, et les proportions selon lesquelles elles entrent dans les combinaisons". In: *Journal de Physique* 73.1 (1811), pp. 58–76.

[Bec+11] M Beck, A Schmidt, J Malmstroem, M Claassen, A Ori, A Szymborska, F Herzog, O Rinner, J Ellenberg, and R Aebersold. "The quantitative proteome of a human cell line". In: *Molecular Systems Biology* 7.1 (2011), p. 549. DOI: 10.1038/msb.2011.82.

[BP07] R Bernardi and P P Pandolfi. "Structure, dynamics and functions of promyelocytic leukaemia nuclear bodies". In: *Nature Reviews Molecular Cell Biology* 8.12 (2007), p. 1006. DOI: 10.1038/nrm2277.

[BPP08] R Bernardi, A Papa, and P P Pandolfi. "Regulation of apoptosis by PML and the PML-NBs". In: *Oncogene* 27.48 (2008), p. 6299. DOI: 10.1038/onc.2008.305.

[Bli+04] M L Blinov, J R Faeder, B Goldstein, and W S Hlavacek. "BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains". In: *Bioinformatics* 20.17 (2004), pp. 3289–3291. DOI: 10.1093/bioinformatics/bth378.

[Bod+97] M N Boddy, E Duprez, K L Borden, and P S Freemont. "Surface residue mutations of the PML RING finger domain alter the formation of nuclear matrix-associated PML bodies". In: *Journal of Cell Science* 110.18 (1997), pp. 2197–2205.

[Bri+98] J M Bridger, H Herrmann, C Munkel, and P Lichter. "Identification of an interchromosomal compartment by polymerization of nuclear-targeted vimentin". In: *Journal of Cell Science* 111.9 (1998), pp. 1241 –1253.

[Cha+12] W C H Chao, K Kulkarni, Z Zhang, E H Kong, and D Barford. "Structure of the mitotic checkpoint complex". In: *Nature* 484.7393 (2012), p. 208. DOI: 10.1038/nature10896.

[CA+98] M K Chelbi-Alix, F Quignon, L Pelicano, M H M Koken, et al. "Resistance to virus infection conferred by the interferon-induced promyelocytic leukemia protein". In: *Journal of Virology* 72.2 (1998), pp. 1043 –1051.

[CL14] J Chen and J Liu. "Spatial-temporal model for silencing of the mitotic spindle assembly checkpoint". In: *Nature Communications* 5.1 (2014), p. 4795. DOI: 10.1038/ncomms5795.

[CC03] L Y Chen and J D Chen. "Daxx silencing sensitizes cells to multiple apoptotic pathways". In: *Molecular and Cellular Biology* 23.20 (2003), pp. 7108–7121. DOI: 10.1128/MCB.23.20.7108-7121.2003.

[CLR11] I Chung, H Leonhardt, and K Rippe. "De novo assembly of a PML nuclear subcompartment occurs through multiple pathways and induces telomere elongation". In: *Journal of Cell Science* (2011), jcs–084681. DOI: 10.1242/jcs.084681.

[Col+13] P Collin, O Nashchekina, Rl Walker, and J Pines. "The spindle assembly checkpoint works like a rheostat rather than a toggle switch". In: *Nature Cell Biology* 15.11 (2013), pp. 1378 –1385. DOI: 10.1038/ncb2855.

[CH97] G M Cooper and R E Hausman. *The cell: Molecular approach*. Zagreb: Medicinska naklada, 1997.

[CIR52] R Courant, E Isaacson, and M Rees. "On the solution of nonlinear hyperbolic differential equations by finite differences". In: *Communications on Pure and Applied Mathematics* 5.3 (1952), pp. 243 –255. DOI: 10.1002/cpa.3160050303.

[DL04] V Danos and C Laneve. "Formal molecular biology". In: *Theoretical Computer Science* 325.1 (2004), pp. 69 –110. DOI: 10.1016/j.tcs.2004.03.065.

[DB08] M I Davidich and S Bornholdt. "Boolean network model predicts cell cycle sequence of fission yeast". In: *PLOS ONE* 3.2 (2008), e1672. DOI: 10.1371/journal.pone.0001672.

[DSM05] A DeAntoni, V Sala, and A Musacchio. "Explaining the oligomerization properties of the spindle assembly checkpoint protein Mad2". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 360.1455 (2005), pp. 637 –647. DOI: 10.1098/rstb.2004.1618.

[DBJ04] G Dellaire and D P Bazett-Jones. "PML nuclear bodies: dynamic sensors of DNA damage and cellular stress". In: *Bioessays* 26.9 (2004), pp. 963 –977. DOI: 10.1002/bies.20089.

[Dem18] A J Dempster. "A new method of positive ray analysis". In: *Physical Review* 11.4 (1918), p. 316. DOI: 10.1103/PhysRev.11.316.

[DG13] A E Dick and D W Gerlich. "Kinetic framework of spindle assembly checkpoint signalling". In: *Nature Cell Biology* 15.11 (2013), pp. 1370 –1377. DOI: 10.1038/ncb2842.

[DBJB05] A Doncic, E Ben-Jacob, and N Barkai. "Evaluating putative mechanisms of the mitotic spindle checkpoint". In: *Proceedings of the National Academy of Sciences* 102.18 (2005), pp. 6332 –6337. DOI: 10.1073/pnas.0409142102.

[Dyc+94] J A Dyck, G G Maul, W H Miller Jr, J D Chen, A Kakizuka, and R M Evans. "A novel macromolecular structure is a target of the promyelocyte-retinoic acid receptor oncoprotein". In: *Cell* 76.2 (1994), pp. 333 –343. DOI: 10.1016/0092-8674(94)90340-9.

[Dyk15] E C Dykeman. "An implementation of the Gillespie algorithm for RNA kinetics with logarithmic time update". In: *Nucleic Acids Research* 43.12 (2015), pp. 5708 –5715. DOI: 10.1093/nar/gkv480.

[EC09] R Erban and S J Chapman. "Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions". In: *Physical Biology* 6.4 (2009), p. 046001. DOI: 10.1088/1478-3975/6/4/046001.

[Eri09] H P Erickson. “Size and shape of protein molecules at the nanometer level determined by sedimentation, gel filtration, and electron microscopy”. In: *Biological Procedures Online* 11.1 (2009), pp. 32–51. DOI: 10.1007/s12575-009-9008-x.

[EM78] D L Ermak and J A McCammon. “Brownian dynamics with hydrodynamic interactions”. In: *Journal of Chemical Physics* 69.4 (1978), pp. 1352 –1360. DOI: 10.1063/1.436761.

[EC+10] E Escobar-Cabrera, D K W Lau, S Giovinazzi, A M Ishov, and L P McIntosh. “Structural characterization of the DAXX N-terminal helical bundle domain and its complex with Rassf1C”. In: *Structure* 18.12 (2010), pp. 1642 –1653. DOI: 10.1016/j.str.2010.09.016.

[Fae+17] A C Faesen, M Thanasoula, S Maffini, C Breit, F Müller, S van Gerwen, T Bange, and A Musacchio. “Basis of catalytic assembly of the mitotic checkpoint complex”. In: *Nature* 542.7642 (2017), p. 498. DOI: 10.1038/nature21384.

[Fan02] G Fang. “Checkpoint protein BubR1 acts synergistically with Mad2 to inhibit anaphase-promoting complex”. In: *Molecular Biology of the Cell* 13.3 (2002), pp. 755 –766. DOI: 10.1091/mbc.01-09-0437.

[Fic95] A Fick. “On liquid diffusion”. In: *Journal of Membrane Science* 100.1 (1995), pp. 33 –38. DOI: 10.1016/0376-7388(94)00230-V.

[GS79] I I Gihman and A V Skorohod. “Stochastic differential equations”. In: *The Theory of Stochastic Processes III*. Ed. by S S Chern et al. Berlin Heidelberg: Springer, 1979, pp. 113 –219.

[Gil77] D T Gillespie. “Exact stochastic simulation of coupled chemical reactions”. In: *Journal of Physical Chemistry* 81.25 (1977), pp. 2340 –2361. DOI: 10.1021/j100540a008.

[Goo+14] C Good, M Puljiz, D Parker, and J E Rowe. *Technical Report on Work Package 1: Theory of Hierarchical Structure*. 2014. DOI: http://www.cs.bham.ac.uk/research/projects/hieratic/docs/hieratic-D1.3.pdf.

[Gri+13] M Griebel, S Knapek, G Zumbusch, and A Caglar. *Numerische Simulation in der Moleküldynamik: Numerik, Algorithmen, Parallelisierung, Anwendungen*. Berlin Heidelberg: Springer, 2013. DOI: 10.1007/978-3-642-18779-7.

[GHN03] S Gruber, C H Haering, and K Nasmyth. “Chromosomal cohesin forms a ring”. In: *Cell* 112.6 (2003), pp. 765 –777. DOI: 10.1016/S0092-8674(03)00162-4.

[GD11] G Gruenert and P Dittrich. “Using the SRSim Software for Spatial and Rule-Based Modeling of Combinatorially Complex Biochemical Reaction Systems”. In: *Membrane Computing: 11th International Conference, CMC 2010, Jena, Germany, August 24-27, 2010. Revised Selected Papers*. Ed. by M Gheorghe, T Hinze, G Păun, G Rozenberg, and A Salomaa. Berlin Heidelberg: Springer, 2011, pp. 240 –256. DOI: 10.1007/978-3-642-18123-8_19.

[Gru+10] G Gruenert, B Ibrahim, T Lenser, M Lohel, T Hinze, and P Dittrich. “Rule-based spatial modeling with diffusing, geometrically constrained molecules”. In: *BMC Bioinformatics* 11.1 (2010), p. 307. DOI: 10.1186/1471-2105-11-307.

[Gut+07] R N Gutenkunst, J J Waterfall, Fl P Casey, K S Brown, C R Myers, and J P Sethna. “Universally sloppy parameter sensitivities in systems biology models”. In: *PLOS Computational Biology* 3.10 (2007), e189. DOI: 10.1371/journal.pcbi.0030189.

[Hag+11] R S Hagan, M S Manak, H K Buch, M G Meier, P Meraldi, J V Shah, and P K Sorger. “p31comet acts to ensure timely spindle checkpoint silencing subsequent to kinetochore attachment”. In: *Molecular Biology of the Cell* 22.22 (2011), pp. 4236 –4246. DOI: 10.1091/mbc.E11-03-0216.

[Han+13] J S Han, A J Holland, D Fachinetti, A Kulukian, B Cetin, and D W Cleveland. “Catalytic assembly of the mitotic checkpoint inhibitor BubR1-Cdc20 by a Mad2-induced functional switch in Cdc20”. In: *Molecular Cell* 51.1 (2013), pp. 92 –104. DOI: 10.1074/jbc.M803594200.

[HKW81] B D Hassard, N D Kazarinoff, and Y-H Wan. *Theory and applications of Hopf bifurcation*. Vol. 41. Cambridge: CUP Archive, 1981.

[Hec+06] C-M Hecker, M Rabiller, K Haglund, P Bayer, and I Dikic. “Specification of SUMO1-and SUMO2-interacting motifs”. In: *Journal of Biological Chemistry* 281.23 (2006), pp. 16117 –16127. DOI: 10.1074/jbc.M512757200.

[Hen18a] R Henze. *DiCoSAD: Particle Simulator that coarse-grains the Diffusion*. https://github.com/DonNeoMir/DICOSAD. Version 1.0. 2018.

[Hen18b] R Henze. *VirtualMicroscopy: Visualizes Particle Configurations*. https : / / github . com / DonNeoMir / VirtualMicroscopy. Version 1.0. 2018.

[HDI17] R Henze, P Dittrich, and B Ibrahim. "A Dynamical Model for Activating and Silencing the Mitotic Checkpoint". In: *Scientific Reports* 7.1 (2017), p. 3865. DOI: 10.1038/s41598-017-04218-2.

[Hen+15] R Henze, J Huwald, N Mostajo, P Dittrich, and B Ibrahim. "Structural analysis of in silico mutant experiments of human inner-kinetochore structure". In: *Biosystems* 127.1 (2015), pp. 47 –59. DOI: https://doi.org/10.1016/j.biosystems.2014.11.004.

[Hen+18a] R Henze et al. "Multi-scale stochastic organization-oriented coarse-graining exemplified on the human mitotic checkpoint". In: *Scientific Reports* (2018). *accepted*.

[Hen+18b] R Henze, G Gruenert, B Ibrahim, and P Dittrich. *Spatial Rule-based Simulations: the SRSim Software*. Ed. by William Hlavacek. *to appear*. Springer, 2018.

[Hid80] T Hida. *Brownian Motion*. New York, NY: Springer, 1980, pp. 44 –113. DOI: 10.1007/978-1-4612-6030-1_2.

[Hil10] A V Hill. "The possible effects of the aggregation of the molecules of hæmoglobin on its dissociation curves". In: *Journal of Physiology* 40.1 (1910), pp. i –vii.

[Hoi+18] C Hoischen, S Monajembashi, K Weisshart, and P Hemmerich. "Multimodal light microscopy approaches to reveal structural and functional properties of promyelocytic leukemia nuclear bodies". In: *Frontiers in Oncology* 8.1 (2018). DOI: 10.3389/fonc.2018.00125.

[How+00] B J Howell, D B Hoffman, G Fang, A W Murray, and E D Salmon. "Visualization of Mad2 Dynamics at Kinetochores, along Spindle Fibers, and at Spindle Poles in Living Cells". In: *Journal of Cell Biology* 150.6 (2000), pp. 1233 –1250. DOI: 10.1083/jcb.150.6.1233.

[How+01] B J Howell, B F McEwen, J C Canman, D B Hoffman, E M Farrar, C L Rieder, and E D Salmon. "Cytoplasmic dynein/dynactin drives kinetochore protein transport to the spindle poles and has a role in mitotic spindle checkpoint inactivation". In: *Journal of Cell Biology* 155.7 (2001), pp. 1159 –1172. DOI: 10.1083/jcb.200105093.

[Hua+04] W C Huang, T P Ko, S S L Li, and A H J Wang. "Crystal structures of the human SUMO-2 protein at 1.6 Å and 1.2 Å resolution: Implication on the functional differences of SUMO proteins". In: *European Journal of Biochemistry* 271.20 (2004), pp. 4114–4122. DOI: 10.1111/j.1432-1033.2004.04349.x.

[Ibr+08a] B Ibrahim, S Diekmann, E Schmitt, and P Dittrich. "In-silico modeling of the mitotic spindle assembly checkpoint". In: *PLOS ONE* 3.2 (2008), e1555. DOI: 10.1371/journal.pone.0001555.

[Ibr+08b] B Ibrahim, P Dittrich, S Diekmann, and E Schmitt. "Mad2 binding is not sufficient for complete Cdc20 sequestering in mitotic transition control (an in silico study)". In: *Biophysical Chemistry* 134.1-2 (2008), pp. 93 –100. DOI: 10.1016/j.bpc.2008.01.007.

[Ibr+09] B Ibrahim, E Schmitt, P Dittrich, and S Diekmann. "In silico study of kinetochore control, amplification, and inhibition effects in MCC assembly". In: *Biosystems* 95.1 (2009), pp. 35 –50. DOI: 10.1016/j.biosystems.2008.06.007.

[Ibr+13] B Ibrahim, R Henze, G Gruenert, M Egbert, J Huwald, and P Dittrich. "Spatial rule-based modeling: a method and its application to the human mitotic kinetochore". In: *Cells* 2.3 (2013), pp. 506 –544. DOI: 10.3390/cells2030506.

[Ish+99] A M Ishov, A G Sotnikov, D Negorev, O V Vladimirova, N Neff, T Kamitani, E T H Yeh, J F Strauss, and G G Maul. "PML is critical for ND10 formation and recruits the PML-interacting protein daxx to this nuclear structure when modified by SUMO-1". In: *Journal of Cell Biology* 147.2 (1999), pp. 221–234. DOI: 10.1083/jcb.147.2.221.

[IP12] D Izawa and J Pines. "Mad2 and the APC/C compete for the same site on Cdc20 to ensure proper chromosome segregation". In: *Journal of Cell Biology* 199.1 (2012), pp. 27 –37. DOI: 10.1083/jcb.201205170.

[IP15] D Izawa and J Pines. "The mitotic checkpoint complex binds a second CDC20 to inhibit active APC/C". In: *Nature* 517.7536 (2015), pp. 631 –634. DOI: 10.1038/nature13911.

[JST81] A Jameson, W Schmidt, and E Turkel. "Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes". In: *14th fluid and plasma dynamics conference*. 1981, p. 1259. DOI: 10.2514/6.1981-1259.

[JRK02] M S Jang, S W Ryu, and E Kim. "Modification of Daxx by small ubiquitin-related modifier-1". In: *Biochemical and Biophysical Research Communications* 295.2 (2002), pp. 495–500. DOI: 10.1016/S0006-291X(02)00699-X.

[Joh04] E S Johnson. "Protein modification by SUMO". In: *Annual Review of Biochemistry* 73.1 (2004), pp. 355 –382. DOI: 10.1146/annurev.biochem.73.011303.074118.

[Kam92] N G van Kampen. *Stochastic processes in physics and chemistry*. Vol. 1. New York: Elsevier, 1992.

[Kau69] S Kauffman. "Homeostasis and differentiation in random genetic control networks". In: *Nature* 224.5215 (1969), pp. 177 –178. DOI: 10.1038/224177a0.

[Kha+01] M M Khan, T Nomura, H Kim, S C Kaul, R Wadhwa, T Shinagawa, E Ichikawa-Iwata, S Zhong, P P Pandolfi, and S Ishii. "Role of PML and PML-RARα in Mad-mediated transcriptional repression". In: *Molecular Cell* 7.6 (2001), pp. 1233 –1243. DOI: 10.1016/S1097-2765(01)00257-X.

[Kni+08] P Knipscheer, A Flotho, H Klug, J V Olsen, W J van Dijk, A Fish, E S Johnson, M Mann, T K Sixma, and A Pichler. "Ubc9 sumoylation regulates SUMO target discrimination". In: *Molecular Cell* 31.3 (2008), pp. 371 –382. DOI: 10.1016/j.molcel.2008.05.022.

[KWC04] G J P L Kops, B A A Weaver, and D W Cleveland. "On the road to cancer: aneuploidy and the mitotic checkpoint". In: *Nature Reviews Cancer* 5.10 (2004), p. 773. DOI: 10.1038/nrc1714.

[KNP11] M Kwiatkowska, G Norman, and D Parker. "PRISM 4.0: Verification of probabilistic real-time systems". In: *International conference on computer aided verification*. Ed. by Gopalakrishnan G and QadeerS. Vol. 6806. Berlin Heidelberg: Springer, 2011, pp. 585 –591. DOI: 10.1007/978-3-642-22110-1_47.

[Lac16] Silvestre François Lacroix. *An elementary treatise on the differential and integral calculus*. Cambridge: J. Smith, 1816.

[LB+01] V Lallemand-Breitenbach, J Zhu, F Puvion, M Koken, N Honoré, A Doubeikovsky, E Duprez, P P Pandolfi, E Puvion, P Freemont, et al. "Role of promyelocytic leukemia (PML) sumolation in nuclear body formation, 11S proteasome recruitment, and As2O3-induced PML or PML/retinoic acid receptor α degradation". In: *Journal of Experimental Medicine* 193.12 (2001), pp. 1361 –1372. DOI: 10.1084/jem.193.12.1361.

[Lan+10] M Lang, T Jegou, I Chung, K Richter, S Münch, A Udvarhelyi, C Cremer, P Hemmerich, J Engelhardt, S W Hell, et al. "Three-dimensional organization of promyelocytic leukemia nuclear bodies". In: *Journal of Cell Science* 123.3 (2010), pp. 392 –400. DOI: 10.1242/jcs.053496.

[LW75] M Levitt and A Warshel. "Computer simulation of protein folding". In: *Nature* 253.5494 (1975), p. 694. DOI: doi:10.1038/253694a0.

[LWH15] X Liu, E S Welf, and J M Haugh. "Linking morphodynamics and directional persistence of T lymphocyte migration". In: *Journal of the Royal Society Interface* 12.106 (2015), p. 20141412. DOI: 10.1098/rsif.2014.1412.

[Loh+09] M Lohel, B Ibrahim, S Diekmann, and P Dittrich. "The role of localization in the operation of the mitotic spindle assembly checkpoint". In: *Cell Cycle* 8.16 (2009), pp. 2650 –2660. DOI: 10.4161/cc.8.16.9383.

[Map+06] M Mapelli, F V Filipp, G Rancati, L Massimiliano, L Nezi, G Stier, R S Hagan, S Confalonieri, S Piatti, M Sattler, et al. "Determinants of conformational dimerization of Mad2 and its inhibition by p31comet". In: *EMBO Journal* 25.6 (2006), pp. 1273 –1284. DOI: 10.1038/sj.emboj.7601033.

[MN10] A A Markov and N M Nagorny. *The Theory of Algorithms*. Berlin Heidelberg: Springer, 2010. DOI: 10.2307/2266585.

[Mar71] Andrey Markov. "Extension of the limit theorems of probability theory to a sum of variables connected in a chain". In: *Dynamic Probabilistic Systems, Volume 1: Markov Models*. Ed. by R Howard. Hoboken: John Wiley & Sons, Inc., 1971, pp. 552 –577.

[Mar+15] S Marques, J Fonseca, P M Silva, and H Bousbaa. "Targeting the spindle assembly checkpoint for breast cancer treatment". In: *Current Cancer Drug*

Targets 15.4 (2015), pp. 272 –281. DOI: 10.2174/1568009615666150302130010.

[MW97] A D McNaught and A Wilkinson. *Compendium of chemical terminology*. Vol. 1669. Oxford: Blackwell Science Oxford, 1997.

[Mic10] X Michalet. "Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium". In: *Physical Review E* 82.4 (2010), p. 041914. DOI: 10.1103/PhysRevE.82.041914.

[Mis+08] H B Mistry, D E MacCallum, R C Jackson, M A J Chaplain, and F A Davidson. "Modeling the temporal evolution of the spindle assembly checkpoint and role of Aurora B kinase". In: *Proceedings of the National Academy of Sciences* 105.51 (2008), pp. 20215 –20220. DOI: 10.1073/pnas.0810706106.

[MCN08] I Mura and A Csikász-Nagy. "Stochastic Petri Net extension of a yeast cell cycle model". In: *Journal of Theoretical Biology* 254.4 (2008), pp. 850 –860. DOI: 10.1016/j.jtbi.2008.07.019.

[Mur94] A Murray. "Cell cycle checkpoints". In: *Current Opinion in Cell Biology* 6.6 (1994), pp. 872 –876. DOI: 10.1016/0955-0674(94)90059-0.

[Mus15] A Musacchio. "The molecular biology of spindle assembly checkpoint signaling dynamics". In: *Current Biology* 25.20 (2015), R1002 –R1018. DOI: 10.1016/j.cub.2015.08.051.

[MS07] A Musacchio and E D Salmon. "The spindle-assembly checkpoint in space and time". In: *Nature Reviews Molecular Cell Biology* 8.5 (2007), pp. 379 –393. DOI: 10.1038/nrm2163.

[NM01] D Negorev and G G Maul. "Cellular proteins localized at and interacting within ND10/PML nuclear bodies/PODs suggest functions of a nuclear depot". In: *Oncogene* 20.49 (2001), p. 7234. DOI: 10.1038/sj.onc.1204764.

[Nis+13] S Nisole, M A Maroui, X Mascle, M Aubry, and M K Chelbi-Alix. "Differential roles of PML isoforms". In: *Frontiers in Oncology* 3.1 (2013), p. 125. DOI: 10.3389/fonc.2013.00125.

[Nov+07] B Novak, J J Tyson, B Gyorffy, and A Csikasz-Nagy. "Irreversible cell-cycle transitions are due to systems-level feedback". In: *Nature Cell Biology* 9.7 (2007), p. 724. DOI: 10.1038/ncb0707-724.

[PP01] M Pearson and P G Pelicci. "PML interaction with p53 and its role in apoptosis and replicative senescence". In: *Oncogene* 20.49 (2001), p. 7250. DOI: 10.1038/sj.onc.1204856.

[Pet62] C A Petri. "Kommunikation mit Automaten". In: (1962).

[Phi+12] R Phillips, J Kondev, J Theriot, and H Garcia. *Physical biology of the cell*. New York: Garland Science, 2012.

[PST17] D Potter, J Stadel, and R Teyssier. "PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys". In: *Computational Astrophysics and Cosmology* 4.1 (2017), p. 2. DOI: 10.1186/s40668-017-0021-1.

[Red+07] S K Reddy, M Rape, W A Margansky, and M W Kirschner. "Ubiquitination by the anaphase-promoting complex drives spindle checkpoint inactivation". In: *Nature* 446.7138 (2007), pp. 921 –925. DOI: 10.1038/nature05734.

[RB+60] M Riviere, W Bernhard, et al. "Examen au microscope électronique de la tumeur VX2 du lapin domestique dérivée du papillome de Shope". In: *Bulletin du Cancer* 47.1 (1960), pp. 570 –584.

[RST08] H G Roos, M Stynes, and L Tobiska. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*. Vol. 24. Berlin Heidelberg: Springer, 2008. DOI: 10.1007/978-3-540-34467-4.

[RB98] R E Rudd and J Q Broughton. "Coarse-grained molecular dynamics and the atomic limit of finite elements". In: *Physical Review B* 58.10 (1998), R5893. DOI: 10.1103/PhysRevB.58.R5893.

[RM96] A D Rudner and A W Murray. "The spindle assembly checkpoint". In: *Current Opinion in Cell Biology* 8.6 (1996), pp. 773 –780. DOI: 10.1016/S0955-0674(96)80077-9.

[SP02] P Salomoni and P P Pandolfi. "The role of PML in tumor suppression". In: *Cell* 108.2 (2002), pp. 165 –170. DOI: 10.1016/S0092-8674(02)00626-8.

[SM09] S Santaguida and A Musacchio. "The life and miracles of kinetochores". In: *EMBO Journal* 28.17 (2009), pp. 2511 –2531. DOI: 10.1038/emboj.2009.173.

[Sch+14] J Schoneberg, M Heck, K P Hofmann, and F Noe. "Explicit spatiotemporal simulation of receptor-G protein coupling in rod cell disk membranes". In: *Biophysical Journal* 107.5 (2014), pp. 1042 –1053. DOI: 10.1016/j.bpj.2014.05.050.

[Sch+11] B Schwanhäusser, D Busse, N Li, G Dittmar, J Schuchhardt, J Wolf, W Chen, and M Selbach. "Global quantification of mammalian gene expression control". In: *Nature* 473.7347 (2011), p. 337. DOI: 10.1038/nature10098.

[SN13] J Schöneberg and F Noé. "ReaDDy - A Software for Particle-Based Reaction-Diffusion Dynamics in Crowded Cellular Environments". In: *PLOS ONE* 8.9 (2013), pp. 1–14. DOI: 10.1371/journal.pone.0074261.

[SH06] R P Sear and M Howard. "Modeling dual pathways for the metazoan spindle assembly checkpoint". In: *Proceedings of the National Academy of Sciences* 103.45 (2006), pp. 16758 –16763. DOI: 10.1073/pnas.0603174103.

[See+01] J S Seeler, A Marchio, R Losson, J M P Desterro, R T Hay, P Chambon, and A Dejean. "Common properties of nuclear body protein SP100 and TIF1α chromatin factor: role of SUMO modification". In: *Molecular and Cellular Biology* 21.10 (2001), pp. 3314–3324. DOI: 10.1128/MCB.21.10.3314-3324.2001.

[Smo18] M von Smoluchowski. "Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen". In: *Zeitschrift für physikalische Chemie* 92.1 (1918), pp. 129 –168. DOI: 10.1515/zpch-1918-9209.

[Ste+14] S N Steinway, J G Zanudo, W Ding, C B Rountree, D J Feith, T P Loughran, and R Albert. "Network modeling of TGFβ signaling in hepatocellular carcinoma epithelial-to-mesenchymal transition reveals joint sonic hedgehog and Wnt pathway activation". In: *Cancer Research* 74.21 (2014), pp. 5963 –5977. DOI: 10.1158/0008-5472.CAN-14-0225.

[SJW97] T Sternsdorf, K Jensen, and H Will. "Evidence for Covalent Modification of the Nuclear Dot - associated Proteins PML and Sp100 by PIC1/SUMO-1". In: *Journal of Cell Biology* 139.7 (1997), pp. 1621 –1634. DOI: 10.1083/jcb.139.7.1621.

[SL96] A B Stundzia and C J Lumsden. "Stochastic simulation of coupled reaction - diffusion processes". In: *Journal of Computational Physics* 127.1 (1996), pp. 196 –207. DOI: 10.1006/jcph.1996.0168.

[Szo+90] C Szostecki, H H Guldner, H J Netter, and H Will. "Isolation and characterization of cDNA encoding a human nuclear antigen predominantly recognized by autoantibodies from patients with primary biliary cirrhosis." In: *Journal of Immunology* 145.12 (1990), pp. 4338 –4347.

[Tan+01] Z Tang, R Bharadwaj, B Li, and H Yu. "Mad2-independent inhibition of APC Cdc20 by the mitotic checkpoint protein BubR1". In: *Developmental Cell* 1.2 (2001), pp. 227 –237. DOI: 10.1016/S1534-5807(01)00019-3.

[Tho02] N L Thompson. "Fluorescence correlation spectroscopy". In: *Topics in Fluorescence Spectroscopy, Volume 1: Techniques*. Ed. by J R Lakowicz. New York: Plenum Press, 2002, pp. 337 –378. DOI: 10.1007/0-306-47057-8_6.

[Tsc+98] W Tschöp, K Kremer, J Batoulis, T Bürger, and O Hahn. "Simulation of polymer melts. I. Coarse-graining procedure for polycarbonates". In: *Acta Polymerica* 49.2-3 (1998), pp. 61 –74. DOI: 10.1002/(SICI)1521-4044(199802)49:2/3<61::AID-APOL61>3.0.CO;2-V.

[Tys+08] J J Tyson, R Albert, A Goldbeter, P Ruoff, and J Sible. "Biological switches and clocks". In: *Journal of The Royal Society Interface* 5.1 (2008), pp. 1 –8. DOI: 10.1098/rsif.2008.0179.focus.

[VH55] L Van Hove. "Energy corrections and persistent perturbation effects in continuous spectra". In: *Physica* 21.6 (1955), pp. 901 –923. DOI: 10.1016/S0031-8914(55)92832-9.

[Ver+13] A Verdugo, P K Vinod, J J Tyson, and B Novak. "Molecular mechanisms creating bistable switches at cell cycle transitions". In: *Open Biology* 3.3 (2013), p. 120179. DOI: 10.1098/rsob.120179..

[Wal86] J B Walsh. "An introduction to stochastic partial differential equations". In: *École d'Été de Probabilités de Saint Flour XIV-1984*. Ed. by R A Carmono. Berlin Heidelberg: Springer, 1986, pp. 265 –439.

[Was09] M R Wasielewski. "Self-assembly strategies for integrating light harvesting and charge separation in artificial photosynthetic systems". In: *Accounts of Chemical Research* 42.12 (2009), pp. 1910 –1921. DOI: 10.1021/ar9001735.

[Was+02] C Wasylyk, S E Schlumberger, P Criqui-Filipe, and B Wasylyk. "Sp100 interacts with ETS-1 and stimulates its transcriptional activity". In: *Molecular and Cellular Biology* 22.8 (2002), pp. 2687 –2702. DOI: 10.1128/MCB.22.8.2687-2702.2002.

[WP+08] S Weidtkamp-Peters, T Lenser, D Negorev, N Gerstner, T G Hofmann, G Schwanitz, C Hoischen, G Maul, P Dittrich, and P Hemmerich. "Dynamics of component exchange at PML nuclear bodies". In: *Journal of Cell Science* 121.16 (2008), pp. 2731 –2743. DOI: 10.1242/jcs.031922.Epub2008Jul29..

[Wes+11] F G Westhorpe, A Tighe, P Lara-Gonzalez, and S S Taylor. "p31 comet-mediated extraction of Mad2 from the MCC promotes efficient mitotic exit". In: *Journal of Cell Science* 124.22 (2011), pp. 3905 –3916. DOI: 10.1242/jcs.093286.

[WC55] C R Wilke and P Chang. "Correlation of diffusion coefficients in dilute solutions". In: *AIChE Journal* 1.2 (1955), pp. 264 –270. DOI: 10.1002/aic.690010222.

[Yeh09] E T H Yeh. "SUMOylation and De-SUMOylation: wrestling with life's processes". In: *Journal of Biological Chemistry* 284.13 (2009), pp. 8223 –8227. DOI: 10.1074/jbc.R800050200.

[YCB80] M E Young, P A Carroad, and R L Bell. "Estimation of diffusion coefficients of proteins". In: *Biotechnology and Bioengineering* 22.5 (1980), pp. 947 –955. DOI: 10.1002/bit.260220504.

[ZT77] O C Zienkiewicz and R L Taylor. *The finite element method*. Vol. 3. London: McGraw-Hill London, 1977.

[ZW05] J S van Zon and P R ten Wolde. "Green's-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space". In: *Journal of Chemical Physics* 123.23 (2005), p. 234910. DOI: 10.1063/1.2137716.

DECLARATION

Declaration with respect to the doctorate procedure.

Hiermit erkläre ich ehrenwörtlich:

- Mir ist die Promotionsordnung der Fakultät für Mathematik und Informatik (vom 8. Juli 2009) bekannt.

- Die Hilfe eines Promotionsberaters bzw. einer Promotionsberaterin wurde nicht in Anspruch genommen. Dritte haben im Zusammenhang mit dem Inhalt dieser Arbeit weder unmittelbar noch mittelbar geldwerte Leistungen erhalten.

- Die Dissertation habe ich selbst angefertigt, wobei keine Textabschnitte oder Ergebnisse eines Dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönliche Mitteilungen und Quellen in meiner Arbeit angebeben wurden sind.

- Die Dissertation (oder Auszüge daraus) wurde noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht.

- Bis jetzt wurde noch keine Dissertation bei einer anderen Hochschule eingereicht.

Bei der Auswahl und Auswertung des Materials sowie bei der Erstellung des Manuskripts wurde ich von folgenden Personen unterstützt: Prof. Dr. Peter Dittrich, Dr. Peter Hemmerich und Dr. Bashar Ibrahim. Der futuristische Support wurde von Prof. Hubert J. Farnsworth geleistet. Für sprachliche und orthographische Aspekte der Arbeit wurde ich von Cilla, Alan und Philipp Wadge unterstützt.

Jena, March 2020

Richard Henze