Extending Regulatory Network Modeling with Multistate Species

Umme Juka Mobassera

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Masters in Computer Science and Applications

Clifford A. Shaffer, Chair
John J. Tyson, Co-Chair
Yang Cao

October 31, 2011
Blacksburg, Virginia

Keywords: Computational Systems Biology, Multistate Species, Rule Based Modeling, SBML, Modeling Tool, Software, JigCell
Copyright 2011, Umme Juka Mobassera
Extending Regulatory Network Modeling with Multistate Species

Umme Juka Mobassera

(ABSTRACT)

By increasing the level of abstraction in the representation of regulatory network models, we can hope to allow modelers to create models that are beyond the threshold of what can currently be expressed reliably. As hundreds of reactions are difficult to understand, maintain, and extend, thousands of reactions become next to impossible without any automation or aid. Using the multistate-species concept we can reduce the number of reactions needed to represent certain systems and thus, lessen the cognitive load on modelers.

A multistate species is an entity with a defined range for state variables, which refers to a group of different forms for a specific species. A multistate reaction involves one or more multistate species and compactly represents a group of similar single reactions. In this work, we have extended JCMB (the JigCell Model Builder) to comply with multistate species and reactions modeling and presented a proposal for enhancing SBML (the Systems Biology Markup Language) standards to support multistate models.
Acknowledgments

I would like to express my appreciation for people who are, directly or indirectly, part of this work.

First of all, I would like to give thanks to Dr. Clifford A. Shaffer, the chair of my committee, for his invaluable guidance all through this work. Specifically, I want to mention about his suggestions in the writing process. He reviewed the drafts thoroughly each time and helped me to correct even the small mistakes. I would like to convey my gratitude to Dr. John Tyson, the co-chair of my committee, for being the caring boss. He managed time for me when required from his busy schedule and made me understand the biological models and the vision of the project. I am grateful to Dr. Yang Cao for the advice and suggestions he gave time to time for the improvement of this work.

Beside my committee, I would like to thank Dr. Debashis Barik for providing me with introductory materials regarding multistate modeling and for helping me to collect the perspectives of modelers for the user interface design. Dr. Kathy Chen helped me to understand the features of the existing software and tools. Whenever I went to her, even for silly problems, she spent from her busy time to explain that to me with patience. Thank you Kathy. I also like to give thanks to all the members of the Tyson Lab for being so supportive colleagues.

I want to mention about the financial support I received for this work from the NIH under Grant R01GM078989.

I also want to take the opportunity here to express my gratitude for my parents and other family members who have supported me with positive inspiration all through my life. I want to mention my grand father who gave me encouragement for pursuing higher studies. And last but not the least, my gratitude to the Bangladeshi students and families in Blacksburg who became a family away from family for me through their endless support.
Dedication

For the most precious gift I ever got - Maheer
# Contents

1 Introduction ......................................................... 1

2 A Small Example of a Multistate System ......................... 4

3 Multistate Modeling: Background and Related Tools ............ 7
   3.1 The SBML Standard and the *multi* package .................. 7
   3.2 Rule-Based Modeling and BioNetGen ............................ 11
   3.3 Other Software .................................................. 13

4 Regulatory Network Modeling with Multistate Reactions ........ 14
   4.1 Multistate Species ............................................... 14
   4.2 Multistate Reaction Format ...................................... 15
   4.3 Rate Laws ........................................................ 16
   4.4 Other Constructs ................................................ 17

5 A Complete Model .................................................. 20
   5.1 Biological Model .................................................. 20
   5.2 The Multistate Model ............................................. 20

6 Multistate Modeling with the JigCell ModelBuilder and SBML .... 24
   6.1 JCMB Spreadsheet ................................................ 25
      6.1.1 Species .................................................... 25
      6.1.2 Rules ...................................................... 26
      6.1.3 Reactions .................................................. 26
      6.1.4 Functions .................................................. 28
      6.1.5 Parameters ................................................ 30
      6.1.6 Equations .................................................. 30
   6.2 Multistate Modeling in SBML .................................... 33
   6.3 Model Flattening ................................................. 36
   6.4 Compliance with Other Tools and Simulators ................. 38

7 Conclusions and Future Work ..................................... 42
## List of Figures

1.1 Modeling of multistate reactions with JigCell and SBML .................................. 2  
2.1 The Clb2-Cdh1 Antagonism Model ........................................................................ 4  
2.2 List of reactions in the Antagonism Model using single state species and reactions 5  
2.3 Reactions in the compact multistate format in the Antagonism Model ............... 6  
2.4 The variable $X_a$ is assigned the expression for activity of Cdh1 in the Antagonism Model ........................................................................................................... 6  
3.1 The SBML $<$Model$>$ and $<$Species$>$ constructs in the multi extension ............ 7  
3.2 The SBML $<$SpeciesType$>$ construct in the multi extension ............................... 8  
3.3 The SBML $<$Selector$>$ and $<$SpeciesTypeState$>$ constructs in the multi extension .............................................................................................................................. 9  
3.4 The SBML $<$SimpleSpeciesReference$>$ construct in the multi extension ............ 10  
3.5 The Antagonism Model expressed in BNGL .......................................................... 12  
5.1 The Cell Cycle Control Model of the budding yeast [1] ....................................... 21  
5.2 Multistate reactions in the Cell Cycle Control Model of the budding yeast (part1) ......................................................................................................................... 22  
5.3 Multistate reactions in the Cell Cycle Control Model of the budding yeast (part2) ......................................................................................................................... 23  
6.1 Species spreadsheet in JCMB .................................................................................. 25  
6.2 A multistate species creation dialog box in JCMB ............................................... 25  
6.3 Rule spreadsheet in JCMB showing use of the $\text{sum}()$ operator ..................... 26  
6.4 Reaction spreadsheet in JCMB: Reaction, Type, Equation, and Expanded Reactions columns for defining multistate reactions ................................................. 26  
6.5 Reaction entry dialog box ....................................................................................... 27  
6.6 An accepted multistate reaction in the Reaction column with undefined range for indices ....................................................................................................................... 27  
6.7 Multistate reaction parameter editor dialog box .................................................. 28  
6.8 Equation column in the Reaction spreadsheet ..................................................... 29  
6.9 Expanded Reactions dialog box showing multistate reactions in the Antagonism Model .................................................................................................................. 29  
6.10 Highlighting columns with red or yellow for guiding in building the model ........ 30
6.11 Functions spreadsheet ................................. 30
6.12 Parameters spreadsheet showing parameters used in the Antagonism Model ................................. 31
6.13 Equations (ODEs) shown in multistate format .................................................. 31
6.14 Modification proposed for the <StateFeature> constructs of the SBML multi package (A) StateFeature (B) StateFeatureInstance .................................................. 32
6.15 Modification proposed for the <Reaction> construct of the SBML for multi package .................................................. 34
6.16 Multistate species definition expressed in SBML using annotation (the Antagonism Model) .................................................. 34
6.17 Multistate reaction definition expressed in SBML using annotation (the Antagonism Model) .................................................. 35
6.18 Summation rule expressed in SBML using MathML and annotation (the Antagonism Model) .................................................. 36
6.19 The flattening process for a multistate model into a standard SBML model .................................................. 37
6.20 List of species after flattening (the Antagonism Model) .................................................. 39
6.21 List of rules after flattening (the Antagonism Model) .................................................. 39
6.22 List of ODEs after conversion (the Antagonism Model) .................................................. 39
6.23 Reactions of the Antagonism Model shown in the COPASI reaction window .................................................. 40
6.24 Deterministic time course simulation of some key regulatory species (Cdh1, ClbM, ClbS, and Cdc14) of the Cell Cycle Control Model of the budding yeast. (a) Plot taken from Figure 3 of [1] (b) Plot generated from the flattened model 41
List of Tables

4.1 Multistate reaction types ......................................................... 15
4.2 Rate law equations for multistate phosphorylation, dephosphorylation, degradation and complexation reactions, with state-independent rate constants . . 18
5.1 Multistate species list in the Cell Cycle Control Model of the budding yeast . 21
Chapter 1

Introduction

The physiology of a living cell is governed by underlying networks of interacting macromolecules such as genes, mRNAs and proteins. Modelers of biochemical systems often work with hundreds of reactions representing the interactions of dozens of different species and parameters. Many tools have been developed to make the task of modeling and editing faster and less error prone. These tools allow modelers to create models with hundreds of reactions, which is beyond their ability to construct by hand. By varying the level of abstraction, we hope that such models can be expressed in ways that preserve the information, while reducing cognitive load on the modeler.

A common theme in intra-cellular regulatory networks is multi-state species, such as, a protein with multiple states. A multistate species can change its binding and reacting properties based on changes in one or many of its binding sites by phosphorylation or methylation, for example. Multi-state species are frequently present in biological systems, so modeling multistate species properly is important for those biochemical models.

A multistate species with $n$ distinct binding sites can have $2^n$ different states if every combination of bindings is a separate state, so, a species with 5 binding sites may exhibit $2^5 = 32$ different binding combinations. Depending on the number of sites and the number of states or values these binding sites can take, the possible number of species' states can become very large and impractical to represent explicitly. Additional complexity results when multiple multistate species in a model combine in many different ways to make complexes. Defining and maintaining such a large system gives the modelers more work and increases the complexity of the resulting simulation. For example, in the tyrosine kinase receptor EGFR multiple (atleast nine) tyrosines are phosphorylated during signaling [13]. As a result, there are possibly $2^9$ or 512 different forms of EGFR and more than 100K distinct combinations or dimers. Ofcourse the actual number of relevant phospho-forms of EGFR is much lower, however, the whole set of forms may create interest for researchers.

A different issue comes into the picture when some states appear often in reactions and some do not appear at all. This might happen when some of the possible phosphorylation sites of a multistate species are not used. Obviously modelers would like to get a way to effectively represent the used states when required with less labor and time. This indicates
the necessity for defining the species compactly but using enumeration for referring to a set of states when needed.

Modeling a system is not a one-time task. Models require frequent editing, and ideally will be reused in the future. The bigger that models become, the more complex and difficult they are to view, handle and use. By managing as much of the bookkeeping as possible, automated tools can make modeling easier and faster for modelers.

![Diagram](image)

Figure 1.1: Modeling of multistate reactions with JigCell and SBML

JigCell [12] is a collection of software tools for modeling biochemical networks. We seek to enhance the JigCell ModelBuilder (JCMB) to support the multistate modeling aspect of the SBML [22] standard. The Systems Biology Markup Language or SBML is a XML-based format for storing and representing biochemical models. SBML comprises many different constructs or structures to represent different biological elements, and it allows storing and communicating among computational models of biological processes with free, open, and widespread software support and a community of users and developers.

We have added to JCMB a convenient way to define multistate species and their interactions, and tested our efforts with real examples from the published literature, with the expectation to have an insight into the core modeling mechanism (Figure 1.1). We have revised the existing SBML proposals for multistate species modeling which we amend to support a more general representation. In this work, we consider four aspects of multistate species and reaction modeling:

- Study and analyze real biological models involving multistate reactions
- Standardize the syntax for expressing and storing these models
- Check the consistency and integrity of the multistate part in overall modeling
- Create a software platform for multistate models and propose a way to reuse them in existing standard tools

In Chapter 2, multistate species and reactions are presented using a small model to show the impact and advantages of multistate modeling. In Chapter 3, we analyze the literature, standards and tools related to multistate modeling. Chapter 4 presents the syntax for defining multistate species and reactions. Chapter 5 gives an example of multistate modeling with a full model. We identify and discuss the multistate features of the model and how we can practically represent them in a compact way. In Chapter 6, the JCMB environment and
SBML standards for multistate modeling are discussed. Chapter 7 discusses future plans, describing limitations of the current work and ways to overcome those.
Chapter 2

A Small Example of a Multistate System

In this chapter a small model (part of the model discussed in [1]) is presented. It illustrates the multistate reaction concept and our modeling approach. Figure 2.1 shows the core reactions of the model, representing mutual antagonism between Clb2 and Cdh1. In this model, dependence of Cdh1 on the ratio of Clb2 kinase vs Cdc14 phosphatase activity is crucial for the phosphorylation stages.

![Figure 2.1: The Clb2-Cdh1 Antagonism Model](image)

In this model, we have a total of 11 forms or states for Cdh1: a non-phosphorylated form (Cdh1) and 10 phosphorylated forms denoted as Cdh1P_1, Cdh1P_2, ..., Cdh1P_{10}. These forms take part in the phosphorylation and dephosphorylation reactions as reactant or product. All of the eleven forms of Cdh1 take part in the degradation of Clb2. In this model there are 34 single-state reactions in total if each phosphorylation state is considered as a distinct species (Figure 2.2).

We can define Cdh1 as a multistate species with a state range represented by indices, such as, Cdh1P_i with 0 ≤ i ≤ 10. Using this compact notation, we can reduce the number of
<table>
<thead>
<tr>
<th>SL</th>
<th>Reactions</th>
<th>Rate law</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \rightarrow \text{Cib2} )</td>
<td>Mass action</td>
<td>( k_a )</td>
</tr>
<tr>
<td>2</td>
<td>( \text{Cib2} \rightarrow )</td>
<td>Mass action</td>
<td>( \text{Cib2.Xa} )</td>
</tr>
<tr>
<td>3</td>
<td>( \text{Cdh1} + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_1 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1} )</td>
</tr>
<tr>
<td>4</td>
<td>( \text{Cdh1P}_1 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_2 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_1 )</td>
</tr>
<tr>
<td>5</td>
<td>( \text{Cdh1P}_2 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_3 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_2 )</td>
</tr>
<tr>
<td>6</td>
<td>( \text{Cdh1P}_3 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_4 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_3 )</td>
</tr>
<tr>
<td>7</td>
<td>( \text{Cdh1P}_4 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_5 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_4 )</td>
</tr>
<tr>
<td>8</td>
<td>( \text{Cdh1P}_5 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_6 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_5 )</td>
</tr>
<tr>
<td>9</td>
<td>( \text{Cdh1P}_6 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_7 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_6 )</td>
</tr>
<tr>
<td>10</td>
<td>( \text{Cdh1P}_7 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_8 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_7 )</td>
</tr>
<tr>
<td>11</td>
<td>( \text{Cdh1P}_8 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}_9 )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_8 )</td>
</tr>
<tr>
<td>12</td>
<td>( \text{Cdh1P}<em>9 + \text{Cib2} \rightarrow \text{Cib2} + \text{Cdh1P}</em>{10} )</td>
<td>Mass action</td>
<td>( k_p.\text{Cib2.Cdh1P}_9 )</td>
</tr>
<tr>
<td>13</td>
<td>( \text{Cdh1P}_{10} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1} )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{10} )</td>
</tr>
<tr>
<td>14</td>
<td>( \text{Cdh1P}_{11} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_1 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{11} )</td>
</tr>
<tr>
<td>15</td>
<td>( \text{Cdh1P}_{12} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_2 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{12} )</td>
</tr>
<tr>
<td>16</td>
<td>( \text{Cdh1P}_{13} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_3 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{13} )</td>
</tr>
<tr>
<td>17</td>
<td>( \text{Cdh1P}_{14} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_4 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{14} )</td>
</tr>
<tr>
<td>18</td>
<td>( \text{Cdh1P}_{15} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_5 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{15} )</td>
</tr>
<tr>
<td>19</td>
<td>( \text{Cdh1P}_{16} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_6 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{16} )</td>
</tr>
<tr>
<td>20</td>
<td>( \text{Cdh1P}_{17} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_7 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{17} )</td>
</tr>
<tr>
<td>21</td>
<td>( \text{Cdh1P}_{18} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_8 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{18} )</td>
</tr>
<tr>
<td>22</td>
<td>( \text{Cdh1P}_{19} + \text{Cdc14} \rightarrow \text{Cdc14} + \text{Cdh1P}_9 )</td>
<td>Mass action</td>
<td>( k_b.\text{Cdc14.Cdh1P}_{19} )</td>
</tr>
<tr>
<td>23</td>
<td>( \text{Cdh1} \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1} )</td>
</tr>
<tr>
<td>24</td>
<td>( \text{Cdh1P}_1 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_1 )</td>
</tr>
<tr>
<td>25</td>
<td>( \text{Cdh1P}_2 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_2 )</td>
</tr>
<tr>
<td>26</td>
<td>( \text{Cdh1P}_3 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_3 )</td>
</tr>
<tr>
<td>27</td>
<td>( \text{Cdh1P}_4 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_4 )</td>
</tr>
<tr>
<td>28</td>
<td>( \text{Cdh1P}_5 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_5 )</td>
</tr>
<tr>
<td>29</td>
<td>( \text{Cdh1P}_6 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_6 )</td>
</tr>
<tr>
<td>30</td>
<td>( \text{Cdh1P}_7 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_7 )</td>
</tr>
<tr>
<td>31</td>
<td>( \text{Cdh1P}_8 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_8 )</td>
</tr>
<tr>
<td>32</td>
<td>( \text{Cdh1P}_9 \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_9 )</td>
</tr>
<tr>
<td>33</td>
<td>( \text{Cdh1P}_{10} \rightarrow )</td>
<td>Mass action</td>
<td>( k_d.\text{Cdh1P}_{10} )</td>
</tr>
<tr>
<td>34</td>
<td>( \rightarrow \text{Cdh1} )</td>
<td>Mass action</td>
<td>( k_n )</td>
</tr>
</tbody>
</table>

Figure 2.2: List of reactions in the *Antagonism Model* using single state species and reactions
reactions to 5 by expressing the reactants and products as multistate species (Figure 2.3). In the compact format, Reaction 3 represents original reactions 3-12 (phosphorylation of Cdh1), Reaction 4 represents original reactions 13-22 (dephosphorylation of Cdh1), and Reaction 5 represents original reactions 23-33 (degradation of Cdh1). Reaction 6 represents the synthesis of the unphosphorylated form of Cdh1 (original reaction 34).

![Table 2.1: Reactions in the compact multistate format in the Antagonism Model](image)

**Figure 2.3:** Reactions in the compact multistate format in the *Antagonism Model*

In the model (Reaction 2 in Figure 2.2), $X_a$ is the activity of Cdh1 expressed as the weighted sum of its phosphorylated and un-phosphorylated forms. Figure 2.4 shows the expression assigned to $X_a$. We need a way to represent this sort of expression in the multistate format. Equation 2.1 shows how we can use the summation operator to make the assignment expression easy to view, manipulate and edit. In Chapters 4 and 6, we discuss the representation of the $\text{sum()}$ operator in SBML and how it is used that in JCMB, respectively.

$$X_a = k_a * Cdh1 + k_i * \sum_{i=1}^{10} Cdh1P_i \quad (2.1)$$

At present, there is no standard mechanism or tool that uses the multistate format directly, for example in simulation. SBML also does not provide a way to express multistate reactions that is acceptable and usable by all related tools. JCMB therefore provides a translator to expand the compact (multistate) model into a single-state species model for use with other SBML-compliant tools.

**Figure 2.4:** The variable $X_a$ is assigned the expression for activity of Cdh1 in the *Antagonism Model*
Chapter 3

Multistate Modeling: Background and Related Tools

In this chapter we survey the literature for prior efforts to represent multistate reactions.

3.1 The SBML Standard and the *multi* package

The SBML Level 3 core specification [23] is based on a modular approach that allows features and proposals supplemented to the core to be supported as additional packages. In SBML Level 3, multistate and multi-compartment species are described using the package *multi*[24] which, at present, is in the proposal phase. Multi also supports creation of complexes made up of different components.

![Figure 3.1: The SBML <Model> and <Species> constructs in the *multi* extension](image)

The proposed SBML level-3 *multi* package, comprising of new *<SpeciesType>* and *<Selector>* constructs, is supposed to capture and represent the multistate modeling concept. Figure 3.1, 3.2, and 3.3 show the major constructs and modifications proposed in this
package. In addition, multi has proposed modifications to some existing constructs of the SBML core, such as, <Species>, <SimpleSpeciesReference>, <Rule>, <Reaction> etc.

A <SpeciesType> can represent a number of state features or sites under one list of state features. Each <StateFeature> is identified by a $Sid$ and an optional name, and contains a list of <PossibleValue>. The values are possibly strings expressing the values or conditions of the state (for example, possible values of a state can be something like bound, free, or closed). A <SpeciesType> also has a boolean attribute <bindingSite> to describe multi-compartment entities.

A <Selector> is a mask describing the rules that a species or any other entity has to pass in order to be used or rejected in a set of reactions. A selector contains one or more <SpeciesTypeState>. A <SpeciesTypeState> contains optional links to a list of <StateFeatureInstances> and a list of <ContainedSpeciesTypes>. From Figure 3.3 we can see that each <StateFeatureInstances> contains a list of allowed possible values as <StateFeatureValue>.

Another important construct in the SBML core is the <SimpleSpeciesReference> which contains a <SIdRef> referring to the corresponding species. The <Reaction> construct represents the reactants and products using <SimpleSpeciesReference>. In multistate and multi-compartment modeling, a species reference instance needs to know the set of species instances it should contain, based on certain restriction and selection criteria. To get this additional effect, in multi the simple species reference construct has been modified to carry a link to the list of <SpeciesTypeRestriction>, which eventually points to an instance of the
Figure 3.3: The SBML <Selector> and <SpeciesTypeState> constructs in the multi extension
Figure 3.4: The SBML <SimpleSpeciesReference> construct in the multi extension

<SpeciesType> (Figure 3.4). Similarly, according to this proposal, the <Rule> construct contains an additional link to one <SpeciesTypeInstance>.

Other modifications proposed in multi include the <Bond>, <BindingSiteReference> and <ReactionRule>. Most of the remaining proposal supports multi-compartment models. In this work we mainly focus on multistate modeling. So we do not discuss those constructs here.

The multistate modeling constructs in multi enable the model to express different state features and binding sites for a species and its related reactions, but the states should be explicitly defined. Multi does not show a way to effectively represent a species list with enumeration. As we saw in Chapter 2, the ability to make state variables with a range is important. In Chapter 6 we discuss in detail how it is possible to modify the proposed standard so that it can include the ordering information of states and also can comply with the existing requirements.

For storing the properties and parameters of a multistate model in standard SBML format, some additions and modifications are needed in the proposed multi package. Since these features are not part of the current SBML standard, we can use SBML <annotations> to convey this information. According to the SBML definition, an annotation can be part of any <SBase> element (for example, <Species> in SBML Level 2). So, it is desirable to define multistate species as a special type of <Species> with state variables included in annotations. Similarly, other related constructs like <Reaction>, <SpeciesReference>, <Rule> etc. should be modified to include state information. Multi-dimensional arrays can be a suitable structure to represent the multistate species concept, if implemented with SBML standards.
3.2 Rule-Based Modeling and BioNetGen

In the rule-based modeling (RBM), the molecular interactions in a system are defined as rules. Rules provide conditions and outcomes for reactions and rate laws to guide the transformations. A rule either can express one simple chemical reaction or can define a large class of reactions, involving a common transformation rate law. Thus using rules it is possible to generate a list of reactions.


In BNGL a model is constructed using some building blocks such as parameters, molecules (optional), species or seed species, reaction rules, observables etc [5]. BNGL allows patterns in the rules block to generate reaction rules using the set of species defined in the species block. For example, a species $XP$ can be defined as $XP(bs1)$ where $bs1$ is the binding site for the species. Subsequently, it is possible to use $XP(bs1)$ for referring to the unbound forms, $XP(bs1!+)$ for referring to the bound forms, and $XP()$ for referring to all forms of $XP$ including the complex forms. BNGL can generate definition of complex forms created using the pattern and reaction rules, even if that is not defined explicitly in the species block.

RuleBender [21] is a tool for creating, debugging, simulating and analyzing rule-based biological models. It is a free tool with an integrated development environment for creating models in BNGL. BioNetGen simulator commands can be applied to the models loaded or created in the RuleBender. We used the RuleBender interface to represent the small model of Chapter 2 in the BioNetGen language. Figure 3.5 shows BNGL code for the antagonism model.

In BNGL a multistate species $XP$ should be represented as $XP(site1 \sim value1 \sim value2 \sim ... \sim valueN, site2, ...)$ where each site is defined with explicit list of state values (separated by a $\sim$). In Figure 3.5 we see the multistate species Cdh1 (which has one site for phosphorylation with 11 possible states) is defined as Cdh1($P \sim 0 \sim 1 \sim 2 \sim 3 \sim 4 \sim 5 \sim 6 \sim 7 \sim 8 \sim 9 \sim 10$). Although BNGL supports the use of patterns and mappings to avoid explicit description of all the states of the bindings that are involved in a reaction, it requires iterative definition of rules (for phosphorylation and dephosphorylation reactions) for each state of the multistate species (Cdh1) for generating our example multistate model. Because, BNGL can not automatically interpret the ordering (for example, next state or previous state) between the different states of a binding site from the species definition and can not recognize the relative order of states of the species in the reaction unless we define the reaction rules explicitly for each state of the multistate species. But such an ordering is required commonly in multistate reactions. Hence, modelers seek a platform to create multistate models with
Figure 3.5: The Antagonism Model expressed in BNGL
3.3 Other Software

There are other tools available for modeling and simulation of similar type of reaction networks. For example, Virtual Cell (a web-based computational environment for modeling and simulation of cell biology models) and NFSim (a simulation platform that simulates straightforward BNGL specification of models) are two such tools. Both the tools are based on BioNetGen and SBML standards, and lack the features of multistate modeling.

COPASI is a tool for modeling, simulation, and analysis of biochemical networks and their dynamics. COPASI, based on SBML standards and its own internal structure, also lacks features to support multi-state modeling. The JigCell and COPASI teams are working to develop an integrated modeling and simulation platform that includes the multistate modeling concept.
Chapter 4

Regulatory Network Modeling with Multistate Reactions

In this chapter, proposals for representing multistate species, reactions, and rules are presented.

4.1 Multistate Species

A multistate species has one or more multi-valued sites or features. Such a set of features can be represented with state or index variables. For example \( X\{i,j\} \) represents a multistate species with 2 index variables (sites) \( i \) and \( j \), each of which may have multiple values defined by specific lower and upper limits. In the model discussed in Chapter 2, Cdh1 is a multistate species that could be represented as CdhP\{i\} with one index variable, \( i \).

The range of values for each variable of a multistate species may be different in different reactions but they must all fall within the pre-defined range of the species index. In expanded form, every possible combination of states could be represented by a separate species, if that particular state is expressed or used in any of the reactions. For example, \( X\{i,j\} \) with \( 0 \leq i \leq 5 \) and \( 0 \leq j \leq 2 \) could be represented by a total of \( 6 \times 3 \) single state species. Reaction parameters such as rate constants may depend on the value of the state variables.

We present a consistent syntax with which to represent multistate species. To keep similarity with standard single state species naming syntax, a pair of curly braces \{\} is used with comma separated index variables inside. For example \( Y\{0:5\} \) is a multistate species with 1 site, representing a set of six variants of the species. In contrast, \( Y\{0\} \) is a single state species. \( Y\{0:2,0:5\} \) represents a species with 2 sites, the first with three states and the second with six states. \( Y\{index1\} \) represents a species with one site referred to as \( index1 \) and the range for \( index1 \) (perhaps, \( 0 \leq index1 \leq 10 \)) needs to be defined separately.
### Table 4.1: Multistate reaction types

<table>
<thead>
<tr>
<th>Index</th>
<th>Reaction Type</th>
<th>Reaction Format</th>
<th>Rate Law Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phosphorylation</td>
<td>( XP{i} \rightarrow XP{i + 1} ) or ( XP{i} + E \rightarrow E + XP{i + 1} )</td>
<td>( k_i \cdot XP{i} ) or ( k'_i \cdot E \cdot XP{i} )</td>
</tr>
<tr>
<td>2</td>
<td>Dephosphorylation</td>
<td>( XP{i + 1} \rightarrow XP{i} ) or ( XP{i + 1} + E \rightarrow E + XP{i} )</td>
<td>( k_i \cdot XP{i} ) or ( k'_i \cdot E \cdot XP{i} )</td>
</tr>
<tr>
<td>3</td>
<td>Degradation</td>
<td>( XP{i} \rightarrow . ) or ( XP{i} + E \rightarrow E )</td>
<td>( k_i \cdot XP{i} ) or ( k'_i \cdot E \cdot XP{i} )</td>
</tr>
<tr>
<td>4</td>
<td>Complex Binding 1</td>
<td>a) ( XP{i} + Y \rightarrow C{i} ) b) ( XP{i} + YP{j} \rightarrow XY{i,j} )</td>
<td>a) ( k_i \cdot XP{i} \cdot Y ) b) ( k_i \cdot XP{i} \cdot YP{j} )</td>
</tr>
<tr>
<td>5</td>
<td>Complex Binding 2</td>
<td>a) ( C{i} \rightarrow XP{i} + Y ) b) ( XY{i,j} \rightarrow XP{i} + YP{j} )</td>
<td>( k_i \cdot C{i} )</td>
</tr>
<tr>
<td>6</td>
<td>Degradation of complex form 1</td>
<td>a) ( C{i} \rightarrow Y; degrades XP{i} ) b) ( C{i} \rightarrow XP{i}; degrades Y (with C{i} a complex of Y and XP{i}) )</td>
<td>( k_i \cdot C{i} )</td>
</tr>
<tr>
<td>7</td>
<td>Degradation of complex form 2</td>
<td>a) ( XY{i,j} \rightarrow YP{j}; degrades XP{i} ) b) ( XY{i,j} \rightarrow XP{i}; degrades YP{j} (with XY{i,j} a complex of XP{i} and YP{j}) )</td>
<td>( k_{ij} \cdot XY{i,j} )</td>
</tr>
</tbody>
</table>

### 4.2 Multistate Reaction Format

Multistate reactions are similar to other reactions but have multistate species as reactants and/or products. We want to express appropriate sets of multistate reactions as a single reaction. The compact structure reduces the cognitive load on the modelers to a meaningful extent for large models. The other concern here is to keep the integrity of the model so that it can be easily converted for use in other tools that do not recognize our multistate representation.

Reactions with multistate properties are involved in a number of different behaviors. For example, some reactions result in modification of proteins or formation of heterogeneous protein complexes with enzymes and substrates. Others involve the targeted degradation of protein complexes. Understanding the nature of these reactions is important for modeling the dynamics of protein-protein interaction networks. These interactions are seen at different levels of protein sites with binding and catalytic activities. Some sample reactions have been listed in Table 4.1.

Based on the number of phosphorylation or dephosphorylation events that occur due to the binding of an enzyme to the target protein, phosphorylation or dephosphorylation reactions can be categorized as either processive or distributive. A phosphorylation (or
dephosphorylation) reaction is called processive if all or a specific number of the available sites of a substrate protein get phosphorylated (or dephosphorylated) when a kinase (or phosphatase) binds to it, with the effect being done in a single binding event (i.e., before the kinase or phosphatase gets dissociated from the binding site). But in a distributive reaction at most one phosphorylation or dephosphorylation event occurs on a single binding of the enzyme (i.e., kinase or phosphatase) to the substrate.

Phosphorylation and dephosphorylation reactions can be dependent on (or independent of) how many sites are already phosphorylated. The mechanism may depend on the sequence of sites involved in phosphorylation or dephosphorylation reactions. For the ordered case, the reaction events involve the sites in a specific sequence (i.e., the sites are phosphorylated in a specific sequence and dephosphorylated in the reverse sequence) and the rate of the reactions does not depend on the number of free or occupied sites. For the disordered case, the reactions can occur at arbitrary sites and the rate of phosphorylation or dephosphorylation is proportional to the number of free or occupied sites.

When the order of the species states is important for a series of reactions, it is required to express relations such as predecessor or successor among reactants and products. For example, \( XP\{i\} \to XP\{i - 1\} \) or \( XP\{i\} \to XP\{i + 1\} \). Here direct reference to previous or next state has been used where \( i \) gives the starting state value. This fashion of expression is especially useful where individual state identity is less significant. For some reactions it might be required to refer to a specific set of states (such as \( XP\{[0, 1, 5]\} \)) or a set of states defined with a function (such as, the odd states for \( XP\{0 : 11\} \) can be defined as \( XP\{2 \ast i + 1\} \) with a defined range for the value of \( i \)). Arithmetic expressions can be used to represent such relations with assigned numerical values for states or a defined range of numerical values for the states as a whole.

The syntax for multistate reactions should include these reaction parameters:
- A range for each site of each multistate species involved in the reaction
- Reaction Mechanism
  - Progressive or distributive
  - Ordered or disordered (except degradation and complexation reactions)
- Rate law and related parameters
  - State dependent or independent parameters
  - (If dependent) specific function (i.e., stepwise, exponential, or custom)

### 4.3 Rate Laws

Rate laws play an important role in the mechanism of a reaction. Rate law equation and parameters can impact significantly the nature of the multistate reaction. Some rate law equations for multistate reactions are shown in Table 4.2. A model editor should allow users to add customized rate law equations, or to choose from common ones. We have mainly focused on distributive models, specifically models with distributive and multistate phosphorylation/dephosphorylation reactions. We have studied and implemented details for
two known rate laws (Mass Action and Michaelis-Menten) and provided option for defining local rate law for the reaction (Local).

We have noticed that the reaction mechanism (ordered or disordered) has an impact on phosphorylation and dephosphorylation reactions, but not on synthesis, degradation or complexation reactions. It is expected that multistate modeling software will be intelligent enough to distinguish among reactions, for example, to distinguish multistate phosphorylation and dephosphorylation (or ubiquitylation and deubiquitylation) reactions from multistate reactions like degradation or complex formation. Based on the reaction type, different reaction mechanisms can be applied.

Another important parameter in multistate modeling is the reaction rate constant. If the rate constant (say $k$) is state independent then it is assigned one value for all the reactions involving the species states. When the rate constant depends on the state of the system, it is assigned values following some function. For example:

- **Stepwise**: The value of the rate constant is different for each state. Then we need to define a set of rate constants, one for each state. We write it as $k\{i\}$.
- **Exponential**: For the exponential case, the rate constant of subsequent states depends exponentially on the first step, for example: $k\{i\} = k_0r^i$.
- **Local Function**: The custom function allows a user to create any function required by the model. For example, $k\{i\} = kp(1 + k_0r^{i+1})$ can be a custom function for the rate constant.

Table 4.2 shows some rate law equations used for multistate reactions using $k_p$, $k_h$, $k_d$, and $k_c$ as rate constants.

### 4.4 Other Constructs

There may be parameters or rules that involve specific states of a multistate species, defined with a range of one or more index variables. For example, the activity of the active or inactive forms of $XP$ (a multistate species which has one unphosphorylated and some phosphorylated states) can be expressed like Equation 4.1 using the summation of concentration of involved states with some weights. In this equation, $a_i$ is the weight for active species, and $L$, $M$ and $N$ are numerical variables to define range of index, $i$.

$$A_{active} = \sum_{i=L}^{M} a_i \cdot XP_i; 0 \leq L \leq M \leq N \quad (4.1)$$

A mathematical implementation of the summation operator is required to represent such expressions. Other constructs required for multistate modeling are *Array* and *Set*. An array is a structure to represent a group of objects in a defined order. It can be used to represent the range of indices of a multistate species. A set is a collection of distinct objects. We can use this construct to define the list of some species states defined by a specific function as described in Section 4.3. As we are using SBML ultimately to store models and SBML follows
Table 4.2: Rate law equations for multistate phosphorylation, dephosphorylation, degradation and complexation reactions, with state-independent rate constants

<table>
<thead>
<tr>
<th>Index</th>
<th>Reaction Example</th>
<th>Reaction Type</th>
<th>Reaction Mechanism</th>
<th>Rate Law</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$XP{i} + E \rightarrow XP{i+1} + E$ $0 \leq i \leq N-1$</td>
<td>Phosphorylation</td>
<td>Ordered</td>
<td>Mass Action</td>
<td>$k_p * E * XP{i}$</td>
</tr>
<tr>
<td>2</td>
<td>$XP{i} + E \rightarrow XP{i-1} + E$ $1 \leq i \leq N$</td>
<td>Dephosphorylation</td>
<td>Ordered</td>
<td>Mass Action</td>
<td>$k_h * E * XP{i}$</td>
</tr>
<tr>
<td>3</td>
<td>$XP{i} + E \rightarrow E$ $0 \leq i \leq N$</td>
<td>Degradation</td>
<td>N/A</td>
<td>Mass Action</td>
<td>$k_d * E * XP{i}$</td>
</tr>
<tr>
<td>4</td>
<td>$XP{i} + Y \rightarrow C{i}$ $0 \leq i \leq N$</td>
<td>Complexation Type 1</td>
<td>N/A</td>
<td>Mass Action</td>
<td>$k_c * XP{i} * Y$</td>
</tr>
<tr>
<td>5</td>
<td>$XP{i} + E \rightarrow XP{i}$ $0 \leq i \leq N$</td>
<td>Complexation Type 2</td>
<td>N/A</td>
<td>Mass Action</td>
<td>$k_c * XP{i} * YP{j}$</td>
</tr>
</tbody>
</table>
MathML constructs for mathematical operators, we have implemented these operators and constructs using MathML in JCMB.

MathML supports two forms of summation:

1. $\sum_{i=i_{\text{min}}}^{i_{\text{max}}} f(X_i)$. Here summation operands are lowlimit $i_{\text{min}}$, uplimit $i_{\text{max}}$ and a function of $X_i$.

2. $\sum_{i \in B} f(X_i)$. Here summation operands are a set $B$ and a function of $X_i$; the summation operation is carried out on $f(X_i)$ over the selected values of $i$ as defined in $B$.

So, it is possible to use the summation operator either over a range of indices or a set of indices as required by the model.
Chapter 5

A Complete Model

Any realistic model for the molecular mechanism of the budding yeast cell-cycle control system naturally includes multistate species and reactions. In this chapter, a version of the model of Barik et al [1]. is discussed in terms of its multistate features.

5.1 Biological Model

Figure 5.1 shows a schematic diagram of Barik’s complete model, a part of which appeared as the small model in Chapter 2. In Figure 5.1, solid arrows represent chemical reactions, dotted arrows represent enzymatic activity, dashed arrows represent multistate phosphorylation chains, and T-shaped arrows with balls on the cross bar indicate reversible binding reactions. This diagram shows the major components of the model. To simplify the view, the synthesis and degradation reactions of Whi5, SBF, Cdh1, Net1, Hbf, Hi5 and Ht1 are not shown here. We assume that the synthesis and degradation reactions follow simple birth-death processes. The diagram includes synthesis and degradation of MbS, which is the mRNA for ClbS, the only regulated mRNA species in the model. Hi5, Hbf and Ht1 are three unregulated phosphatases, which oppose cyclin-dependent kinases on the Whi5, SBF and Net1 phosphorylation chains, respectively. The number of molecules of ClbM plays an important role in the model by triggering the event of cell division. At a certain threshold of ClbM the cell divides, so cell volume becomes half of the previous volume and a new cell cycle begins.

5.2 The Multistate Model

In the schematic diagram (Figure 5.1), there are six multistate species (see Table 5.1) and 17 non-multistate or single species. All of the multistate species are one dimensional (i.e., consisting of one phosphorylation sequence), though the number of possible phosphorylation states differs from species to species.

In Figure 5.1 the synthesis and degradation of the four proteins (Net, Cdh1, SBF and
Figure 5.1: The *Cell Cycle Control Model* of the budding yeast [1]

Table 5.1: Multistate species list in the *Cell Cycle Control Model* of the budding yeast

<table>
<thead>
<tr>
<th>Index</th>
<th>Species Name</th>
<th>Phosphorylation States</th>
<th>Representation as Multistate Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sbf</td>
<td>5</td>
<td>sbf[0:4]</td>
</tr>
<tr>
<td>2</td>
<td>whi5</td>
<td>7</td>
<td>whi5[0:6]</td>
</tr>
<tr>
<td>3</td>
<td>cmp</td>
<td>3</td>
<td>cmp[0:2]</td>
</tr>
<tr>
<td>4</td>
<td>cdh1</td>
<td>11</td>
<td>cdh1[0:10]</td>
</tr>
<tr>
<td>5</td>
<td>net1</td>
<td>9</td>
<td>net1[0:8]</td>
</tr>
<tr>
<td>6</td>
<td>rnt</td>
<td>6</td>
<td>rnt[0:5]</td>
</tr>
</tbody>
</table>
Whi5) has not been shown. Also the details of the creation and destruction of the complexes they form (i.e., Cmp and RNT) are missing. There are a total of 220 single-state reactions in this model, including all the synthesis and degradation reactions missing in the diagram as mentioned above. Using multistate reactions, we can reduce that to 73 (see Figures 5.2 and 5.3).

<table>
<thead>
<tr>
<th>Name</th>
<th>Reaction</th>
<th>Type</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-&gt; dbm</td>
<td>Local</td>
<td>kbmb * (cell/PI) * mbm</td>
</tr>
<tr>
<td>2</td>
<td>cdh1[0] + dbm -&gt; cdh1[0]</td>
<td>Local</td>
<td>kbmbna * cdh1[0] * dbm / cell</td>
</tr>
<tr>
<td>3</td>
<td>dbm -&gt;</td>
<td>Local</td>
<td>kbmbni * cdh1P * dbm / cell</td>
</tr>
<tr>
<td>4</td>
<td>-&gt; cdh1[0]</td>
<td>Multistate</td>
<td>gdmn * dbm</td>
</tr>
<tr>
<td>5</td>
<td>cdh1[0] -&gt; cdh1[i+1]; 1 &lt;= i &lt;= 10</td>
<td>Multistate - Ordered - Mass Action</td>
<td>kdh1P * cdh1[0]</td>
</tr>
<tr>
<td>6</td>
<td>cdh1[0] + cdc14 -&gt; cdh1[i-1] + cdc14; 1 &lt;= i &lt;= 10</td>
<td>Multistate - Local</td>
<td>kdh1 * cdh1[0] + cdc14 / cell</td>
</tr>
<tr>
<td>7</td>
<td>cdh1[0] -&gt; ; 0 &lt;= i &lt;= 10</td>
<td>Multistate - Mass Action</td>
<td>gdh1 * cdh1[0]</td>
</tr>
<tr>
<td>8</td>
<td>cdh1[0] + cdc14 -&gt; cdh1[i+1] + cdc14; 0 &lt;= i &lt;= 9</td>
<td>Multistate - Local</td>
<td>kdh1 * cdh1[0] * cdc14 / cell</td>
</tr>
<tr>
<td>9</td>
<td>cdh1[0] + dbm -&gt; cdh1[i+1]; 0 &lt;= i &lt;= 9</td>
<td>Multistate - Local</td>
<td>kdh1P * cdh1[0] * dbm / cell</td>
</tr>
<tr>
<td>10</td>
<td>-&gt; mbm</td>
<td>Local</td>
<td>ksn1 * P</td>
</tr>
<tr>
<td>11</td>
<td>mbm -&gt;</td>
<td>Local</td>
<td>km1</td>
</tr>
<tr>
<td>12</td>
<td>-&gt; mb1</td>
<td>Local</td>
<td>ksn1 * mb1</td>
</tr>
<tr>
<td>13</td>
<td>nh1 -&gt;</td>
<td>Local</td>
<td>km1n1</td>
</tr>
<tr>
<td>14</td>
<td>mnt3 + cdc3 + mm3</td>
<td>Local</td>
<td>kan3 * (cell/PI) + (cell/PI) + mm3</td>
</tr>
<tr>
<td>15</td>
<td>cdc3 -&gt;</td>
<td>Local</td>
<td>kbdc3</td>
</tr>
<tr>
<td>16</td>
<td>-&gt; cdc3</td>
<td>Local</td>
<td>kbdc3</td>
</tr>
<tr>
<td>17</td>
<td>cdc4 -&gt;</td>
<td>Local</td>
<td>kscdc4</td>
</tr>
<tr>
<td>18</td>
<td>cdc4 -&gt;</td>
<td>Local</td>
<td>kscdc4</td>
</tr>
<tr>
<td>19</td>
<td>cdc14 -&gt;</td>
<td>Local</td>
<td>kscdc14</td>
</tr>
<tr>
<td>20</td>
<td>cdc14 -&gt;</td>
<td>Local</td>
<td>kscdc14</td>
</tr>
<tr>
<td>21</td>
<td>mnt3 + cdc14 -&gt; mnt3; 0 &lt;= i &lt;= 5</td>
<td>Multistate - Local</td>
<td>kant3 * cdc14 + mnt3 / cell</td>
</tr>
<tr>
<td>22</td>
<td>mnt3 -&gt; cdc14; 0 &lt;= i &lt;= 5</td>
<td>Multistate - Mass Action</td>
<td>kant3 * cdc14 / cell</td>
</tr>
<tr>
<td>23</td>
<td>mnt3 + cdc14 + cdc14; 0 &lt;= i &lt;= 5</td>
<td>Multistate - Mass Action</td>
<td>kant3 * cdc14</td>
</tr>
<tr>
<td>24</td>
<td>-&gt; cmf1[0]</td>
<td>Mass Action</td>
<td>kcbf1</td>
</tr>
<tr>
<td>25</td>
<td>sbrf[0] + wh5[0] -&gt; cmf1[0]; 0 &lt;= i &lt;= 2</td>
<td>Multistate - Local</td>
<td>kmescbf[0] * wh5[0] / cell</td>
</tr>
<tr>
<td>26</td>
<td>cmf1[0] + sbrf[0] + wh5[0]; 0 &lt;= i &lt;= 2</td>
<td>Multistate - Mass Action</td>
<td>kmescbf1</td>
</tr>
<tr>
<td>27</td>
<td>cmf1[0] + sbrf[0]; 0 &lt;= i &lt;= 2</td>
<td>Multistate - Mass Action</td>
<td>kmescbf1</td>
</tr>
<tr>
<td>28</td>
<td>sbrf[0] + hbf -&gt; sbrf[0] + hbf; 1 &lt;= i &lt;= 4</td>
<td>Multistate - Local</td>
<td>kmescbf[0] + hbf</td>
</tr>
<tr>
<td>29</td>
<td>sbrf[0] + cdc14 + sbrf[0] + cdc14; 1 &lt;= i &lt;= 4</td>
<td>Multistate - Local</td>
<td>kmescbf[0] * cdc14 / cell</td>
</tr>
<tr>
<td>30</td>
<td>sbrf[0] + sbrf[0]; 0 &lt;= i &lt;= 4</td>
<td>Multistate - Mass Action</td>
<td>kmescbf[0]</td>
</tr>
<tr>
<td>31</td>
<td>sbrf[0] + cdc14 + sbrf[0] + cdc14; 0 &lt;= i &lt;= 3</td>
<td>Multistate - Local</td>
<td>kmescbf[0]</td>
</tr>
<tr>
<td>32</td>
<td>-&gt; hbf</td>
<td>Local</td>
<td>kcbf1 * (cell/PI) + hbf</td>
</tr>
<tr>
<td>33</td>
<td>hbf -&gt;</td>
<td>Mass Action</td>
<td>kcbf1</td>
</tr>
<tr>
<td>34</td>
<td>wh5[0] + cdc14 + wh5[0+1] + cdc14; 1 &lt;= i &lt;= 6</td>
<td>Multistate - Local</td>
<td>kwh5[0] * cdc14 / cell</td>
</tr>
<tr>
<td>35</td>
<td>wh5[0] + h5[0] + cdc14 + h5[0+1] + cdc14; 1 &lt;= i &lt;= 6</td>
<td>Multistate - Local</td>
<td>kwh5[0] * h5[0] / cell</td>
</tr>
<tr>
<td>36</td>
<td>wh5[0] + cdc14 + wh5[0+1]; 0 &lt;= i &lt;= 5</td>
<td>Multistate - Local</td>
<td>kwh5[0] + cdc14 + h5[0] / cell</td>
</tr>
<tr>
<td>37</td>
<td>wh5[0] + cdc14 + wh5[0+1]; 0 &lt;= i &lt;= 5</td>
<td>Multistate - Local</td>
<td>kwh5[0] + cdc14 + h5[0] / cell</td>
</tr>
</tbody>
</table>

Figure 5.2: Multistate reactions in the Cell Cycle Control Model of the budding yeast (part 1)

Most of the kinetics for the reactions follows the simple mass action rate law, and the multistate reactions use the ordered mechanism. A few of the reactions are not mass action, and the rate of reaction is affected by species that are not necessarily a reactant. There are more than 100 parameters and variables used in this model. In the reactions we used the
term *cell* which represents the volume of the cell. *PI* refers to the ploidy of the cell, which in this model of the yeast cell is one.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reaction</th>
<th>Type</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>-&gt; whs5[0]</td>
<td>Local</td>
<td>kbs * (cell / PI) * mls</td>
</tr>
<tr>
<td>39</td>
<td>whs5[0] -&gt; ; 0 &lt;= i &lt;= 6</td>
<td>multistate - Mass Action</td>
<td>gds * whs5[i]</td>
</tr>
<tr>
<td>40</td>
<td>cmp[0] -&gt; whs5[0] ; 0 &lt;= i &lt;= 2</td>
<td>multistate - Mass Action</td>
<td>gd6f * cmp[0]</td>
</tr>
<tr>
<td>41</td>
<td>-&gt; h5</td>
<td>Local</td>
<td>khs5 * (cell / PI) * mhs5</td>
</tr>
<tr>
<td>42</td>
<td>h5 -&gt;</td>
<td>Mass Action</td>
<td>gds5 * h5</td>
</tr>
<tr>
<td>43</td>
<td>-&gt; net1[0]</td>
<td>Local</td>
<td>kht1 * (cell / PI) * mt1</td>
</tr>
<tr>
<td>44</td>
<td>net1[0] + htl -&gt; net1(0-1) + htl ; 1 &lt;= i &lt;= 8</td>
<td>multistate - Local</td>
<td>kht1 * net1[0] * htl / cell</td>
</tr>
<tr>
<td>45</td>
<td>net1[0] + cdm -&gt; net1(0+1) + cdm ; 0 &lt;= i &lt;= 7</td>
<td>multistate - Local</td>
<td>kht1 * net1[0] * cdm / cell</td>
</tr>
<tr>
<td>46</td>
<td>net1[0] -&gt; ; 0 &lt;= i &lt;= 8</td>
<td>multistate - Mass Action</td>
<td>gd1 * net1[0]</td>
</tr>
<tr>
<td>47</td>
<td>mt[0] &gt; net1[0] ; 0 &lt;= i &lt;= 5</td>
<td>multistate - Mass Action</td>
<td>gd14 * net1[0]</td>
</tr>
<tr>
<td>48</td>
<td>mt[0] + cdm -&gt; mt(0+1) + cdm ; 0 &lt;= i &lt;= 4</td>
<td>multistate - Local</td>
<td>kmpnt * mt[0] * cdm / cell</td>
</tr>
<tr>
<td>49</td>
<td>mt[0] + htl -&gt; mt(0+1) + htl ; 1 &lt;= i &lt;= 5</td>
<td>multistate - Local</td>
<td>kmpnt * mt[0] * htl / cell</td>
</tr>
<tr>
<td>50</td>
<td>-&gt; htl</td>
<td>Local</td>
<td>kht11 * (cell / PI) * mht1</td>
</tr>
<tr>
<td>51</td>
<td>htl -&gt;</td>
<td>Mass Action</td>
<td>gdh11 * htl</td>
</tr>
<tr>
<td>52</td>
<td>cmp[0] + cdc14 -&gt; cmp(0-1) + cdc14 ; 1 &lt;= i &lt;= 2</td>
<td>multistate - Local</td>
<td>kcmp * cmp[0] * cdc14 / cell</td>
</tr>
<tr>
<td>53</td>
<td>cmp[0] + his -&gt; cmp(0-2) + his ; 1 &lt;= i &lt;= 2</td>
<td>multistate - Local</td>
<td>kcmp * cmp[0] * his / cell</td>
</tr>
<tr>
<td>54</td>
<td>cmp[0] + cbs -&gt; cmp(0+1) + cbs ; 0 &lt;= i &lt;= 1</td>
<td>multistate - Local</td>
<td>kcmp * cmp[0] * cbs / cell</td>
</tr>
<tr>
<td>55</td>
<td>cmp[0] + cns3 -&gt; cmp(0+1) + cns3 ; 0 &lt;= i &lt;= 1</td>
<td>multistate - Local</td>
<td>kcmp * cmp[0] * cns3 / cell</td>
</tr>
<tr>
<td>56</td>
<td>-&gt; mns3</td>
<td>Local</td>
<td>ksmn3 * mnt</td>
</tr>
<tr>
<td>57</td>
<td>mns3 -&gt;</td>
<td>Mass Action</td>
<td>gdmn3 * mns3</td>
</tr>
<tr>
<td>58</td>
<td>sbf(0) -&gt; ga + sbf(0)</td>
<td>Local</td>
<td>kag * (sbf(0) / cell) * (PI - ga)</td>
</tr>
<tr>
<td>59</td>
<td>ga -&gt;</td>
<td>Mass Action</td>
<td>kug * ga</td>
</tr>
<tr>
<td>60</td>
<td>ga -&gt; nbs * ga</td>
<td>Mass Action</td>
<td>ksnb * ga</td>
</tr>
<tr>
<td>61</td>
<td>nbs -&gt;</td>
<td>Mass Action</td>
<td>ksnb * nbs</td>
</tr>
<tr>
<td>62</td>
<td>-&gt; mc14</td>
<td>Local</td>
<td>kmc14 * PI</td>
</tr>
<tr>
<td>63</td>
<td>mc14 -&gt;</td>
<td>Mass Action</td>
<td>gdmc14 * mc14</td>
</tr>
<tr>
<td>64</td>
<td>-&gt; nshb</td>
<td>Local</td>
<td>kshb * PI</td>
</tr>
<tr>
<td>65</td>
<td>nshb -&gt;</td>
<td>Mass Action</td>
<td>gdnshb * nshb</td>
</tr>
<tr>
<td>66</td>
<td>-&gt; msh</td>
<td>Local</td>
<td>ksmsh * PI</td>
</tr>
<tr>
<td>67</td>
<td>msh -&gt;</td>
<td>Mass Action</td>
<td>gdmsh * msh</td>
</tr>
<tr>
<td>68</td>
<td>msh -&gt;</td>
<td>Local</td>
<td>kpmsh * PI</td>
</tr>
<tr>
<td>69</td>
<td>msh -&gt;</td>
<td>Mass Action</td>
<td>gdmnsh * msh</td>
</tr>
<tr>
<td>70</td>
<td>-&gt; htl</td>
<td>Local</td>
<td>kmsh1 * PI</td>
</tr>
<tr>
<td>71</td>
<td>htl -&gt;</td>
<td>Mass Action</td>
<td>gdmh1 * htl</td>
</tr>
<tr>
<td>72</td>
<td>-&gt; htl</td>
<td>Local</td>
<td>ksmht1 * htl</td>
</tr>
<tr>
<td>73</td>
<td>htl -&gt;</td>
<td>Mass Action</td>
<td>gdmh1 * htl</td>
</tr>
</tbody>
</table>

Figure 5.3: Multistate reactions in the *Cell Cycle Control Model* of the budding yeast (part2)

In Reaction 3 of Figure 5.2 we used the variable *cdh1pT*. In this model *Cdh1{0}* (the unphosphorylated form of Cdh1) is the high activity form (with rate constant kdbma = 0.025 fL molec⁻¹ min⁻¹). *cdh1pT* represents the summation of the low activity (phosphorylated) forms of Cdh1 (with rate constant kdbmi = 1.7 * 10⁻⁵ fL molec⁻¹ min⁻¹). We used the summation operator to define *cdh1pT* as sum(*cdh1{i};10).
Chapter 6

Multistate Modeling with the JigCell ModelBuilder and SBML

In Chapter 3, we saw the BioNetGen (and related tools such as NFSim) does not define relations of species (as reactants and products) involved in a reaction as successor or predecessor (or with other arithmetic expressions) in a straight-forward manner. Although BioNetGen can define states of species using patterns and mapping, there is a limitation of expressing relations in a group of reactions. For example, consider the phosphorylation reaction of a multistate species \( XP \) defined as \( XP\{i\} + E \rightarrow XP\{i+1\} + E \), where the value of \( i \) indicates the number of phosphate groups attached to XP. In this reaction \( XP\{i+1\} \) is the successor state of \( XP\{i\} \) as it contains one more phosphate group than \( XP\{i\} \) does. In BioNetGen, expressions for successor or predecessor such as \( i \rightarrow i + 1 \) or \( i \rightarrow i - 1 \) in species with indices are allowed only after providing explicit definitions for the state order and reaction rules (as we saw in Figure 3.5). But the order of species states plays an important role in the multistate reactions and demands a meaningful way of expression.

Multistate phosphorylation reactions appear frequently in living cells. In some cases, number of states for a substrate involved in the reaction (i.e. the number of phosphate groups attached to the species in a specific site or number of phosphorylated sites) appears to be more important than the individual states (i.e., the resulting phosphorylated residues). In a cascade of phosphorylation reactions with multiple states, the option to identify a state by index or position is required often. For example, multistate phosphorylation of the yeast cyclin dependent kinase inhibitor Sic1 has a threshold of six sites (six or more must be phosphorylated out of the nine possible phosphorylation sites of Sic1) to promote binding to the ubiquitin ligase (SCF^{Cdc4}) for Sic1 degradation regardless of which six are bound [17]. Providing the explicit definition of the order of states (which is the only option with tools such as BioNetGen) is cumbersome for users, we need a way to create models with less effort and a conversion mechanism for using the models with other simulation tools such as BioNetGen.
6.1 JCMB Spreadsheet

We have extended JCMB to support multistate modeling. In this section, we use the antagonism model of Chapter 2 to describe the modified JigCell GUI. Details for how JigCell is used to model single state reactions can be found in [25].

6.1.1 Species

For modeling a system with JCMB, defining the species is an important step. In JCMB,

- It is possible to define the species directly in the Species spreadsheet before entering the reactions as described in Figure 6.1.
- Alternatively, while the reactions are entered, JCMB can identify and automatically create the species involved by asking the required information (Figure 6.2).

![Figure 6.1: Species spreadsheet in JCMB](image)

![Figure 6.2: A multistate species creation dialog box in JCMB](image)

Before accepting a reaction in the reaction spreadsheet, JCMB will ensure that all involved species are defined appropriately. In other words, it will check that there is no conflict
between single state and multistate species naming, and the multistate species are defined properly with state ranges defined. For example, in Figure 6.2, Cdh1P has been identified as a one dimensional (with one site) multistate species and that acceptable values for ‘?:?’ inside the curly braces are numbers. Cdh1P{0:10} is the correct definition for this model.

### 6.1.2 Rules

The *Rules* spreadsheet accepts assignment rules to create variables for use in reactions. Besides other existing arithmetic operators, enhanced JCMB offers the summation operator as `sum(expression, lowerlimit, upperlimit)`. For multistate reactions, this operator is used to express activity of a multistate species. The range defined with lowerlimit and upperlimit automatically selects the related states of the species identified by index value. Figure 6.3 shows how we can define the $X_a$ variable for activity of Cdh1P.

![Figure 6.3: Rule spreadsheet in JCMB showing use of the `sum()` operator](image)

### 6.1.3 Reactions

![Figure 6.4: Reaction spreadsheet in JCMB: Reaction, Type, Equation, and Expanded Reactions columns for defining multistate reactions](image)

The *Reaction* spreadsheet is the main input area for reactions and related parameters in JCMB. There are four columns in this spreadsheet that are most important for multistate reactions as shown in Figure 6.4. A click in the *Reaction* column brings out the reaction entry dialog box (Figure 6.5) which takes reactions as $Reactant(s) \rightarrow Product(s)$. A multistate
reactant/product must be entered with some index variables (for example, $XP\{i\}$), whereas one specific state of the multistate species is treated as a normal species and can be entered directly with the state value (for example, $XP\{0\} \rightarrow$ is a normal single state reaction). The reaction entered in the dialog box will be accepted and shown in the Reaction field of that row (as in Figure 6.6) only after a successful pass through syntax checking and the species creation steps. The indices of multistate species will initially be shown as undefined or null.

![Figure 6.5: Reaction entry dialog box](image)

![Figure 6.6: An accepted multistate reaction in the Reaction column with undefined range for indices](image)

The second input column is named **Type**, highlighted in red in Figure 6.6 to indicate it is the next field that needs to be filled out. For normal reactions, this field will offer a simple dropdown box to select the rate law. The **Multistate Reaction Parameter Editor** dialog box will appear after clicking on this field if the reaction involves one or more multistate species. This dialog box, as shown in Figure 6.7, takes information for some parameters, as follows,

- Based on the reaction type, the **RateLaw** field shows some relevant rate laws in its dropdown selection box. The **Local** rate law gives options to customize the rate equation later in the **Equation** column. Any predefined rate law (defined in the **Functions** spreadsheet, discussed in the next section) will also appear here.
- The next area is for index variables used in this reaction. An acceptable range (with lower and upper limit) should be given for each index.
- Based on the type of the reaction, the reactions can be defined as ordered or disordered in the Reaction Mechanism area. In this reaction the option is disabled as, according
to definition, complex binding reactions do not have this sort of mechanism.
The fourth choice is for state dependency of the rate constant, \( k \). The rate constant
can be independent of the state value or dependent on it. For the later case, \( k \) can be
evaluated by a known function (e.g., Exponential, Stepwise) or some custom equation
which can be given in a textbox by selecting Local. Details of the possible values of
this field has been discussed under Chapter 4.3.

![Multistate reaction parameter editor dialog box](image)

Figure 6.7: Multistate reaction parameter editor dialog box

The \textit{Equation} column takes the rate equation based on the rate law and parameters
defined in the \textit{Type} field as shown in Figure 6.8. The \textit{Expanded Reactions} field becomes
enabled only for the multistate reactions to show the expanded view of the compact multi-
state reaction. It is helpful for verifying that the compact form represents the set of reactions
correctly. Figure 6.9 shows a view of this feature with all the reactions of the \textit{Antagonism}
model in multistate format.

Color has been used to assist modelers in filling out the required information in this
spreadsheet. After a reaction is given in the \textit{Reactions} column, the column that should be
filled out next (in this case, \textit{Type}) becomes red. As for multistate reactions, the \textit{Type} field
takes multiple parameters, and yellow highlighting is used to indicate incomplete information.
In Figure 6.10, rate law information is not provided but the range for the index \( i \) is already
given, so the field is highlighted yellow. If information is given completely and correctly then
the \textit{Equation} column becomes red, because it is the next column to be filled out.

6.1.4 Functions

This spreadsheet helps to define custom, model-specific rate laws and also lists the rate laws
and functions used in the model so far. For multistate reactions we may need to define
some custom rate laws as listed in Table 4.2. Figure 6.11 shows the rate laws used in the
Figure 6.8: *Equation* column in the Reaction spreadsheet

Figure 6.9: *Expanded Reactions* dialog box showing multistate reactions in the *Antagonism Model*
Figure 6.10: Highlighting columns with red or yellow for guiding in building the model

*Antagonism* model, for example, *Mass_Action.2.Ordered* is the name of the rate law for ordered phosphorylation reaction with two substrates.

![Figure 6.11: Functions spreadsheet](image)

6.1.5 Parameters

This spreadsheet is used for its original purpose (i.e., for definition and assignment of parameters) as for single state models. Figure 6.12 shows the list of parameters of the *Antagonism* model.

6.1.6 Equations

In multistate modeling, the *Equations* spreadsheet (Figure 6.13) can provide a means to check that everything is going as expected. With flattening of the model it is possible to get the list of ODEs with single state species to use in a simulation or in other tools.
Figure 6.12: Parameters spreadsheet showing parameters used in the *Antagonism Model*

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Units</th>
<th>Constant</th>
<th>Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_s$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$k_d$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$k_{a1}$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$k_a$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$k_b$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$k_1$</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Figure 6.13: Equations (ODEs) shown in multistate format

<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{b2}/dt$</td>
<td>$(k_d) - (k_a \cdot C_b)$</td>
<td></td>
</tr>
<tr>
<td>$C_{ih}(P)_{hit}$</td>
<td>$- (k_1 \cdot C_{ih}(P)) + (k_{a1})$</td>
<td></td>
</tr>
<tr>
<td>$C_{ih}(P)_{hit}$</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>$K_e$</td>
<td>$K_e \cdot C_{ih}(P(0)) + K_e \cdot \text{sum}<em>{C</em>{ih}(P)}$</td>
<td>Set by rule</td>
</tr>
<tr>
<td>$\text{sum}<em>{C</em>{ih}(P)}$</td>
<td>$\text{sum}(C_{ih}(P(0):1:10))$</td>
<td>Set by rule</td>
</tr>
<tr>
<td>$C_{ih}(P(0))$</td>
<td>Set by constant condition</td>
<td></td>
</tr>
<tr>
<td>$C_{ih}(P(0))$</td>
<td>Set by constant condition</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.14: Modification proposed for the `<StateFeature>` constructs of the SBML multi package (A) StateFeature (B) StateFeatureInstance
6.2 Multistate Modeling in SBML

A goal of JCMB is to store and reuse regulatory network models in a standard way. It uses the standard SBML format for its underlying structure. We have already discussed different kinds of multistate reactions in Chapter 4. In Chapter 3, we discussed the SBML level 3 multi package which is still under review by the SBML community. Here we will discuss how we can modify multi to include the definition of multistate models containing species with range of indices.

From Figure 3.2 we can see, in the current proposal of multi package, each <SpeciesType> has <ListOfStateFeatures> and each <StateFeature> can have a number of <PossibleValue>. This is a set of explicitly defined values. We propose to modify the <StateFeature> to make option to include a range of values or mathematical expression (Figure 6.14 (A)). Here, the existing <ListOfPossibleValues> and the newly added {<LowerIndex>, <UpperIndex>} or <Expression> constructs can not be used at the same time on one element. <LowerIndex> and <UpperIndex> both should be present with proper value if either one is present. The <Expression> construct is particularly useful for reactions (we will talk about it shortly).

<Selector> is the other construct that needs modification. <StateFeatureInstance> under the selector construct represents a list of state feature values which is a list of allowed values for applying restriction related to the reaction or rule. We propose to add {<LowerIndex>, <UpperIndex>} and <Expression> constructs in addition to the <ListOfStateFeatureValues> (Figure 6.14 (B)). Conditions and constraints are similar to the <StateFeature> construct.

The modification in <StateFeature> value also has impact on <SimpleSpeciesReference> under the <Reaction> construct. From Figure 3.4, we can see that <SimpleSpeciesReference> eventually links to a <SpeciesTypeInstance>, which contains the allowed range of species states to define conditional species reference for the corresponding reaction. According to our proposal (Figure 6.14), a <SpeciesType> can take mathematical expressions which are useful to define reactions similar to $XP\{i\} + E \rightarrow XP\{i + 1\} + E$ or $X\{i\} + Y\{j\} \rightarrow XY\{i, j\}$. Here, the variables $i, j$ are considered properties of the reactions and are used to define the allowed range or set of states for reactants and products. In this context, we propose to modify the <Reaction> construct (Figure 6.15). Here each <Reaction> has an optional <ListOfLocalVariable> and each <LocalVariable> defines either a range (with {<LowerLimit>, <UpperLimit>}) or a set (by <Values> which is a MathML Set construct) of values for that variable. The variable is called local as the scope of it is inside the corresponding reaction only.

We have implemented the multistate enhancement in JCMB using <annotations>. The representation for Cdh1 (the multistate species in the Antagonism model) using annotation is shown in Figure 6.16. Similarly, annotations for <SpeciesReference> and <Reactions> have been used to store index ranges of reactants/products and other reaction related parameters like mechanism or state dependency (Figure 6.17). The resulting SBML code for the summation rule has been shown in Figure 6.18.
Figure 6.15: Modification proposed for the <Reaction> construct of the SBML for multi package

```xml
<listOfSpecies>
  <species id="Clb2_1" name="Clb2" compartment="cell_1" />
  <species id="Cdh1P_1" name="Cdh1P" compartment="cell_1" />
  <annotation
    xmlns:jigcell="http://www.sbml.org/2001/ns/jigcell"
    <jigcell:index jigcell:lowerIndex="0"
      jigcell:upperIndex="10"></jigcell:index>
  </annotation>
  <species id="Cdc14_1" name="Cdc14" compartment="cell_1" />
  <species id="Xa_1" name="Xa" compartment="cell_1" />
</listOfSpecies>
```

Figure 6.16: Multistate species definition expressed in SBML using annotation (the Antagonism Model)
Figure 6.17: Multistate reaction definition expressed in SBML using annotation (the Antagonism Model)
6.3 Model Flattening

JCMB provides options to save models in multistate format for future editing. It also allows conversion of a multistate model into standard SBML format (Figure 6.19). The process is referred to as flattening the multistate model. As SBML constructs are interrelated, we need to take care of all dependencies. Fortunately, there is a particular order which allows flattening with no dependency violation. Figure 6.19 presents the flow diagram of this process.

The flattening process follows the steps described below to keep the integrity of the flattened model:

1. A standard SBML model is created with non-conflicting constructs like compartments, units etc.

2. The functions for rate laws generated from the multistate model are converted. If a multistate phosphorylation reaction uses the mass action-ordered rate law, a separate function with identifiable function name and specific equation format is generated in the multistate model. So in the conversion process, name and $SId$ of the function is recreated, the equation is resolved and an entry is given in the function map for identifying and replacing the function in the following steps.

3. Each multistate species is expanded into a list of single state species. Appropriate and distinguishable names are given and a unique $SId$ is created for each single state species in the list. All species from the multistate model are added to the new model.
Figure 6.19: The flattening process for a multistate model into a standard SBML model
4. In a multistate model, there might be some rules with a summation operator. In Step 4 the rules are expanded as required and also multistate species are replaced with single state species created in Step 3. For example, the assignment rule \( \text{sum}_{} Cdh1P = \text{sum}(Cdh1P\{i\}; 1; 10) \) (as in Figure 6.3) becomes \( \text{sum}_{} Cdh1P = Cdh1P\{1\} + Cdh1P\{2\} + Cdh1P\{3\} + Cdh1P\{4\} + Cdh1P\{5\} + Cdh1P\{6\} + Cdh1P\{7\} + Cdh1P\{8\} + Cdh1P\{9\} + Cdh1P\{10\} \) after the expansion of the summation operator. Then each multistate species \( Cdh1P\{1\}, \ldots, Cdh1P\{10\} \) is replaced with corresponding single state species \( Cdh1P_{1}, \ldots, Cdh1P_{10} \).

5. Each multistate reaction is expanded with replacement of each multistate reactant and product by corresponding single state species. For each reactant/product, the \text{SpeciesReference} construct is updated at this step.

6. Function definitions in \textit{annotation} and \textit{kineticlaw} are converted using the map created in Step 2 and species list from Step 3.

7. In multistate modeling parameters are used for various purposes, for instance, for expressing reaction mechanism (ordered or disordered), state dependency (stepwise, exponential or local) etc. In JCMB we store these parameters in the \textit{annotation} field. In this step, these parameters are used to expand the model. For example, the state dependency parameter is used to expand the rate law equation. If a multistate reaction contains an exponential state dependent rate constant \( k_{i} = k_{0}r_{i} \), then using the value of the state variable \( i \) the rate constant is evaluated for each single state reaction of the group. Finally those parameters are removed from the \textit{Reaction} construct in the flattened model.

8. Using the map created in Step 2, functions are inserted into the model. Unused parameters are detected from the multistate model and used parameters are inserted in the new model.

9. The rest of the constructs, such as \textit{events} etc., are added, the conservation relations are checked and the overall model integrity is validated.

10. A standard SBML model is generated and list of ODEs can be exported to other simulation tools.

After conversion the new SBML model contains the list of species shown in Figure 6.20, the list of rules shown in Figure 6.21 and the list of reactions as shown previously in Figure 2.2. JCMB also produces a list of ODEs (Figure 6.22) that can be used in simulation tools like \textit{XPPAUT} or \textit{OSCILL8}.

### 6.4 Compliance with Other Tools and Simulators

To check the integrity and compliance of the flattened SBML model, we loaded the flattened \textit{Antagonism Model} into COPASI and checked the details of the model constructs. It shows
Figure 6.20: List of species after flattening (the Antagonism Model)

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial Substance</th>
<th>Initial Concentration</th>
<th>Charge</th>
<th>Compartment</th>
<th>Substance Units</th>
<th>Spatial Size Units</th>
<th>Only Substance Units</th>
<th>Boundary Condition</th>
<th>Constant</th>
<th>Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1h02</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_0</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_1</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_2</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_3</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_4</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_5</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_6</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_7</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_8</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_9</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1P_10</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1h1H</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kn</td>
<td>cell</td>
<td>substance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum_C1h1P</td>
<td>cell</td>
<td>sum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.21: List of rules after flattening (the Antagonism Model)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum_C1h1P</td>
<td>Assignment</td>
<td>C1h1P_1 + C1h1P_2 + C1h1P_3 + C1h1P_4 + C1h1P_5 + C1h1P_6 + C1h1P_7 + C1h1P_8 + C1h1P_9 + C1h1P_10</td>
</tr>
<tr>
<td>kn</td>
<td>Assignment</td>
<td>kn * C1h1P_0 + 16 * sum_C1h1P</td>
</tr>
</tbody>
</table>

Figure 6.22: List of ODEs after conversion (the Antagonism Model)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dC1h02dt</td>
<td>(kn * (C1h1P_0) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_0dt</td>
<td>(kn * (C1h1P_0) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_1dt</td>
<td>(kn * (C1h1P_1) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_2dt</td>
<td>(kn * (C1h1P_2) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_3dt</td>
<td>(kn * (C1h1P_3) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_4dt</td>
<td>(kn * (C1h1P_4) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_5dt</td>
<td>(kn * (C1h1P_5) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_6dt</td>
<td>(kn * (C1h1P_6) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_7dt</td>
<td>(kn * (C1h1P_7) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_8dt</td>
<td>(kn * (C1h1P_8) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_9dt</td>
<td>(kn * (C1h1P_9) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1P_10dt</td>
<td>(kn * (C1h1P_10) + C1h1P) / C1h1P</td>
</tr>
<tr>
<td>dC1h1Hdt</td>
<td>0.0</td>
</tr>
<tr>
<td>kn</td>
<td>(kn * C1h1P_0 + 16 * sum_C1h1P)</td>
</tr>
<tr>
<td>sum_C1h1P</td>
<td>C1h1P_1 + C1h1P_2 + C1h1P_3 + C1h1P_4 + C1h1P_5 + C1h1P_6 + C1h1P_7 + C1h1P_8 + C1h1P_9 + C1h1P_10</td>
</tr>
</tbody>
</table>

39
that we can use the generated (flattened) SBML files with other standard tools and perform simulation tasks. A snapshot of the COPASI reactions is shown in Figure 6.23.

![COPASI reaction window](image)

Figure 6.23: Reactions of the Antagonism Model shown in the COPASI reaction window

In Chapter 5 we showed the multistate Cell Cycle Control Model of the budding yeast (Figures 5.2 and 5.3). Using JCMB we flattened that model as well and generated the list of ODEs (shown in Appendix). We then reproduced the deterministic time course simulation of the ClbM-Cdh1 antagonism part of the model. In this simulation we considered the ODEs generated from JCMB and used the parameter values and initial concentrations of species as described in the paper and supplementary materials of [1]. We considered the exponential growth of cell volume (i.e. \( \text{cell} \)) using the ODE \( \frac{d\text{cell}}{dt} = \alpha \times \text{cell} \) and also considered the cell division constraint of the model (i.e. cell is divided when concentration of ClbM goes below 12.5 nM). We then compared our simulation (Figure 6.24 b) with the plot (Figure 6.24 a) from [1] to demonstrate the correctness of the generated ODE list.
Figure 6.24: Deterministic time course simulation of some key regulatory species (Cdh1, ClbM, ClbS, and Cdc14) of the Cell Cycle Control Model of the budding yeast. (a) Plot taken from Figure 3 of [1] (b) Plot generated from the flattened model
Chapter 7

Conclusions and Future Work

In this thesis, we have discussed a representation for multistate species and reactions in regulatory network models. We have analyzed how the approach can be made more compliant and standardized with respect to existing constructs and the requirements of current research papers.

We have implemented a platform for multistate modeling by enhancing the JigCell software tools. Although we have attempted to collect and consider all possible types of multistate reactions, we could not implement all the features of those reactions. For example, we have implemented and tested only the distributive form of phosphorylation reactions in the current tool. But with the extendable structure of the tool it will be easy to implement more reaction types in future when needed. Apart from this, we have not yet tested our models with the concept of set instead of a range of states and also for species with more than two sites. This tool is a means to aid the modelers with current models and to give a platform for future enhancements.

As an extension to the multistate modeling efforts, we aim to integrate the multistate constructs with the concept of model aggregation, proposed by Randhawa et al. [20]. We are looking forward to enhancing aggregation proposal based on the analysis of large realistic models and the current demands of modelers. Our ultimate goal is to fuse the multistate and aggregation concepts together and provide an integrated graphical user interface for modelers where they can create models (single or multistate) in a hierarchical approach.
Bibliography


Appendix

This appendix lists the set of ODEs generated using JCMB from the Cell Cycle Control Model of budding yeast.

\[
\begin{align*}
\frac{dcdh1pT}{dt} & = cdh1_{1.1} + cdh1_{1.2} + cdh1_{1.3} + cdh1_{1.4} + cdh1_{1.5} \\
& + cdh1_{6.1} + cdh1_{7.1} + cdh1_{8.1} + cdh1_{9.1} + cdh1_{10.1} \\
\frac{dclbm}{dt} & = ksbm * (cell/PI) * mbm - (kdbma * cdh1_{0.1} * clbm)/cell \\
& - (kdbmi * cdh1pT * clbm)/cell - gdbm * clbm \\
\frac{dcdh1_{0}}{dt} & = (ksh1 * (cell/PI) * mh1) + kdh1 * cdh1_{1.1} + (kdh1 * cdh1_{1.1} * cdc14)/cell \\
& - gdh1 * cdh1_{0.1} - (kph1 * cdh1_{0.1} * clbs)/cell - (kph1p * cdh1_{0.1} * clbm)/cell \\
\frac{dcdh1_{1}}{dt} & = -kdh1p * cdh1_{1.1} + kdh1p * cdh1_{2.1} - (kdh1 * cdh1_{1.1} * cdc14)/cell \\
& + (kdh1 * cdh1_{2.1} * cdc14)/cell - gdh1 * cdh1_{1.1} + (kph1 * cdh1_{0.1} * clbs)/cell \\
& - (kph1 * cdh1_{1.1} * clbs)/cell + (kph1p * cdh1_{0.1} * clbm)/cell \\
& - (kph1p * cdh1_{1.1} * clbm)/cell \\
\frac{dcdh1_{2}}{dt} & = -kdh1p * cdh1_{2.1} + kdh1p * cdh1_{3.1} - (kdh1 * cdh1_{2.1} * cdc14)/cell \\
& + (kdh1 * cdh1_{3.1} * cdc14)/cell - gdh1 * cdh1_{2.1} + (kph1 * cdh1_{1.1} * clbs)/cell \\
& - (kph1 * cdh1_{2.1} * clbs)/cell + (kph1p * cdh1_{1.1} * clbm)/cell \\
& - (kph1p * cdh1_{2.1} * clbm)/cell \\
\frac{dcdh1_{3}}{dt} & = -kdh1p * cdh1_{3.1} + kdh1p * cdh1_{4.1} - (kdh1 * cdh1_{3.1} * cdc14)/cell \\
& + (kdh1 * cdh1_{4.1} * cdc14)/cell - gdh1 * cdh1_{3.1} + (kph1 * cdh1_{2.1} * clbs)/cell \\
& - (kph1 * cdh1_{3.1} * clbs)/cell + (kph1p * cdh1_{2.1} * clbm)/cell \\
& - (kph1p * cdh1_{3.1} * clbm)/cell \\
\frac{dcdh1_{4}}{dt} & = -kdh1p * cdh1_{4.1} + kdh1p * cdh1_{5.1} - (kdh1 * cdh1_{4.1} * cdc14)/cell \\
& + (kdh1 * cdh1_{5.1} * cdc14)/cell - gdh1 * cdh1_{4.1} + (kph1 * cdh1_{3.1} * clbs)/cell \\
& - (kph1 * cdh1_{4.1} * clbs)/cell + (kph1p * cdh1_{3.1} * clbm)/cell
\end{align*}
\]
\[
\begin{align*}
\frac{dcdh_{15}}{dt} &= -(kph1p * cdh1_{4} * clbm) / cell \\
\frac{dcdh_{16}}{dt} &= -kdh1p * cdh1_{5} + kdh1p * cdh1_{6} - (kdh1 * cdh1_{5} * cdc14) / cell \\
&+ (kdh1 * cdh1_{6} * cdc14) / cell - gdh1 * cdh1_{5} + (kph1 * cdh1_{4} * clbs) / cell \\
&- (kph1 * cdh1_{5} * clbs) / cell + (kph1p * cdh1_{4} * clbm) / cell \\
&- (kph1p * cdh1_{5} * clbm) / cell \\
\frac{dcdh_{17}}{dt} &= -(kdh1p * cdh1_{7}) + (kdh1p * cdh1_{8}) - (kdh1 * cdh1_{7} * cdc14) / cell \\
&+ (kdh1 * cdh1_{8} * cdc14) / cell - (gdh1 * cdh1_{7}) + (kph1 * cdh1_{6} * clbs) / cell \\
&- (kph1 * cdh1_{7} * clbs) / cell + (kph1p * cdh1_{6} * clbm) / cell \\
&- (kph1p * cdh1_{7} * clbm) / cell \\
\frac{dcdh_{18}}{dt} &= -(kdh1p * cdh1_{8}) + (kdh1p * cdh1_{9}) - (kdh1 * cdh1_{8} * cdc14) / cell \\
&+ (kdh1 * cdh1_{9} * cdc14) / cell - (gdh1 * cdh1_{8}) + (kph1 * cdh1_{7} * clbs) / cell \\
&- (kph1 * cdh1_{8} * clbs) / cell + (kph1p * cdh1_{7} * clbm) / cell \\
&- (kph1p * cdh1_{8} * clbm) / cell \\
\frac{dcdh_{19}}{dt} &= -(kdh1p * cdh1_{9}) + (kdh1p * cdh1_{10}) - (kdh1 * cdh1_{9} * cdc14) / cell \\
&+ (kdh1 * cdh1_{10} * cdc14) / cell - (gdh1 * cdh1_{9}) + (kph1 * cdh1_{8} * clbs) / cell \\
&- (kph1 * cdh1_{9} * clbs) / cell + (kph1p * cdh1_{8} * clbm) / cell \\
&- (kph1p * cdh1_{9} * clbm) / cell \\
\frac{dcdh_{10}}{dt} &= -(kdh1p * cdh1_{10}) - (kdh1 * cdh1_{10} * cdc14) / cell - (gdh1 * cdh1_{10}) \\
&+ (kph1 * cdh1_{9} * clbs) / cell + (kph1p * cdh1_{9} * clbm) / cell \\
\frac{dcdc14}{dt} &= ((ksc14 * (cell / PI) * mc14) - (gd14 * cdc14) - (kar * cdc14 * net1_0)) / cell \\
&- (kar * cdc14 * net1_1) / cell - (kar * cdc14 * net1_2) / cell - (kar * cdc14 * net1_3) / cell \\
&- (kar * cdc14 * net1_4) / cell - (kar * cdc14 * net1_5) / cell + (gd1 * rnt_0) \\
&+ (gd1 * rnt_1) + (gd1 * rnt_2) + (gd1 * rnt_3) + (gd1 * rnt_4) + (gd1 * rnt_5) \\
&+ (kdr * rnt_0) + (kdr * rnt_1) + (kdr * rnt_2) + (kdr * rnt_3) \\
&+ (kdr * rnt_4) + (kdr * rnt_5) \\
\frac{dlb}{dt} &= ((ksbs * (cell / PI) * mbs) - (gdb * clbs)) \\
\end{align*}
\]
\[
\begin{align*}
\frac{dnmbm}{dt} &= (ksmbm * PI) - (gdmbm * mbm) \\
\frac{dnhm1}{dt} &= (ksmh1 * PI) - (gdnhm1 * mh1) \\
\frac{dnn3}{dt} &= (ksnn3 * PI) - (gdnn3 * nn3) \\
\frac{dnctn3}{dt} &= (ksn3 * (cell/PI) * (cell/PI) * mn3) - (gdn3 * clnn3) \\
\frac{dnet1_0}{dt} &= -(kar * cdc14 * net1_0)/cell + (kdr * rnt_0) + ((kst1 * (cell/PI) * mnt1) \\
&+ (knt1 * net1_1 * hnt1)/cell - (kpt1 * net1_0 * clbm)/cell \\
&- (gdt1 * net1_0) + (gdt4 * rnt_0) \\
\frac{dnet1_1}{dt} &= -(kar * cdc14 * net1_1)/cell + (kdr * rnt_1) - (knt1 * net1_1 * hnt1)/cell \\
&+ (knt1 * net1_2 * hnt1)/cell + (kpt1 * net1_0 * clbm)/cell \\
&- (kpt1 * net1_1 * clbm)/cell - (gdt1 * net1_1) + (gdt4 * rnt_1) \\
\frac{dnet1_2}{dt} &= -(kar * cdc14 * net1_2)/cell + (kdr * rnt_2) - (knt1 * net1_2 * hnt1)/cell \\
&+ (knt1 * net1_3 * hnt1)/cell + (kpt1 * net1_1 * clbm)/cell \\
&- (kpt1 * net1_2 * clbm)/cell - (gdt1 * net1_2) + (gdt4 * rnt_2) \\
\frac{dnet1_3}{dt} &= -(kar * cdc14 * net1_3)/cell + (kdr * rnt_3) - (knt1 * net1_3 * hnt1)/cell \\
&+ (knt1 * net1_4 * hnt1)/cell + (kpt1 * net1_2 * clbm)/cell \\
&- (kpt1 * net1_3 * clbm)/cell - (gdt1 * net1_3) + (gdt4 * rnt_3) \\
\frac{dnet1_4}{dt} &= -(kar * cdc14 * net1_4)/cell + (kdr * rnt_4) - (knt1 * net1_4 * hnt1)/cell \\
&+ (knt1 * net1_5 * hnt1)/cell + (kpt1 * net1_3 * clbm)/cell \\
&- (kpt1 * net1_4 * clbm)/cell - (gdt1 * net1_4) + (gdt4 * rnt_4) \\
\frac{dnet1_5}{dt} &= -(kar * cdc14 * net1_5)/cell + (kdr * rnt_5) - (knt1 * net1_5 * hnt1)/cell \\
&+ (knt1 * net1_6 * hnt1)/cell + (kpt1 * net1_4 * clbm)/cell \\
&- (kpt1 * net1_5 * clbm)/cell - (gdt1 * net1_5) + (gdt4 * rnt_5) \\
\frac{dnet1_6}{dt} &= -(knt1 * net1_6 * hnt1)/cell + (knt1 * net1_7 * hnt1)/cell \\
&+ (knt1 * net1_5 * clbm)/cell - (kpt1 * net1_6 * clbm)/cell - (gdt1 * net1_6) \\
\frac{dnet1_7}{dt} &= -(knt1 * net1_7 * hnt1)/cell + (knt1 * net1_8 * hnt1)/cell \\
&+ (knt1 * net1_6 * clbm)/cell - (kpt1 * net1_7 * clbm)/cell - (gdt1 * net1_7) \\
\frac{dnet1_8}{dt} &= -(knt1 * net1_8 * hnt1)/cell
\end{align*}
\]
\[
\frac{drnt_0}{dt} = (kar \ast cdc14 \ast net1_0)/cell - (gd1 \ast rnt_0) - (kdr \ast rnt_0) \\
- (gd14 \ast rnt_0) - (kpnt \ast rnt_0 \ast clbm)/cell + (kdnt \ast rnt_1 \ast ht1)/cell
\]

\[
\frac{drnt_1}{dt} = (kar \ast cdc14 \ast net1_1)/cell - (gd1 \ast rnt_1) - (kdr \ast rnt_1) \\
- (gd14 \ast rnt_1) + (kpnt \ast rnt_0 \ast clbm)/cell - (kpnt \ast rnt_1 \ast clbm)/cell \\
- (kdnt \ast rnt_1 \ast ht1)/cell + (kdnt \ast rnt_2 \ast ht1)/cell
\]

\[
\frac{drnt_2}{dt} = (kar \ast cdc14 \ast net1_2)/cell - (gd1 \ast rnt_2) - (kdr \ast rnt_2) \\
- (gd14 \ast rnt_2) + (kpnt \ast rnt_1 \ast clbm)/cell - (kpnt \ast rnt_2 \ast clbm)/cell \\
- (kdnt \ast rnt_2 \ast ht1)/cell + (kdnt \ast rnt_3 \ast ht1)/cell
\]

\[
\frac{drnt_3}{dt} = (kar \ast cdc14 \ast net1_3)/cell - (gd1 \ast rnt_3) - (kdr \ast rnt_3) \\
- (gd14 \ast rnt_3) + (kpnt \ast rnt_2 \ast clbm)/cell - (kpnt \ast rnt_3 \ast clbm)/cell \\
- (kdnt \ast rnt_3 \ast ht1)/cell + (kdnt \ast rnt_4 \ast ht1)/cell
\]

\[
\frac{drnt_4}{dt} = (kar \ast cdc14 \ast net1_4)/cell - (gd1 \ast rnt_4) - (kdr \ast rnt_4) \\
- (gd14 \ast rnt_4) + (kpnt \ast rnt_3 \ast clbm)/cell - (kpnt \ast rnt_4 \ast clbm)/cell \\
- (kdnt \ast rnt_4 \ast ht1)/cell + (kdnt \ast rnt_5 \ast ht1)/cell
\]

\[
\frac{drnt_5}{dt} = (kar \ast cdc14 \ast net1_5)/cell - (gd1 \ast rnt_5) - (kdr \ast rnt_5) \\
- (gd14 \ast rnt_5) + (kpnt \ast rnt_4 \ast clbm)/cell - (kpnt \ast rnt_5 \ast clbm)/cell
\]

\[
\frac{dsbf_0}{dt} = (kac \ast sbf_0 \ast whi5_0)/cell - (kac \ast sbf_0 \ast whi5_1)/cell \\
- (sbf_0 \ast whi5_2)/cell + (kdc \ast cmp_0) + (kdc \ast cmp_1) \\
+ (kdbf \ast whi5_2)/cell + (kdbf \ast whi5_3)/cell + (kdnt \ast sbf_0)/cell - (kdnt \ast sbf_0)/cell
\]

\[
\frac{dsbf_1}{dt} = -(kdbf \ast sbf_1 \ast hbf)/cell + (kdbf \ast sbf_2 \ast hbf)/cell - (kdbf \ast sbf_1 \ast cdc14)/cell \\
+ (kdbf \ast sbf_2 \ast cdc14)/cell - (kdbf \ast sbf_1)/cell \\
+ (kdbf \ast sbf_0 \ast clbm)/cell - (kdbf \ast sbf_1 \ast clbm)/cell
\]

\[
\frac{dsbf_2}{dt} = -(kdbf \ast sbf_2 \ast hbf)/cell + (kdbf \ast sbf_3 \ast hbf)/cell - (kdbf \ast sbf_2 \ast cdc14)/cell \\
+ (kdbf \ast sbf_3 \ast cdc14)/cell - (kdbf \ast sbf_2)/cell \\
+ (kdbf \ast sbf_1 \ast clbm)/cell - (kdbf \ast sbf_2 \ast clbm)/cell
\]
\[
\begin{align*}
\frac{dsbf_3}{dt} &= -(kdbf * sbf_3 * hbf) / cell + (kdbf * sbf_4 * hbf) / cell - (kdbfp * sbf_3 * cdc14) / cell \\
&\quad + (kdbfp * sbf_4 * cdc14) / cell - (gdbf * sbf_3) \\
&\quad + (kpf * sbf_2 * clbm) / cell - (kpf * sbf_3 * clbm) / cell \\
\frac{dsbf_4}{dt} &= -(kdbf * sbf_4 * hbf) / cell - (kdbfp * sbf_4 * cdc14) / cell \\
&\quad - (gdbf * sbf_4) + (kpf * sbf_3 * clbm) / cell \\
\frac{dwhi5_0}{dt} &= -(kac * sbf_0 * whi5_0) / cell + (kdc * cmp_0) + (kdi5p * whi5_1 * cdc14) / cell \\
&\quad + (kdi5 * whi5_1 * hi5) / cell - (kpi5 * whi5_0 * cln3) / cell \\
&\quad - (kpi5p * whi5_0 * clbs) / cell + ((ksi5 * (cell/PI) * mi5) \\
&\quad - (gdi5 * whi5_0) + (gdbf * cmp_0) \\
\frac{dwhi5_1}{dt} &= -(kac * sbf_0 * whi5_1) / cell + (kdc * cmp_1) - (kdi5p * whi5_1 * cdc14) / cell \\
&\quad + (kdi5p * whi5_2 * cdc14) / cell - (kdi5 * whi5_1 * hi5) / cell + (kdi5 * whi5_2 * hi5) / cell \\
&\quad + (kpi5 * whi5_0 * cln3) / cell - (kpi5 * whi5_1 * cln3) / cell + (kpi5p * whi5_0 * clbs) / cell \\
&\quad - (kpi5p * whi5_1 * clbs) / cell - (gdi5 * whi5_1) + (gdbf * cmp_1) \\
\frac{dwhi5_2}{dt} &= -(kac * sbf_0 * whi5_2) / cell + (kdc * cmp_2) - (kdi5p * whi5_2 * cdc14) / cell \\
&\quad + (kdi5p * whi5_3 * cdc14) / cell - (kdi5 * whi5_2 * hi5) / cell + (kdi5 * whi5_3 * hi5) / cell \\
&\quad + (kpi5 * whi5_0 * cln3) / cell - (kpi5 * whi5_2 * cln3) / cell + (kpi5p * whi5_1 * clbs) / cell \\
&\quad - (kpi5p * whi5_2 * clbs) / cell - (gdi5 * whi5_2) + (gdbf * cmp_2) \\
\frac{dwhi5_3}{dt} &= -(kdi5p * whi5_3 * cdc14) / cell + (kdi5p * whi5_4 * cdc14) / cell \\
&\quad - (kdi5 * whi5_3 * hi5) / cell + (kdi5 * whi5_4 * hi5) / cell + (kpi5 * whi5_2 * cln3) / cell \\
&\quad - (kpi5 * whi5_3 * cln3) / cell + (kpi5p * whi5_2 * clbs) / cell \\
&\quad - (kpi5p * whi5_3 * clbs) / cell - (gdi5 * whi5_3) \\
\frac{dwhi5_4}{dt} &= -(kdi5p * whi5_4 * cdc14) / cell + (kdi5p * whi5_5 * cdc14) / cell \\
&\quad - (kdi5 * whi5_4 * hi5) / cell + (kdi5 * whi5_5 * hi5) / cell \\
&\quad + (kpi5 * whi5_3 * cln3) / cell - (kpi5 * whi5_4 * cln3) / cell \\
&\quad + (kpi5p * whi5_3 * clbs) / cell - (kpi5p * whi5_4 * clbs) / cell - (gdi5 * whi5_4) \\
\frac{dwhi5_5}{dt} &= -(kdi5p * whi5_5 * cdc14) / cell + (kdi5p * whi5_6 * cdc14) / cell \\
&\quad - (kdi5 * whi5_5 * hi5) / cell + (kdi5 * whi5_6 * hi5) / cell + (kpi5 * whi5_4 * cln3) / cell \\
&\quad - (kpi5 * whi5_5 * cln3) / cell + (kpi5p * whi5_4 * clbs) / cell \\
&\quad - (kpi5p * whi5_5 * clbs) / cell - (gdi5 * whi5_5) \\
\frac{dwhi5_6}{dt} &= -(kdi5p * whi5_6 * cdc14) / cell - (kdi5 * whi5_6 * hi5) / cell \\
\end{align*}
\]
\[
\frac{dcmp_0}{dt} = \frac{(kpi5 * whi5_5 * cln3)}{cell} + \frac{(kpi5p * whi5_5 * clbs)}{cell} - (gdi5 * whi5_6)
\]
\[
\frac{dcmp_1}{dt} = \frac{(kac * sbf_0 * whi5_0)}{cell} - (kdc * cmp_0) - (gdi5 * cmp_0) - (gdbf * cmp_0)
\]
\[
\frac{dcmp_2}{dt} = \frac{(kac * sbf_0 * whi5_1)}{cell} - (kdc * cmp_1 - gdi5 * cmp_1 - gdbf * cmp_1)
\]

\[
\frac{dhbf}{dt} = kshbf * (cell/PI) * mhb - gdhbf * hbf
\]
\[
\frac{dhi}{dt} = kshi5 * (cell/PI) * mhi5 - gdhi5 * hi5
\]
\[
\frac{dhi1}{dt} = ksht1 * (cell/PI) * mht1 - gdht1 * ht1
\]
\[
\frac{dga}{dt} = kag * (sbf_0/cell) * (PI - ga) - kdg * ga
\]
\[
\frac{dmbs}{dt} = ksmbs * ga - gdmbs * mbs
\]
\[
\frac{dmc14}{dt} = ksmc14 * PI - gdmc14 * mc14
\]
\[
\frac{dmhbf}{dt} = ksmhbf * PI - gdmhbf * mhb
\]
\[
\frac{dmi}{dt} = ksmi5 * PI - gdmi5 * mi5
\]
\[
\frac{dmhi}{dt} = kshmhi5 * PI - gdmhi5 * mhi5
\]
\[
\frac{dmt}{dt} = ksmt1 * PI - gdmnt1 * mt1
\]
\[
\frac{dhmt}{dt} = ksmht1 * PI - gdmhmt1 * mht1
\]