

MODELAGEM E ANÁLISE DE SISTEMAS DE
TRANSPORTE DE ÍONS EM MEMBRANAS
CELULARES USANDO VERIFICAÇÃO DE
MODELOS

MIRLAINE APARECIDA CREPALDE

ORIENTADOR: SÉRGIO VALE DE AGUIAR CAMPOS

COORIENTADOR: ALESSANDRA CONCEIÇÃO FARIA-CAMPOS

COLABORADOR: JADER CRUZ

**MODELAGEM E ANÁLISE DE SISTEMAS DE
TRANSPORTE DE ÍONS EM MEMBRANAS
CELULARES USANDO VERIFICAÇÃO DE
MODELOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais. Departamento de Ciência da Computação. como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte

27 de agosto de 2011

© 2011, Mirlaine Aparecida Crepalde.
Todos os direitos reservados.

Crepalde, Mirlaine Aparecida

C917m Modelagem e Análise de Sistemas de Transporte
de Íons em Membranas Celulares usando Verificação
de Modelos / Mirlaine Aparecida Crepalde. — Belo
Horizonte, 2011
xvii, 165 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais. Departamento de Ciência da
Computação.

Orientador: Sérgio Vale de Aguiar Campos

Coorientador: Alessandra Conceição Faria-Campos

Colaborador: Jader Cruz

1. Computação - Teses. 2. Software - Verificação -
Teses. 3. Métodos estatísticos - Teses. I. Orientador.
II. Coorientadora. III. Colaborador. IV. Título.

CDU 519.6*32 (043)




UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

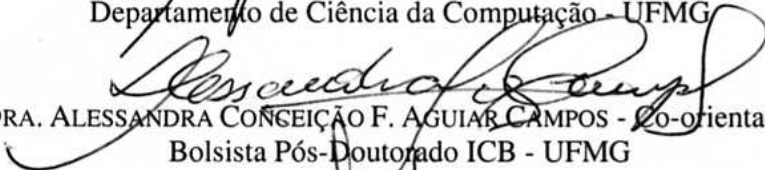
FOLHA DE APROVAÇÃO

Modelagem e análise de sistemas de transporte de íons em membranas celulares
usando verificação de modelos

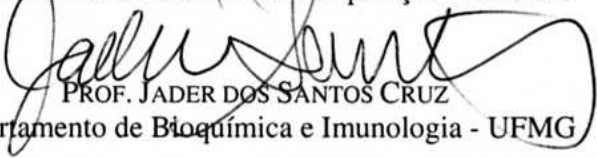
MIRLAINE APARECIDA CREPALDE

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. SÉRGIO VALE AGUIAR CAMPOS - Orientador
Departamento de Ciência da Computação - UFMG


DRA. ALESSANDRA CONCEIÇÃO F. AGUIAR CAMPOS - Co-orientadora
Bolsista Pós-Doutorado ICB - UFMG


PROF. RODRIGO RICHARD GOMES
Departamento de Ciência da Computação - PUCMG


PROF. JADER DOS SANTOS CRUZ
Departamento de Bioquímica e Imunologia - UFMG

Belo Horizonte, 27 de julho de 2011.

Dedico este trabalho a todas as pessoas que me incentivaram durante essa caminhada.

Agradecimentos

Agradeço primeiramente à Deus por me dar forças para continuar mesmo quando a vontade não era essa.

Aos meus pais pela compreensão nos momentos de ausência e por sempre me apoiarem.

Ao meu noivo Marco Aurélio, também pela enorme compreensão, amor e por tornar esses dois anos de estudos mais divertidos.

Ao Rogério, bom amigo, que sempre me ouviu nos momentos de indecisão.

Ao meu orientador Sérgio Campos pelas constantes palavras de motivação, pela orientação e por compartilhar pensamentos.

A minha co-orientadora Alessandra Conceição Faria-Campos, pelos conselhos e ajuda sempre que possível.

Ao professor Jader Cruz pelas reuniões e por todos os esclarecimentos de dúvidas.

Ao professor Rodrigo Richard Gomes por prontamente aceitar o convite para participar da minha banca examinadora.

A todos os colegas que passaram pelo laboratório LUAR e colaboraram com o bote do biscoto.

Aos professores do DCC por contribuírem para a minha formação profissional e pessoal.

Aos funcionários do DCC que sempre me ajudaram e me atenderam com simpatia.

Finalmente, a todos as demais pessoas que de alguma maneira contribuíram para a realização deste trabalho.

“A sabedoria é a meta da alma humana; mas a pessoa, à medida que em seus conhecimentos avança, vê o horizonte do desconhecido cada vez mais longe.”

(Heráclito)

Resumo

Recentemente há um interesse crescente na aplicação da Verificação Probabilística de Modelos (do inglês, *Probabilistic Model Checking - PMC*) na especificação formal e análise de sistemas biológicos. PMC é uma técnica que permite uma exploração exhaustiva do conjunto de estados de um sistema estocástico e pode fornecer visões valiosas do seu comportamento que são mais difíceis de serem obtidas usando somente abordagens tradicionais de análise de sistemas, como as simulações estocástica e determinística. Neste trabalho, propõe-se o uso de PMC para modelar e analisar sistemas em membranas celulares responsáveis pelo transporte de íons e cruciais para a vida das células.

Uma especificação formal e quantitativa da bomba de sódio-potássio é inicialmente apresentada. Essa bomba é um importante sistema de transporte de íons presente nas células animais e responsável por manter o equilíbrio das concentrações de sódio e potássio dentro da célula. O trabalho descreve como modelar o mecanismo da bomba usando PMC, considerando diferentes técnicas de modelagem com o objetivo de lidar com o problema da explosão de estados, inerente da técnica de verificação de modelos, independente do contexto. Importantes propriedades relacionadas à reversibilidade da bomba são verificadas usando PMC. Adicionalmente, o comportamento da bomba é investigado em termos de tendências de reações que computam se existe uma probabilidade superior de certas reações ocorrerem. Essas tendências permitem identificar os gargalos que levam ao lento funcionamento ou mesmo à interrupção da bomba com o passar do tempo.

O trabalho também introduz uma abordagem para detectar comportamento periódico em sistemas estocásticos usando uma técnica de verificação de modelos estatística. A abordagem proposta é aplicada para um sistema de oscilação intracelular de cálcio, encontrado em um grande número de tipos celulares em resposta à estímulos de componentes químicos extracelulares. Finalmente, uma metodologia para lidar com a coexistência de espécies químicas e moleculares com grandes diferenças em suas concentrações é também introduzida. Os resultados são ainda preliminares, mas indicam os ganhos em escalabilidade de recursos de memória com a abordagem proposta.

Abstract

Recently there has been a growing interest in the application of Probabilistic Model Checking (PMC) for the formal specification and analysis of biological systems. PMC is able to exhaustively explore all states of a stochastic model and can provide valuable insight into its behavior which are more difficult to see using only traditional methods for system analysis such as deterministic and stochastic simulation. In this work we propose the use of PMC for the modeling and analysis of cell membrane systems that are capable of moving ions across cell membranes and are crucial for the living cells.

We present a quantitative formal specification of sodium-potassium exchange pump, which is an important transport system present in all animal cells and responsible for keeping the equilibrium of potassium and sodium concentrations inside the cell.

We describe how the pump mechanism can be modeled using PMC, taking into consideration different modeling methodologies in order to deal with the state space explosion problem, inherent in model checking, regardless of context. Furthermore, we present some important properties about the pump reversibility that can be addressed directly with model checking, whereas with other traditional approaches they cannot be easily covered. Additionally, we reason about the pump behavior in terms of trend labels for the pump reactions that compute if there is a greater probability that the system takes specific transitions. These trends allow us to identify the bottlenecks that are responsible for the slow operation of the pump or even for its interruption over time.

The work also introduces an approach to detect noisy periodic behavior in stochastic systems using a statistical probabilistic model checking. The proposed approach is applied to the intracellular calcium oscillations, which is membrane system commonly observed in a large number of cell types in response to stimulation by an extracellular agonist. Finally a modeling methodology to deal with the coexistences of chemical and molecular species with huge differences in their concentrations is also introduced. Although the results are still preliminary, they indicate gains in scalability of memory resources using the proposed methodology.

Sumário

1	Introdução	1
1.1	Organização	3
2	Trabalhos Relacionados	5
2.1	Metodologias para Análise de Sistemas de Transporte de Íons em Membranas Celulares	5
2.2	Modelagem e Análise Formal de Sistemas Biológicos	7
3	Verificação de Modelos	10
3.1	Verificação Simbólica de Modelos	11
3.1.1	Estruturas de Kripke	12
3.1.2	Representação Simbólica da Estrutura de Kripke	13
3.1.3	Diagramas de Decisão Binária	15
3.1.4	Lógicas Temporais	18
3.2	Verificação Probabilística de Modelos	20
3.3	Ferramentas Utilizadas	24
3.3.1	PRISM	25
3.3.2	BIOLAB	31
3.3.3	Comparação entre PRISM e BIOLAB	37
4	Sistemas de Transporte de Íons em Membranas Celulares	41
4.1	A Célula e a Membrana Celular	41
4.2	Canais Iônicos	43
4.3	Bombas Iônicas	45
5	Metodologias para Modelagem de Sistemas Biológicos	48
5.1	Abordagem Individual	48
5.2	Abordagem Populacional	49
5.3	Lei da Ação das Massas	52
5.4	Abordagem Populacional Baseada na Química Discreta	53

5.5	Abordagem Populacional Baseada em Níveis	55
5.5.1	Imprecisão nos Modelos CTMC com Níveis	57
5.6	Comparação dos Modelos Baseados em População usando a Química Discreta e Abordagem em Níveis	59
6	Modelagem da Bomba de Sódio-Potássio	62
6.1	A Bomba de Sódio e Potássio e seu Funcionamento	62
6.2	Parâmetros Iniciais dos Modelos e Considerações	65
6.3	Modelo baseado na Química Discreta no PRISM	67
6.3.1	Abordagem Individual	70
6.3.2	Abordagem Populacional	70
6.3.3	Discussões	72
6.4	Abordagem Baseada em Níveis no PRISM	72
6.4.1	Discussões	74
6.5	Modelo no BIOLAB	76
6.5.1	Discussões	78
6.6	Comparações	79
7	Análise da Bomba de Sódio-Potássio	83
7.1	Modelo Empregado	83
7.2	Esgotamento do Potássio fora da Célula	84
7.3	Retorno do Potássio fora da Célula à sua Quantidade Inicial	87
7.4	Reversibilidade da Na,K-ATPase	87
7.5	Entendendo o Comportamento da Bomba através de Tendências	88
7.5.1	Tendências nos Modelos PRISM	90
7.5.2	Estudo das Tendências Considerando os Parâmetros Iniciais	93
7.5.3	Aumentando a Concentração de Potássio fora da Célula	96
7.5.4	Adicionando Reações de Síntese de ATP	99
7.6	Discussões	103
8	Estudos Adicionais	105
8.1	Oscilação Intracelular de Cálcio	105
8.1.1	Modelo Empregado	107
8.1.2	Influência do Tamanho do Passo	108
8.1.3	Detectando Comportamento Oscilatório	108
8.1.4	Detectando Comportamento Periódico	111
8.1.5	Implementação da Detecção de Periodicidade no BIOLAB	114
8.1.6	Detectando Períodos no Sistema de Oscilação Intracelular do Cálcio	116

8.2	Sistemas Multiescalares	117
8.2.1	Canal BK	119
8.2.2	Modelagem da Integração do Canal BK e do Sistema de Oscilação de Cálcio no PRISM	119
8.2.3	Resultados	122
9	Conclusões e Trabalhos Futuros	125
A	Modelos da Na,K-ATPase	128
A.1	Modelo Baseado na Química Discreta no PRISM	128
A.2	Modelo Baseado na Química Discreta no PRISM e com Número de Bombas Modelado usando a Abordagem Individual	131
A.3	Modelo Baseado na Química Discreta no PRISM e com Número de Bombas Modelado usando a Abordagem Populacional	134
A.4	Modelo Baseado em Níveis no PRISM	136
A.5	Modelo Discreto no BIOLAB	139
B	Tendências de Taxas de Transição dos Modelos da Na,K-ATPase	142
B.1	Tendências do Modelo PRISM Baseado na Química Discreta	142
B.2	Tendências do Modelo PRISM Baseado em Níveis	143
B.3	Tendências no modelo BIOLAB	145
C	Modelos do Sistema de Oscilação Intracelular de Cálcio e do Canal BK	147
C.1	Modelo PRISM Baseado em Níveis da Oscilação Intracelular de Cálcio	147
C.2	Modelo PRISM Baseado em Níveis da Integração da Oscilação Intracelular de Cálcio e do Canal BK	149
D	Solução Determinística do Sistema de Oscilação Intracelular de Cálcio no MATLAB	158
D.1	Conjunto de ODEs	158
D.2	Resolvendo o Conjunto de ODEs	159
	Referências Bibliográficas	160

Lista de Figuras

3.1	(a) Estrutura Kripke de dois estados. (b) Caminho na estrutura de Kripke.	13
3.2	Exemplo de uma transição e sua representação simbólica.	14
3.3	Árvore de decisão binária para a fórmula booleana $(a \wedge b) \vee (c \wedge d)$	16
3.4	OBDD para a fórmula $(a \wedge b) \vee (c \wedge d)$	18
3.5	(a) Estrutura de Kripke. (b) Expansão da estrutura de Kripke em uma árvore infinita.	19
3.6	Operadores CTL básicos.	21
3.7	Operadores LTL básicos.	22
3.8	Exemplo de modelo CTMC.	23
3.9	Exemplo de um sistema de reações químicas	25
3.10	Modelo no PRISM para o sistema de reações da Figura 3.9.	26
3.11	Rewards para modelo PRISM da Figura 3.10.	27
3.12	(a) Exemplo de um modelo CTMC. (b) MTBDD para o modelo CTMC.	31
3.13	Modelo no BIONETGEN para o sistema de reações da Figura 3.9.	32
3.14	Moléculas A e B	33
3.15	Reação de complexação das moléculas A e B	34
3.16	Reação de degradação da molécula A	34
3.17	Diagrama que mostra arquitetura simplificada do BIOLAB	37
4.1	Visão esquemática da membrana celular [Cindy e Geisiane, 2010].	42
4.2	Esquema de canal de íons [sophion, 2010].	44
4.3	Diagrama cinético para um simples canal com dois estados.	45
4.4	Mecanismo de mudança de configuração da bomba. E' e E'' são estados com sítios de ligação voltados para fora e dentro da célula, respectivamente. No <i>occluded state</i> , o íon não pode mover nem para dentro, nem para fora da célula. Pequenos círculos brancos representam grupos ligantes da proteína.	46
5.1	Sistema de reações químicas empregado como exemplo ao longo do capítulo.	49
5.2	Modelo no PRISM para o sistema de reações da Figura 5.1 considerando uma molécula de cada espécie.	50

5.3	Extensão do modelo PRISM da Figura 5.2 de modo a considerar duas moléculas de cada espécie e empregando a abordagem individual.	50
5.4	Novo modelo baseado na abordagem populacional para o sistema de reações da Figura 5.1 quando há duas moléculas de cada espécie.	52
5.5	Modificações no modelo populacional PRISM dado na química discreta para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5\text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11}\text{ M}^{-1}\text{s}^{-1}$, $r2 = 1\text{ s}^{-1}$, $r3 = 0,1\text{ s}^{-1}$ e um volume de 10^{-18} litros.	55
5.6	Modificações no modelo populacional BIOLAB dado na química discreta para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5\text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11}\text{ M}^{-1}\text{s}^{-1}$, $r2 = 1\text{ s}^{-1}$, $r3 = 0,1\text{ s}^{-1}$ e um volume de 10^{-18} litros.	56
5.7	Modelo populacional baseado em níveis para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5\text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11}\text{ M}^{-1}\text{s}^{-1}$, $r2 = 1\text{ s}^{-1}$, $r3 = 0,1\text{ s}^{-1}$ e $h = 2,5 \times 10^{-5}\text{ M}$	58
6.1	Mecanismo básico de funcionamento da bomba de sódio-potássio [Karp, 2008].	63
6.2	Esquema de reações para a Na,K-ATPase baseado no modelo de Albers-Post, retirado de Chapman et al. [1983]	64
6.3	Primeiro modelo discreto da Na,K-ATPase no PRISM	68
6.4	Extensão do modelo discreto da Na,K-ATPase no PRISM para permitir duas bombas usando abordagem individual.	70
6.5	Extensão do modelo discreto da Na,K-ATPase no PRISM para permitir duas bombas usando abordagem populacional.	71
6.6	Ilustração da redução do volume celular no modelo discreto.	72
6.7	Modelo baseado em níveis da Na,K-ATPase no PRISM.	74
6.8	Ilustração do agrupamento das moléculas e íons no modelo baseado em níveis.	75
6.9	Modelo discreto da Na,K-ATPase no BIOLAB.	76
7.1	Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase.	85
7.2	Definição do <i>reward</i> 'time' para modelos PRISM da Na,K-ATPase.	85
7.3	Tendências para a reação r_1 considerando o modelo PRISM discreto da Na,K-ATPase.	91
7.4	Extensão dos modelos CTMCs no PRISM para definição dos rewards plusKout e minusKout	92

7.5	Etapas de funcionamento da Na,K-ATPase para o modelo PRISM discreto. As setas são mostradas apenas para as reações que exibem tendência positiva. Configurações da bomba em cinza são aquelas cujas reações no sentido direto são as que exibem tendências positivas, enquanto aquelas na cor branca têm reações no sentido reverso com tendências ascendentes. A grossura da seta central no sentido direto de funcionamento da bomba indica a força da tendência geral de funcionamento da bomba nesse sentido. Assim, quanto mais grossa a seta, maior é a tendência geral da bomba de funcionar no sentido direto. (a) Etapa inicial. O círculo duplo representa a configuração inicial do sistema. (b) Etapa intermediária. (c) Etapa final.	94
7.6	Divisão ilustrativa do gráfico que representa a concentração de potássio fora da célula. O gráfico mostra a concentração de potássio fora da célula dada pela abordagem determinística e por dois caminhos gerados por simulação estocástica. As divisões (a), (b) e (c) do gráfico correspondem as respectivas etapas (a), (b) e (c) da Figura 7.5.	95
7.7	Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com alta concentração de potássio fora da célula.	96
7.8	Etapas de funcionamento da Na,K-ATPase para o modelo PRISM discreto quando o potássio fora da célula é $0,1 M$. As setas são mostradas apenas para as reações que exibem tendência positiva. Configurações da bomba em cinza são aquelas cujas reações no sentido direto são as que exibem tendências positivas, enquanto aquelas na cor branca têm reações no sentido reverso com tendências ascendentes. A grossura da seta central no sentido direto de funcionamento da bomba indica a força da tendência geral de funcionamento da bomba nesse sentido. Assim, quanto mais grossa a seta, maior é a tendência geral da bomba de funcionar no sentido direto. (a) Etapa inicial. O círculo duplo representa a configuração inicial do sistema. (b) Etapa intermediária. (c) Etapa final.	98
7.9	Divisão ilustrativa do gráfico que representa a concentração de potássio fora da célula. O gráfico mostra a concentração de potássio fora da célula dada pela abordagem determinística e por um caminho gerado por simulação estocástica. As divisões (a), (b) e (c) do gráfico correspondem as respectivas etapas (a), (b) e (c) da Figura 7.8.	99
7.10	Extensão do modelo discreto no PRISM para simular a síntese de ATP dentro da célula.	100
7.11	Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com reações de síntese de ATP.	101

7.12	Sumarização das tendências das reações no ciclo da Na,K-ATPase apresentado na Figura 6.2 com uma concentração externa de potássio equivalente à $0,1 M$ e com síntese de ATP.	101
7.13	Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com reações de síntese de ATP, considerando tempos longos.	102
8.1	Representação esquemática para oscilações do Ca^{2+} . Os fluxos de Ca^{2+} são representados por setas.	106
8.2	Modelo baseado em níveis para a oscilação intracelular de cálcio no PRISM.	109
8.3	Gráfico que representa a concentração de Ca_{SR} ao longo do tempo. As curvas em cinza representam simulações estocásticas do modelo baseado em níveis no PRISM variando o valor de h . Já a curva preta é o resultado de uma simulação determinística.	110
8.4	Extensão do modelo baseado em níveis para a oscilação intracelular de cálcio no PRISM a fim de detectar a oscilação do cálcio no SR.	111
8.5	Exemplo de caminho estocástico que exhibe periodicidade.	112
8.6	Autômato que detecta periodicidade.	113
8.7	Diagrama que mostra arquitetura simplificada da integração do BIOLAB e da ferramenta PRISM	115
8.8	Gráfico que representa a concentração de Ca_{SR} ao longo do tempo. A curva preta é o resultado de uma simulação estocástica do modelo baseado em níveis no PRISM usando $h = 0,001 \mu M$. Já a curva cinza é o resultado de uma simulação estocástica do modelo baseado em níveis no PRISM usando $h = 0,0001 \mu M$. As duas retas pontilhadas correspondem aos valores b_{low} e b_{high} que necessitam ser passados ao algoritmo, e delimitam a amplitude mínima $A = b_{low} - b_{high} $ do comportamento periódico.	117
8.9	Ilustração da divisão da concentração de um substrato em níveis e a sua respectiva atualização quando o tamanho passo da reação da qual participa é h_1 ou h_2	118
8.10	Modelo cinético que representa as mudanças de estado de um canal BK.	120
8.11	Representação esquemática para a integração do sistema de oscilação intracelular de cálcio e o canal BK. Os fluxos de Ca^{2+} são representados por setas.	120
8.12	Modelo PRISM baseado em níveis que representa a integração do sistema de oscilação intracelular de cálcio e o canal BK.	123

Lista de Tabelas

3.1	Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$	39
3.2	Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,001 e β (Erro Tipo 2) = 0,001. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$	39
3.3	Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,001 e β (Erro Tipo 2) = 0,001. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,001$	40
4.1	Patologias associadas ao mau funcionamento de canais iônicos. Obtido de [Émerson Flamarion da Cruz, 2006]	44
5.1	Variação do <i>Tamanho</i> do modelo em termos do número de nós da representação MTBDD, dos tempos de construção do modelo (T_c) e dos tempos de verificação do modelo (T_v) devido às variações no número de moléculas do modelo (N).	51
5.2	<i>Tamanho</i> do modelo em termos do número de nós da representação MTBDD, tempos de construção (T_c) e tempos de verificação (T_v) para modelos PRISM baseados na química discreta e em níveis. <i>Número de Caminhos</i> e tempo de verificação (T_v) para modelo BIOLAB baseado na química discreta.	60
5.3	Tempos de verificação (T_v) para modelos PRISM baseados na química discreta e em níveis. <i>Propriedade (3): 'init' $\Rightarrow \mathcal{P}_{\geq 0,5} [\mathbf{F} ab \geq 1]$</i> e <i>Propriedade (4): $\mathcal{P}_{=?} [\mathbf{F} ab \geq 1]$</i>	60
6.1	Reações do modelo de Albers-Post	65
6.2	Parâmetros fisiológicos normais associados ao esquema da Figura 6.2. . . .	66

6.3	Varição do <i>tamanho</i> do model (número de nós do MTBDD que representa a matriz de transição), do número de íons de sódio dentro e fora da célula ($\#Na_{in}^+$ e $\#Na_{out}^+$), do tempo para contruir o modelo (T_c) e verificá-lo (T_{v1} e T_{v2}) devido às variações no volume celular.	69
6.4	Varição do <i>tamanho</i> do modelo, em termos do número de nós da representação MTBDD, dos tempos de construção do modelo (T_c) e dos tempos de verificação do modelo (T_{v1} e T_{v2}) devido às variações do número de bombas (NB). O volume celular utilizado nos modelos foi de 10^{-20} l.	71
6.5	Varição do <i>tamanho</i> do modelo, em termos do número de nós da representação MTBDD, do tempo para construir o modelo ($Tempo_c$) e dos tempos de verificação ($Tempo_{v1}$ e $Tempo_{v2}$) devido às variações do tamanho do passo (h).	73
6.6	Varição do tempo para verificar o modelo da Na,K-ATPase no BIOLAB devido às variações no volume celular. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$	78
6.7	Varição do tempo para verificar o modelo da Na,K-ATPase no BIOLAB devido às variações no número de bombas (NP). O volume celular considerado é 10^{-20} l. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$	79
6.8	Tabela com resumo dos resultados da verificação dos vários modelos para o sistema de reações mostrado na Figura 3.9 e para a bomba de sódio e potássio. A coluna <i>Tempo</i> da tabela para modelos PRISM apresenta a soma dos tempos de construção e verificação. Adicionalmente, os valores T e F na coluna de <i>Resultado</i> indicam, respectivamente, se a propriedade sendo verificada é verdadeira ou falsa.	81
8.1	Parâmetros usados nas definições das taxas que regem a cinética do sistema de oscilação intracelular de cálcio.	107
8.2	Constantes usadas nas definições das taxas que regem a cinética do canal BK. A unidade das constantes marcadas com (*) é $\mu M^{-1} min^{-1}$, enquanto a unidade das demais constantes é min^{-1}	121

Capítulo 1

Introdução

A biologia de sistemas é um campo da biologia que visa analisar o comportamento e as relações de sistemas biológicos e é caracterizada pela combinação de experimentação, teoria e computação [Sauro et al., 2006]. Tradicionalmente, a modelagem e análise computacional de sistemas biológicos é baseada em um conjunto de equações diferenciais ordinárias (do inglês, *Ordinary Differential Equations - ODE*), que descrevem a evolução da concentração molecular média no tempo, ou na abordagem estocástica e discreta, originalmente proposta por Gillespie, e que representa o comportamento individual das moléculas [Wolkenhauer et al., 2004].

Há uma demanda crescente por técnicas de análise baseadas em métodos de verificação formal [Sauro et al., 2006], que são fundamentados em formalismos matemáticos para a especificação, desenvolvimento e verificação dos sistemas. A alcançabilidade de certos estados e as precondições para certos comportamentos padrões são questões que podem ser respondidas através dessas abordagens de maneira direta, ao invés de interpretar resultados de simulações, o que pode ser uma tarefa complexa. Nesse contexto, a técnica de verificação de modelos [Clarke e Emerson, 1981; Queille e Sifakis, 1982] se apresenta como uma ferramenta alternativa para investigar a dinâmica de sistemas biológicos. É uma técnica bem estabelecida que provê um caminho para formalmente descrever e analisar sistemas de estados finitos. Ela é capaz de explorar exhaustivamente todos os estados de um sistema modelado e, por isso, identificar eventos e condições ocultos. Verificação Probabilística de Modelos (PMC) é uma variante dessa técnica utilizada para modelar e analisar sistemas que exibem comportamento estocástico, como é o caso de muitos sistemas biológicos. Com PMC, informações quantitativas considerando as probabilidades ou taxas em que as transições ocorrem são incorporadas ao modelo. Adicionalmente, lógicas temporais são empregadas na especificação formal das propriedades do sistema que se deseja verificar. Essa abordagem vem sendo empregada no estudo de vários sistemas biológicos [Kwiatkowska et al., 2008; Calder et al.,

2006; Heath e et al, 2008; Clarke et al., 2008] e várias metodologias de modelagem vêm sendo apresentadas a fim de lidar com o problema da explosão do espaço de estados, inerente da técnica de verificação de modelos independente do contexto. O problema é caracterizado pelo crescimento exponencial do número de estados do modelo devido ao aumento do número de variáveis no mesmo. Isso gera um grande número de estados a ser explorado, tornando a verificação do modelo impraticável.

Neste trabalho, pretende-se modelar e analisar sistemas de transporte de íons em membranas celulares, essenciais para a vida das células, utilizando-se PMC. Esses sistemas participam de vários processos biológicos e são comumente analisados através de simulações estocásticas ou determinísticas. Uma das grandes motivações pelo atual interesse no estudo desses sistemas se deve ao fato de que várias patologias, como síndrome do QT longo, hipertemia maligna, epilepsia, esquizofrenia e doença de Alzheimer, serem relacionadas ao mau funcionamento dos mesmos [Dworakowska e Dolowy, 2000]. Além disso, eles são potenciais alvos de toxinas, principalmente os canais iônicos, sendo, por isso, de grande interesse na pesquisa biomédica. A intenção é que a técnica de verificação de modelos seja utilizada conjuntamente com as abordagens bem estabelecidas para amplificar o entendimento desses sistemas.

Dois sistemas de transporte de íons em membranas celulares serão estudados neste trabalho: bombas e canais iônicos. Canais iônicos são complexos proteicos que possuem poros aquosos. Esses poros podem abrir e fechar permitindo o fluxo de íons através da membrana celular na mesma direção do gradiente eletroquímico. Os canais iônicos são freqüentemente os principais alvos das toxinas de animais e das pesquisas por novos fármacos, sendo, por isso, interessantes na pesquisa biomédica. As bombas, por outro lado, movem íons através das membranas contra o gradiente eletroquímico. Para promover esse fluxo ativo de íons, elas utilizam fontes de energia primária como luz ou energia derivada da hidrólise do trifosfato de adenosina (ATP) [Lauger, 1991].

Este trabalho sumariza as principais técnicas para a modelagem e análise de sistemas biológicos utilizando PMC. As abordagens de modelagem são utilizadas para modelar vários representantes de uma mesma molécula ou íon e algumas delas, como a abordagem populacional e a abordagem baseada em níveis, visam lidar com o problema da explosão de estados que tipicamente surge no contexto biológico por esse aumento no número de cópias dos substratos envolvidos no sistema.

Essas técnicas de modelagem são aplicadas na descrição da bomba de sódio e potássio, enfatizando os principais desafios de cada uma delas. A bomba de sódio-potássio (Na,K-ATPase) é encontrada na membrana plasmática de praticamente todas as células animais e é responsável pelo transporte ativo de sódio e potássio através da membrana. O estudo realizado neste trabalho apresenta algumas questões sobre a dinâmica

de funcionamento dessa bomba, como o esgotamento de certos substratos e a sua reversibilidade. Além disso, uma metodologia que emprega a definição de tendência de reação é introduzida e utilizada no estudo do comportamento desse sistema biológico, visando identificar e compreender os motivos que levam ao lento funcionamento da bomba com o passar do tempo ou mesmo a sua interrupção.

Este trabalho introduz também uma abordagem para detectar o comportamento periódico ruidoso em sistemas biológicos estocásticos, empregando um algoritmo de verificação de modelos aproximado. A abordagem proposta é aplicada à oscilação intracelular de cálcio, que é um sistema biológico observado em uma variedade de tipos celulares em resposta aos estímulos de certas moléculas extracelulares e tem papel importante no controle de muitos processos celulares. Finalmente, uma metodologia para lidar com a característica multiescalar de certos sistemas biológicos é aplicada ao sistema composto pela oscilação intracelular de cálcio e pelo canal BK, um outro sistema biológico que permite o fluxo passivo de íons de pótassio através da célula. Apesar dos resultados serem ainda preliminares, eles indicam algumas das vantagens das metodologias propostas.

1.1 Organização

O capítulo 2 sumariza as principais abordagens que vêm sendo empregadas na modelagem e análise de sistemas de transporte de íons, e também apresenta os principais trabalhos na área de verificação de modelos biológicos, enfatizando as técnicas de modelagem ou análise introduzidas pelos mesmos, bem como os sistemas biológicos que estudam. O capítulo 3 introduz a teoria de verificação de modelos, iniciando pelos conceitos da verificação de modelos simbólica e passando pela descrição de sua versão probabilística. As ferramentas de verificação de modelos PRISM e BIOLAB, empregadas ao longo do trabalho, são também apresentadas nesse capítulo. Enquanto a ferramenta PRISM é uma técnica de verificação de modelos exata, a abordagem alternativa empregada, o BIOLAB, utiliza um algoritmo de verificação de modelos aproximado e que, por isso, pode lidar com sistemas biológicos mais complexos. O capítulo 4 trata de conceitos biológicos envolvidos com os sistemas de transporte de íons em membranas celulares. As bombas e os canais iônicos, que são dois representantes desses sistemas modelados e analisados neste trabalho, são detalhadamente descritos. O capítulo 5 sumariza o conjunto de técnicas de modelagem de sistemas biológicos usualmente adotadas quando se emprega a técnica de verificação de modelos. Duas das técnicas de modelagem apresentadas, a abordagem baseada em população e a abordagem baseada em níveis, visam lidar com o problema da explosão do espaço de estados. As técnicas

são empregadas na modelagem da bomba de sódio e potássio, apresentada no capítulo 6, o qual enfatiza os pontos fortes e fracos de cada uma delas. O capítulo 7 emprega a técnica de verificação de modelos para estudar alguns eventos que podem ocorrer na dinâmica da bomba de sódio e potássio, além de introduzir o conceito de tendência de reação que ajuda a compreender porque a bomba pára de funcionar no sentido normal com o passar do tempo. O capítulo 8 apresenta alguns estudos adicionais sobre a detecção de comportamentos periódicos através da ferramenta BIOLAB e sobre a modelagem de sistemas biológicos multiescalares. Finalmente, o capítulo 9 conclui esta dissertação e brevemente discute os trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

A primeira seção deste capítulo apresenta as principais metodologias e trabalhos na análise de sistemas de transporte de íons em membranas celulares, que é fundamentalmente baseada em simulações. Já a segunda seção sumariza os principais trabalhos na área de verificação de modelos biológicos, enfatizando as técnicas de modelagem ou análise introduzidas pelos mesmos, bem como os sistemas biológicos que estudam.

2.1 Metodologias para Análise de Sistemas de Transporte de Íons em Membranas Celulares

Várias técnicas experimentais vêm sendo utilizadas para estudar bombas e canais iônicos e vários modelos matemáticos e computacionais têm sido propostos para entendê-los. É interessante ressaltar, entretanto, que muito do esforço na tentativa de compreender esses sistemas de transporte iônicos através das membranas celulares é realizado por abordagens que utilizam simulações [Chapman et al., 1983; Rodrigues et al., 2008; Rothberg e Magleby, 2000; Pancrazio, 1994] e que, devido a sua natureza e a aleatoriedade envolvida com esses sistemas, podem ocultar eventos significativos. Por exemplo é não trivial determinar se um comportamento observado na simulação é típico ou uma anomalia.

Essas simulações podem ser determinísticas ou estocásticas. Na abordagem determinística, a modelagem das reações biológicas é baseada em um conjunto de equações diferenciais ordinárias não lineares (do inglês, *Ordinary Differential Equation* - ODE) que descrevem a evolução média das concentrações dos substratos envolvidos no tempo [Wolkenhauer et al., 2004]. Há, nesse caso, uma premissa de que a concentração dos substratos varia de forma contínua e determinística, embora a evolução de um sistema de reações químicas não seja um processo contínuo, porque os níveis das populações

moleculares podem mudar somente por quantidades inteiras discretas, e nem a evolução do processo no tempo é determinística [Gillespie, 1977]. Apesar disso, essa abordagem é bastante empregada e é interessante de ser utilizada quando as reações são rápidas e o número de moléculas envolvidas é grande. O paradigma alternativo de modelagem, originalmente proposto por Gillespie [1977], produz contagem de moléculas ou íons, cujas taxas de interação são controladas por distribuições exponenciais. Em um modelo estocástico, diferentemente da abordagem determinística, existem várias possibilidades para o comportamento do sistema, onde cada possibilidade é associada a uma certa probabilidade. A maneira usual de analisar esses modelos estocásticos é através de simulação, gerando caminhos que provêm diferentes visões do sistema. Nesse caso, para recuperar resultados relevantes estatisticamente é geralmente necessário gerar um grande número de caminhos, o que normalmente é mais demorado que somente resolver um conjunto de equações diferenciais não lineares [Spieler, 2009].

Existem ainda repositórios de modelos de canais e bombas de íons, como o CellML¹ e o Systems Biology Markup Language (SBML)² que descrevem os modelos biológicos utilizando a linguagem de marcação XML. Há ainda um vasto arcabouço de ferramentas associadas a esses repositórios, incluindo simuladores.

Alguns poucos trabalhos empregam outras abordagens para análise, diferentes de simulação, como Owen et al. [2006], que utiliza a notação formal Z para construir uma especificação formal do comportamento de canais do modelo de Beeler-Reuter do miócito ventricular de mamíferos. A notação Z é baseada na matemática padrão usada na teoria axiomática de conjuntos, em λ -*calculus* e na lógica de predicado de primeira ordem. O trabalho prova que um modelo mais refinado da célula cardíaca é consistente logicamente e matematicamente com um primeiro modelo mais abstrato.

Ciobanu et al. [2002] expressa a cinética da bomba de sódio-potássio em termos da linguagem de álgebra de processo π -*calculus*. O mecanismo da bomba é sumarizado em um conjunto de reações elementares e reversíveis, que representam interações moleculares e mudanças de configuração da bomba, permitindo um raciocínio formal sobre o sistema. Além disso, nesse trabalho a ferramenta *Mobility Workbench* (MWB), um algoritmo de verificação de modelos para manipular e analisar sistemas em π -*calculus*, é utilizada para verificar questões computacionais relacionadas com bisimilaridade e com estados blocantes (estados que podem ser atingidos mas que não possuem transições a partir deles). A bisimilaridade é uma técnica de redução exata do espaço de estados de um sistema: o conjunto de estados de um modelo é particionado tal que os estados equivalentes satisfazem as mesmas proposições atômicas. Análises podem então ser realizadas no modelo reduzido, com resultados idênticos para o modelo original.

¹<http://www.cellml.org/>

²<http://sbml.org/>

Apesar de ser o trabalho que serviu como ponto de partida para a modelagem e análise quantitativa da bomba de sódio-potássio usando PMC, apresentadas nos capítulos 6 e 7, ele não foca em uma modelagem e análise quantitativas.

Ciocchetta e Guerriero [2009] modelam o canal iônico receptor de inositol 1,4,5-trifosfato (IP_3), envolvido no movimento de íons de cálcio dentro da célula, utilizando uma extensão do Bio-PEPA que engloba a idéia de compartimentos. Bio-PEPA é uma linguagem formal para modelagem e análise de redes bioquímicas, baseada em álgebra de processos. Esse trabalho também realiza análises utilizando simulação determinística e a ferramenta PRISM para verificar a característica oscilatória do sistema biológico modelado. Entretanto, o foco do trabalho é a apresentação da extensão da linguagem Bio-PEPA para suportar compartimentos, sendo a modelagem e análise do canal iônico apenas um exemplo da aplicação da nova definição.

2.2 Modelagem e Análise Formal de Sistemas Biológicos

Recentemente há interesse crescente no uso de técnicas de verificação de modelos no entendimento das diversas vias bioquímicas que compõem os seres vivos. Primeiramente, a verificação de modelos pode ser utilizada para garantir que o modelo é uma abstração razoável do sistema real através da verificação de certas propriedades considerando todos os caminhos possíveis, como oposto à técnica de simulação, que fornece apenas um único caminho do sistema modelado. Ela também pode ser empregada para obter novas informações sobre o sistema, tentando provar e verificar propriedades cujas respostas são ainda desconhecidas. Isso pode prover uma maneira relativamente barata para determinar se um experimento no laboratório deve ser executado, dado que esses últimos são tipicamente caros e demorados [Pinto et al., 2007].

Um dos principais problemas da técnica de verificação de modelos na prática é o conhecido problema da explosão do espaço de estados, que decorre do crescimento exponencial no espaço de estados do modelo matemático que representa o sistema se sua complexidade aumenta. Os trabalhos de Kwiatkowska et al. [2010]; Ciocchetta et al. [2008]; Andrei e Calder [2010] apresentam técnicas de modelagem de sistemas biológicos que visam lidar com esse problema no contexto biológico. No trabalho de Kwiatkowska et al. [2010], um estudo de caso baseado na via de sinalização do Fator de Crescimento do Fibroblasto (FCF) na biossíntese de receptores é utilizado para ilustrar o emprego das técnicas de modelagem conhecidas como abordagem individual e abordagem populacional. Essas técnicas são úteis quando há necessidade de modelar mais de um representante de uma mesma espécie (um tipo de molécula ou íon, por

exemplo). A abordagem individual é algumas vezes necessária quando a propriedade de interesse se refere ao comportamento de um representante particular da espécie. Entretanto, ela sofre do problema da explosão do espaço de estados. A metodologia alternativa, a abordagem populacional, modela apenas o número de representantes de cada tipo de espécie através de contadores, sendo bem mais escalável. O trabalho de Ciocchetta et al. [2008] introduz a abordagem baseada em níveis, que emprega modelos mais abstratos que os modelos que aqueles caracterizados por contagem de moléculas ou íons. Em um modelo baseado em níveis, os estados do sistema são caracterizados por intervalos de concentrações, discretizados em um dado número de níveis para cada espécie. A grande vantagem dessa metodologia é a redução do espaço de estados do modelo. Andrei e Calder [2010] empregam essa abordagem para modelar a interferência da proteína *scaffold* nas vias do Monofosfato de Adenosina Cíclico (cAMP) e do Raf-1/MEK/ERK, interação que tem papel fundamental no processo de sobrevivência, transformação e proliferação celular. Além disso, esse trabalho utilizou a ferramenta de verificação probabilística de modelos PRISM [Kwiatkowska et al., 2010] no estudo de eventos dependentes e do comportamento pulsante do sistema.

Ballarini et al. [2009]; Spieler [2009] focam no estudo de comportamentos oscilatórios e periódicos, que caracterizam uma variedade de sistemas biológicos, empregando a técnica de verificação de modelos. Tais comportamentos podem não ser evidentes quando se utiliza uma análise via ODE, dado que essa abordagem obtém os valores médios das concentrações dos substratos no tempo. Por outro lado, embora a simples inspeção visual de alguns caminhos gerados através da simulação estocástica possa revelar a presença de oscilações, essas observações podem ser verdadeiras somente para os caminhos gerados. Outros caminhos podem fornecer outras visões da dinâmica do modelo. Com a técnica de verificação de modelos é possível precisamente verificar se a oscilação continua indefinidamente ou pára, de uma maneira progressiva ou abrupta. Essa análise formal é útil principalmente se a oscilação existe, mas sua fase, período e amplitude variam estocasticamente, sendo, neste caso, de difícil identificação usando somente a simulação estocástica. Spieler [2009] integra um automata finito capaz de identificar oscilações, bem como recuperar os tamanhos dos períodos em sistemas oscilatórios, em modelos descritos na ferramenta PRISM. Ballarini et al. [2009] caracteriza sistemas oscilatórios em termos de propriedades baseadas em lógica temporal e utiliza também a ferramenta PRISM para identificar aspectos oscilatórios no sistema de reações conhecido como *3-way oscillator*.

Monteiro et al. [2008] estuda o problema de perguntar questões interessantes e relevantes, que é crítico na modelagem em geral e mais ainda no contexto de técnicas de verificação formal, dado que não é trivial para não especialistas formular perguntas

usando a lógica temporal. Ele elabora quatro padrões para análise de modelos biológicos, baseados numa revisão de questões frequentemente perguntadas por modeladores, e que podem ser automaticamente traduzidas para lógica temporal. Padrões em lógica temporal são descrições em alto nível de questões frequentemente perguntadas em um domínio da aplicação e formuladas em lógica temporal. Além disso, nesse trabalho utiliza-se os padrões definidos para uma análise qualitativa da rede de regulação genética envolvida na resposta à privação de carbono na *Escherichia Coli*, empregando a ferramenta de verificação não-probabilística de modelos NuSMV [Cimatti et al., 2002].

Finalmente o trabalho [Kwiatkowska et al., 2008] emprega a ferramenta PRISM para modelar e verificar a via de reações da proteína quinase ativada por mitógeno, que também exerce papel fundamental na proliferação e sobrevivência de muitas células. Já Clarke et al. [2008] emprega a ferramenta de verificação aproximada de modelos BIOLAB na análise da via de sinalização de receptores das células T, as quais têm papel crucial no sistema imunológico. Esse trabalho foca na apresentação do algoritmo de verificação BIOLAB e também no estudo da biestabilidade na dinâmica da rede de sinalização de receptores da célula T.

Capítulo 3

Verificação de Modelos

Verificação de modelos (do inglês, *model checking*) é uma técnica poderosa de verificação formal para garantir a correção automática de sistemas de estados finitos. Ela foi inicialmente proposta de forma independente por Clarke e Emerson [1981] e Queille e Sifakis [1982], no início dos anos 80, e vem sendo utilizada para garantir a correção de sistemas de software, hardware e protocolos de comunicação, e recentemente na análise de sistemas biológicos. Nessa técnica os sistemas são modelados em uma máquina de estados finitos, usualmente descritos em linguagens de alto nível, e propriedades de interesse especificadas em lógicas temporais. Dado um modelo M , um conjunto de estados iniciais S_0 e uma propriedade ϕ , o algoritmo de verificação automaticamente determina se o modelo satisfaz a propriedade ϕ . Isso é feito explorando as transições entre estados exaustivamente para verificar as propriedades especificadas.

Comparado com outros métodos de análise de sistemas como testes e simulação, a técnica de verificação de modelos possui diversas vantagens. Primeiro, ela é totalmente automática: após modelar o sistema e especificar as propriedades desejadas, o programa verificador realiza a análise sem necessidade de intervenção humana. Além disso, garante que o modelo atende às propriedades especificadas já que todo o espaço de estados é examinado exaustivamente. Com isso, mesmo erros pequenos que passam despercebidos por outras técnicas, como emulação, testes e simulação, podem potencialmente ser revelados usando a verificação do modelo. Por fim, caso o modelo falhe em atender alguma propriedade, o verificador geralmente produz um contra-exemplo, que é bastante útil para compreender e corrigir o erro encontrado.

Mesmo sendo poderosa e amplamente empregada, a técnica sofre com o problema da explosão de estados: à medida que a complexidade do sistema sendo modelado aumenta, caracterizado pelo número de variáveis que o sistema possui, o número de estados cresce exponencialmente. Cada estado é representado por uma combinação única de valores das variáveis que compõe o modelo. Seja $v_i \in V$ uma variável arbi-

trária pertencente ao conjunto V de todas as variáveis do modelo e D_i o conjunto que representa todos os possíveis valores que v_i pode assumir. O número total de estados que o sistema terá é $\prod_{v_i \in V} |D_i|$.

Diversas técnicas foram propostas a fim de minimizar este problema. McMillan [1992] deu uma contribuição fundamental ao propor a utilização de Diagramas de Decisão Binária (do inglês, *Binary Decision Diagrams - BDDs*) [Bryant, 1986a] para representação simbólica das relações de transição entre os estados. Nessa representação implícita, cada estado é codificado pela atribuição de valores booleanos para o conjunto de variáveis associadas com o sistema. As transições podem, por isso, ser expressas como uma fórmula booleana em termos dos dois conjuntos de variáveis, um conjunto codificando o estado antigo e o outro codificando o atual. Assim, é possível evitar a construção explícita de grafos de estados do sistema. Isto possibilitou uma descrição mais compacta do modelo e deu um salto no tamanho de um sistema factível para a casa dos 10^{20} estados. Desde então, várias melhorias foram propostas na técnica baseada em BDDs, permitindo a verificação de sistemas com até 10^{120} estados. Campos e Clarke [2001] apresentam, por exemplo, uma maneira eficiente de representar tempo empregando BDDs, permitindo, assim, a verificação de sistemas que se aproximam mais da realidade. Outra técnica para reduzir o tamanho do espaço a ser explorado é a redução por ordem parcial [Godefroid et al., 1996] que agrupa sequências de eventos assíncronos que resultam no mesmo estado.

A seção seguinte apresenta a versão simbólica da técnica de verificação de modelos, introduzindo conceitos básicos relacionados com a mesma e que são também empregados em sua versão probabilística, a qual é descrita posteriormente. Por fim, as ferramentas de verificação de modelos empregadas neste trabalho são também discutidas.

3.1 Verificação Simbólica de Modelos

Verificação simbólica de modelos (do inglês, *Symbolic Model Checking*) é um conjunto de algoritmos eficientes que utiliza uma representação concisa do grafo de transições de estado do modelo. Ele é chamado de simbólico porque é baseado na manipulação de fórmulas booleanas.

As subseções seguintes apresentam os conceitos envolvidos com essa metodologia baseados nas seguintes referências: Clarke et al. [1999] e Song [2004]. Inicialmente, a estrutura de Kripke, um modelo matemático para representar um sistema, bem com a sua transformação para a representação simbólica são introduzidos. Posteriormente, o emprego de BDDs para obter uma representação compacta das fórmulas booleanas é

descrito. Por fim, o conceito de lógicas temporais, formalismos usados para representar especificações formais a serem verificadas, é também introduzido.

3.1.1 Estruturas de Kripke

Estrutura de Kripke é um formalismo matemático para descrever o sistema a ser verificado. Ele é representado como um grafo direcionado, onde os vértices são os estados do modelo e as arestas as transições entre os estados. Adicionalmente, uma função mapeia cada estado no subconjunto de proposições atômicas¹ que são verdadeiras naquele estado.

Formalmente, seja AP um conjunto de proposições atômicas, uma *estrutura de Kripke* M sobre AP é uma tupla $M = (S, S_0, R, L)$ onde:

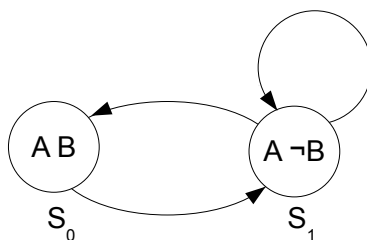
1. S é o conjunto finito de todos os estados possíveis do sistema;
2. $S_0 \subseteq S$ é o conjunto de estados iniciais do sistema;
3. $R : (S \times S)$ é uma relação de transição total, isto é, para cada estado $s \in S$, há um estado $s' \in S$ tal que $R(s, s')$;
4. $L : S \rightarrow 2^{AP}$ é uma função que associa cada estado com o conjunto de proposições atômicas verdadeiras naquele estado.

Um caminho em uma estrutura M a partir de um estado inicial s_0 é uma sequência infinita de estados $\pi = s_0s_1s_2\dots$ e a relação $R(s_i, s_{i+1})$ é segura para todo $i \geq 0$.

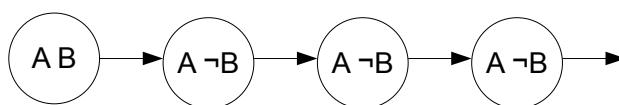
A Figura 3.1 mostra a representação gráfica de uma *estrutura de Kripke* para um modelo e um caminho na sua estrutura. Os componentes de M que definem a estrutura de Kripke são:

1. $S = \{\mathcal{S}_0, \mathcal{S}_1\}$;
2. $S_0 = \{\mathcal{S}_0\}$;
3. $R = \{(\mathcal{S}_0, \mathcal{S}_1), (\mathcal{S}_1, \mathcal{S}_0), (\mathcal{S}_1, \mathcal{S}_1)\}$;
4. $L(\mathcal{S}_0) = \{A, B\}$, $L(\mathcal{S}_1) = \{A, \neg B\}$.

¹Uma proposição é uma sentença (matemática ou não) que pode ser verdadeira ou falsa. Uma proposição é atômica ou simples quando não contém nenhuma outra proposição como parte integrante de si mesma. Exemplos de proposições atômicas: 'A lua é o único satélite do planeta terra' (verdade); 'O número 712 é ímpar' (falso).



(a)



(b)

Figura 3.1. (a) Estrutura Kripke de dois estados. (b) Caminho na estrutura de Kripke.

3.1.2 Representação Simbólica da Estrutura de Kripke

Uma estrutura de Kripke pode ser adicionalmente representada através de fórmulas booleanas, que empregam conectores lógicos (conjunção \wedge , disjunção \vee , negação \neg e implicação \rightarrow).

Seja $V = \{v_1, \dots, v_n\}$ o conjunto de variáveis do sistema e $D = \{D_1, \dots, D_n\}$ o conjunto finito dos domínios, tal que D_i representa o conjunto de todos os possíveis valores que v_i pode assumir. Um estado do sistema pode ser visto como uma avaliação do sistema que captura os valores para todas as variáveis em V em um instante particular do tempo. Por exemplo, seja $V = \{v_1, v_2, v_3\}$ e $D_1 = D_2 = D_3 = \{0, 1\}$, $(v_1 = 0, v_2 = 0, v_3 = 0)$, $(v_1 = 0, v_2 = 0, v_3 = 1)$ e $(v_1 = 0, v_2 = 1, v_3 = 1)$ são exemplos de estados possíveis do sistema, cujas representações em fórmulas booleanas (representação simbólica) são, respectivamente $(\neg v_1) \wedge (\neg v_2) \wedge (\neg v_3)$, $(\neg v_1) \wedge (\neg v_2) \wedge (v_3)$ e $(\neg v_1) \wedge (v_2) \wedge (v_3)$, em que v_i significa que a variável v_i assume o valor 1 e $\neg v_i$ que a variável v_i assume o valor 0.

Além de estados, é preciso também representar as transições usando fórmulas booleanas. Para isso, um segundo conjunto de variáveis V' é criado para representar as variáveis do próximo estado, sendo que o conjunto V representa as variáveis do estado corrente. Para cada $v \in V$ cria-se uma variável de próximo estado $v' \in V'$. Uma tran-

sição pode ser vista como um par ordenado, formado pela avaliação do sistema para as variáveis em V e V' , e que pode ser representado usando fórmulas como anteriormente. Se f é a fórmula para o estado corrente s e f' é a fórmula para o próximo estado s' , então a transição do estado s para o estado s' é representada pela fórmula $f \wedge f'$. Por exemplo, a transição de um estado onde $(v_1 = 0, v_2 = 0, v_3 = 0)$ para um estado onde $(v_1 = 0, v_2 = 0, v_3 = 1)$ (Figura 3.2) é representada pela fórmula proposicional $(\neg v_1) \wedge (\neg v_2) \wedge (\neg v_3) \wedge (\neg v'_1) \wedge (\neg v'_2) \wedge (v'_3)$.

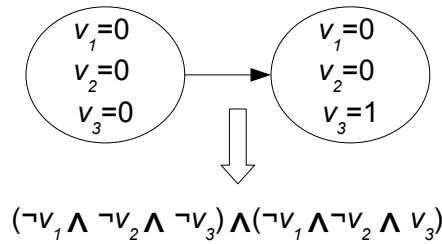


Figura 3.2. Exemplo de uma transição e sua representação simbólica.

Fórmulas podem representar conjunto de estados e conjuntos de transições. Se f_1, f_2, \dots, f_n representam todas as transições de uma estrutura de Kripke, a fórmula para o conjunto de todas as transições é descrita pela fórmula $f_R = f_1 \vee f_2 \vee \dots \vee f_n$. O mesmo raciocínio é utilizado para compor o conjunto de todos os estados da estrutura de Kripke.

O agrupamento de transições em uma fórmula simples, que simplifica o processo de caminhamento no grafo, acontece automaticamente quando as fórmulas são implementadas usando BDDs. Isso é uma das principais razões para a eficiência de algoritmos simbólicos.

Além disso, é preciso definir um conjunto AP de proposições atômicas de modo a escrever especificações que descrevem propriedades dos sistemas. Uma proposição atômica é uma expressão que tem a forma $v = d$, onde $v \in V$ e $d \in D$. Uma proposição $v = d$ será verdade em um estado s , se a avaliação da fórmula que representa o estado torna-se verdade quando v assume o valor d .

Para ilustrar o processo descrito nesta seção, a representação simbólica da estrutura de Kripke da Figura 3.1 é apresentada. Nesse caso, considere $V = \{A, B\}$ e $D_A = D_B = \{0, 1\}$ para o modelo. Além disso, é preciso adicionar duas novas variáveis $V' = \{A', B'\}$ para codificar estados sucessores. A representação simbólica do estado \mathcal{S}_0 é dada pela fórmula $A \wedge B$ e do estado \mathcal{S}_1 pela fórmula $A \wedge \neg B$. Finalmente, a transição do estado \mathcal{S}_0 para o estado \mathcal{S}_1 é dada por $\mathcal{R}(\mathcal{S}_0, \mathcal{S}_1) \equiv A \wedge B \wedge A' \wedge \neg B'$.

A fórmula booleana que representa a relação de transição completa do modelo contém três disjunções, representando o número de transições que a estrutura de Kripke possui:

$$(A \wedge B \wedge A' \wedge \neg B') \vee (A \wedge \neg B \wedge A' \wedge B') \vee (A \wedge \neg B \wedge A' \wedge \neg B')$$

Nesse exemplo, a função L possui os seguintes mapeamentos: $L(\mathcal{S}_0) = \{A = 1, B = 1\}$ e $L(\mathcal{S}_1) = \{A = 1, B = 0\}$. Como $A = 1$ e $B = 1 \in L(\mathcal{S}_0)$, quando A e B assumem, respectivamente, os valores 1 e 1 no estado S_0 a fórmula que representa este estado $(A \wedge B)$ é verdadeira.

As definições anteriores foram realizadas considerando um domínio binário único para todas as variáveis $D = \{0, 1\}$. Entretanto, é possível converter elementos de um domínio D qualquer para trabalhar sempre com variáveis binárias. A conversão é feita por utilizar j bits para codificar o domínio D_i de cada variável v_i através de um número binário e representar v_i como j variáveis booleanas. Por exemplo, se assumirmos que as variáveis v_i assumem somente valores inteiros no domínio $D = \{0, 1, 2, 3, 4, 5, 6, 7\}$, três bits são necessários para codificar cada variável v_i . Dessa forma, o predicado $(v_1 = 2)$ pode ser representado pela conjunção de três novas variáveis booleanas $(v_{1.1} = 0 \wedge v_{1.2} = 1 \wedge v_{1.3} = 0)$, sendo cada uma delas a representação de um bit da codificação binária de 2 (010). Por fim, a fórmula $(v_1 = 2) \wedge (v_2 = 3) \wedge (v_3 = 4)$ seria representada em um domínio binário como $(v_{1.1} = 0 \wedge v_{1.2} = 1 \wedge v_{1.3} = 0) \wedge (v_{2.1} = 0 \wedge v_{2.2} = 1 \wedge v_{2.3} = 1) \wedge (v_{3.1} = 1 \wedge v_{3.2} = 0 \wedge v_{3.3} = 0)$

3.1.3 Diagramas de Decisão Binária

Diagrama de Decisão Binária (do inglês, *Binary Decision Diagram - BDD*) é uma representação compacta para fórmulas booleanas, por descartar informações redundantes, e que pode ser manipulada de maneira eficiente. Sua origem vem da representação de funções booleanas, descritas anteriormente, através de árvores de decisão binária, que é uma árvore direcionada com dois tipos de vértices: terminais e não-terminais.

Em uma árvore de decisão binária, um vértice v não-terminal é rotulado com uma variável booleana, dada por $var(v)$, e possui dois sucessores: $zero(v)$, quando $var(v) = 0$, e $um(v)$, quando $var(v) = 1$. Um vértice terminal v assume somente o valor 0 ou 1, dado pela função $valor(v)$. As arestas da árvore são rotuladas com o valor 0 ou 1. A Figura 3.3 mostra um exemplo de uma árvore de decisão binária para a fórmula booleana $(a \wedge b) \vee (c \wedge d)$. Um caminho através da árvore inicia do vértice raiz e a escolha de $zero(v)$ ou $um(v)$ conduzirá para um vértice terminal que representa a avaliação da fórmula. No exemplo, a atribuição $(a = 1, b = 0, c = 1, d = 0)$ leva a um

vértice terminal com valor 0, o que significa que a fórmula booleana é falsa para essa atribuição.

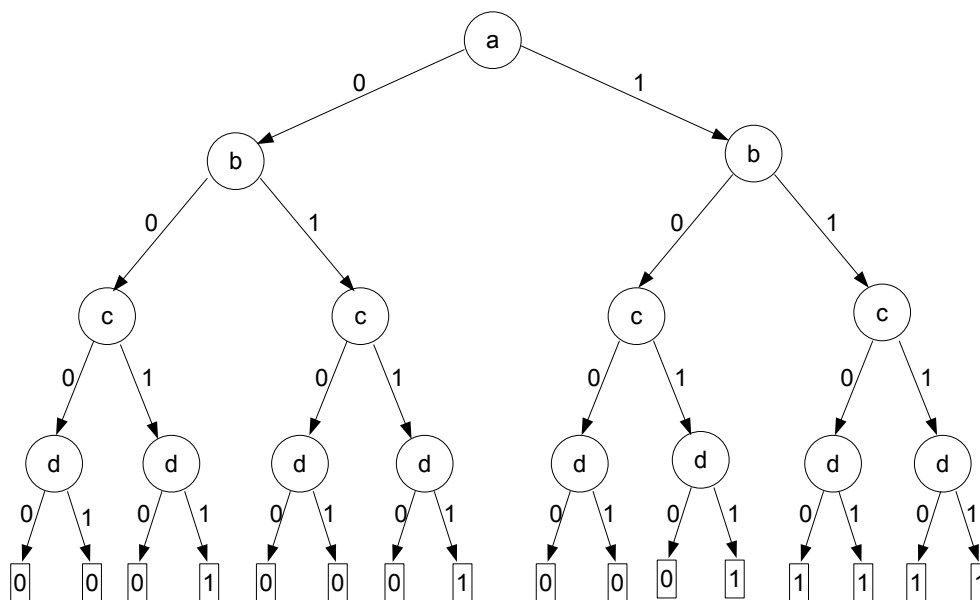


Figura 3.3. Árvore de decisão binária para a fórmula booleana $(a \wedge b) \vee (c \wedge d)$.

As árvores de decisão binária possuem muitas informações redundantes. Por exemplo, na Figura 3.3 existem 8 subárvores cujas raízes são rotuladas com a variável booleana **d**. Entretanto, somente três delas são distintas: na primeira subárvore, a atribuição tanto do valor 1 quanto do valor 0 à variável **d** conduzirá a um terminal com valor 0; na segunda subárvore, a atribuição de 1 ou de 0 à variável **d** sempre conduzirá a um terminal com valor 1; por fim, na terceira subárvore, a atribuição de 0 à variável **d** leva a um terminal com valor 0, enquanto que a atribuição de 1 à variável **d** leva a um terminal com valor 1.

Assim, basicamente, o BDD é obtido por fundir subárvores idênticas e eliminar nodos com informação repetida. A estrutura resultante é um grafo acíclico direcionado chamado de BDD que, ao contrário da árvore, permite que vértices e subestruturas sejam compartilhados. Vale ainda enfatizar que os BDDs são representações canônicas para fórmulas booleanas. Isso significa que duas fórmulas são equivalentes se, e somente se, seus BDDs são isomórficos².

Bryant [1986a] mostrou como obter uma representação canônica para fórmulas booleanas ao impor duas restrições sobre os BDDs. Primeiramente, as variáveis devem

²Dois BDDs são isomórficos se há uma função injetora h que mapeia terminais e não-terminais de um BDD para o outro.

aparecer na mesma ordem ao longo do caminho da raiz até um vértice terminal no diagrama. Em segundo lugar, não devem existir subárvores isomórficas ou vértices redundantes no diagrama. O primeiro requisito é satisfeito por fixar uma ordem total para as variáveis ($var_1 < var_2 < \dots < var_n$) que rotulam os vértices no BDD, ou seja, se o vértice u precede o vértice não-terminal v (partindo da raiz do BDD), então a variável que rotula o vértice u precede a variável que rotula v na ordenação ($var(u) < var(v)$). O segundo requisito é satisfeito ao repetidamente aplicar três regras de transformação que não alteram a função representada pelo diagrama:

1. *Remova terminais duplicados*: mantenha somente um vértice terminal com valor 0 e outro com valor 1. Redirecione todas as arestas de entrada dos demais vértices terminais eliminados para esses dois vértices terminais;
2. *remova não-terminais duplicados*: se dois não-terminais u e v são rotulados com a mesma variável ($var(u) = var(v)$) e têm mesmos sucessores ($zero(u) = zero(v) \wedge um(u) = um(v)$), elimine o vértice v e redirecione todas as suas arestas de entrada para o vértice u ;
3. *remova filhos redundantes*: se um vértice não-terminal v tem $zero(v) = um(v)$, então elimine v e redirecione todos os vértices de entrada de v para $zero(v)$.

A forma canônica do BDD, obtida ao impor as duas restrições descritas anteriormente, é chamada de Diagrama de Decisão Binário Ordenado (do inglês, *Ordinary Binary Decision Diagram - OBDD*). A Figura 3.4 mostra o OBDD para a árvore de decisão da Figura 3.3, considerando a ordem $a < b < c < d$.

O tamanho do OBDD é altamente dependente da ordenação das variáveis e, por isso, seu crescimento pode ser linear ou exponencial em relação ao número de variáveis da fórmula dependendo da ordem escolhida. Além disso, o problema de escolher uma ordenação para as variáveis que minimize o tamanho do OBDD é um problema co NP-completo [Bryant, 1986b]. Apesar disso, várias heurísticas tem sido propostas para gerar automaticamente ordenações que produzam OBDDs com tamanho reduzido. Por exemplo, se a fórmula booleana que representa o conjunto de transições do modelo é dada por um circuito combinacional, então heurísticas baseadas em caminhamento em profundidade geralmente produzem bons resultados. Essas heurísticas se baseiam na observação que OBDDs tendem a ser pequenos quando variáveis relacionadas estão próximas na ordenação. Já a técnica conhecida como reordenamento dinâmico se apresenta como uma técnica útil para as situações em que não é trivial identificar a heurística a ser aplicada. Essa técnica reordena periodicamente as variáveis de maneira a reduzir o número total de vértices em uso [Clarke et al., 1999]. Maiores detalhes sobre

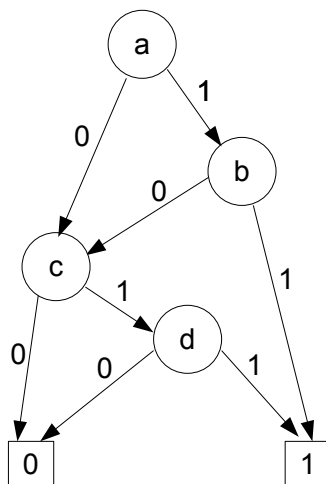


Figura 3.4. OBDD para a fórmula $(a \wedge b) \vee (c \wedge d)$.

heurísticas de ordenação de variáveis podem ser encontrados no trabalho de Grumberg et al. [2003].

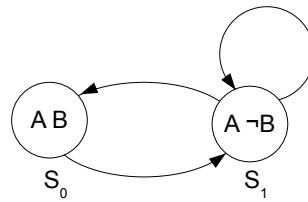
3.1.4 Lógicas Temporais

Lógica temporal é o termo que designa qualquer sistema de regras e símbolos usados para representar e raciocinar sobre proposições qualificadas em termos de tempo, sem contudo explicitamente introduzir essa variável. Ao invés disso, as análises do sistema são realizadas em termos de ocorrência de eventos, como, por exemplo, se um evento pode ocorrer, se ele sempre ocorre ou se ele nunca ocorre.

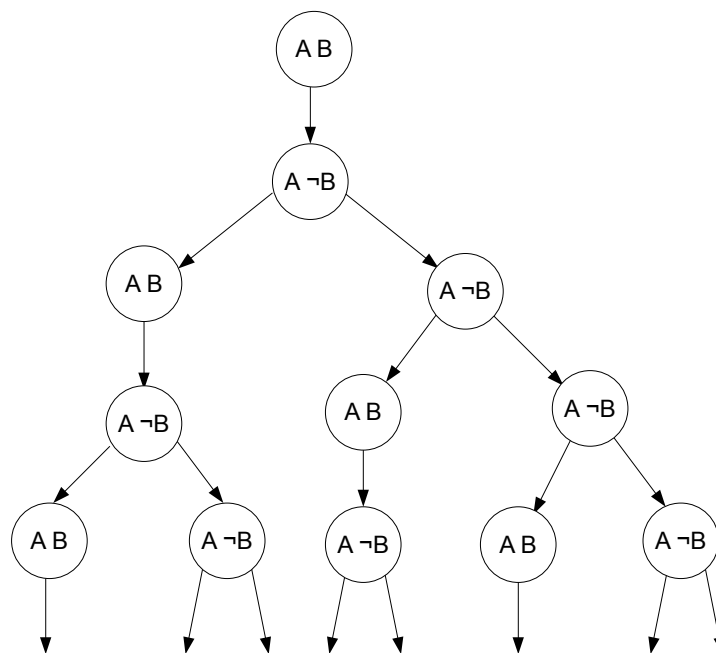
Existem várias lógicas temporais, entretanto, somente as lógicas de tempo linear (do inglês, *Linear-time Temporal Logic* - *LTL*) e de tempo ramificado (do inglês, *Computational Tree Logic* - *CTL*) serão descritas aqui, pois constituem a base para as lógicas que irão ser empregadas no decorrer deste trabalho. Enquanto na lógica LTL os operadores descrevem eventos ao longo de um único caminho infinito, na CTL eles descrevem eventos ao longo de uma árvore de caminhos de profundidade ilimitada cuja raiz é o estado s_0 . Vale a pena ressaltar que as três lógicas possuem expressividade diferentes, sendo que uma fórmula em uma dada lógica pode não ter correspondência em outra.

Tanto LTL quanto CTL são formadas por dois tipos de operadores: quantificadores de caminho e operadores temporais. Quantificadores de caminho são usados para descrever a estrutura ramificada na árvore de computação, que é formada ao desenrolar a estrutura de Kripke a partir de um estado inicial e compor uma árvore infinita como

ilustrado na Figura 3.5.



(a)



(b)

Figura 3.5. (a) Estrutura de Kripke. (b) Expansão da estrutura de Kripke em uma árvore infinita.

Há dois quantificadores de caminho: **A** (para todo caminho) e **E** (para algum caminho). Esses operadores são utilizados em um dado estado para especificar que a propriedade desejada é válida em todos caminhos ou em algum caminho iniciando naquele estado. Na lógica LTL só é permitido o uso do quantificador **A**.

Os operadores temporais são usados para descrever propriedades em um caminho da árvore, sendo os principais deles, listados a seguir:

- **X** f (*next*): requer que a propriedade f seja verdadeira no próximo estado do

caminho;

- **F** f (*eventually*): é usada para garantir que a propriedade f é válida em algum estado no caminho;
- **G** f (*always*): requer que a propriedade f seja verdadeira em todos estados do caminho;
- f **U** g (*until*): requer que a propriedade f seja verdadeira até que g se torne verdade. A propriedade g deve ser verdade em algum estado do caminho.

3.1.4.1 Lógica Temporal CTL

Na lógica CTL os operadores temporais **X**, **F**, **G** e **U** devem ser precedidos por um quantificador de caminho. Na verdade, esses dois operadores devem sempre aparecer em pares. Um exemplo de uma fórmula CTL é **AG(EF p)**: de qualquer estado é possível atingir um estado onde p é verdade.

A Figura 3.6 ilustra a semântica, em termos da árvore de computação obtida da estrutura de Kripke, de quatro fórmulas CTL **AG p**, **AF p**, **EG p** e **EF p**, sendo p uma proposição atômica.

3.1.4.2 Lógica Temporal LTL

A lógica LTL consiste de fórmulas que possuem a forma **A** f , onde f é uma fórmula de caminho, que tem a seguinte sintaxe em LTL:

- **true**, **false** ou p , sendo $p \in AP$ e AP o conjunto de proposições atômicas;
- $\neg f$, $f \vee g$, $f \wedge g$, **X** f , **F** f , **G** f , f **U** g , sendo f e g fórmulas de caminho.

Um exemplo de uma fórmula LTL é **A(FG p)**: ao longo de todos caminhos, existe algum estado a partir do qual p é sempre verdade. A Figura 3.7 ilustra a semântica, em termos de alguns caminhos da árvore de computação obtida da estrutura de Kripke, de quatro fórmulas LTL **AX p**, **AF p**, **AG p** e **A(p U q)**, sendo p e q proposições atômicas.

3.2 Verificação Probabilística de Modelos

Recentemente há um interesse crescente na aplicação da técnica de verificação de modelos em biologia, como é apresentado nos trabalhos de Kwiatkowska et al. [2008], Calder et al. [2006] e Heath et al [2008]. Sistemas biológicos, entretanto, apresentam algumas peculiaridades: necessitam de modelos altamente parametrizados, porque é

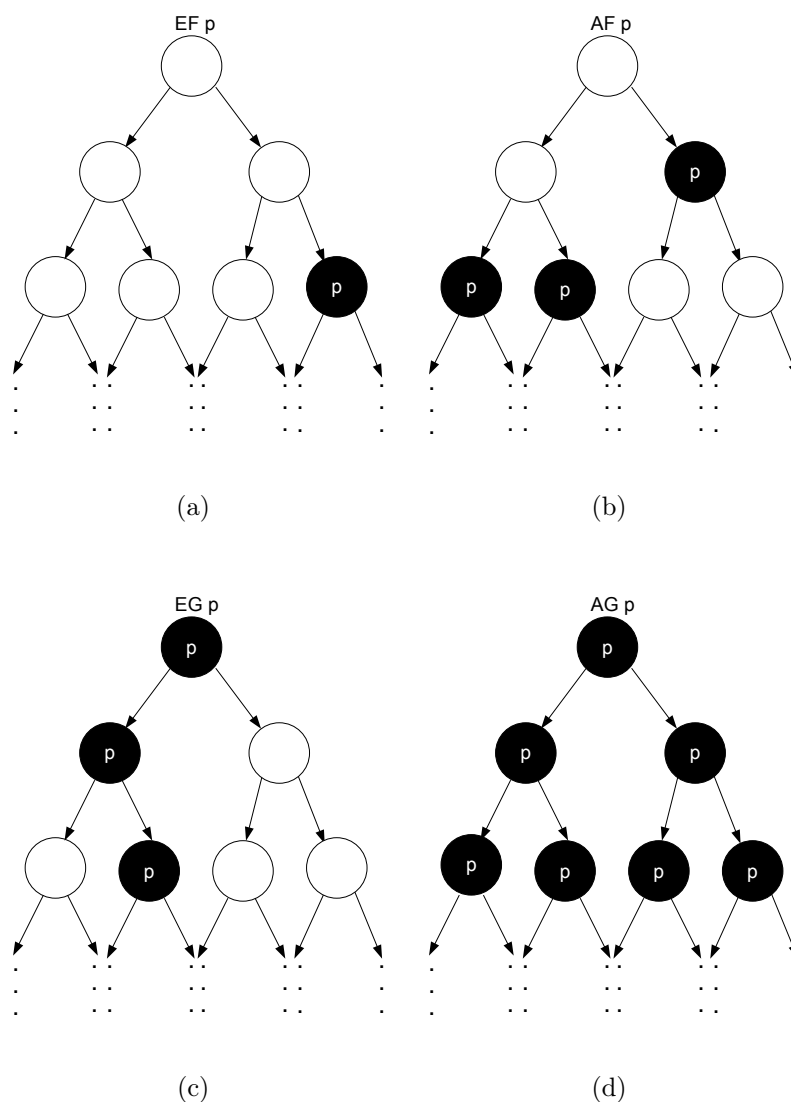


Figura 3.6. Operadores CTL básicos.

importante para biólogos determinar se uma dada propriedade é ou não sensível a um conjunto de condições iniciais; e são inerentemente estocásticos, ou seja, apresentam aspectos aleatórios, sendo o estudo das dinâmicas dos mesmos realizado em termos probabilísticos. Em um sistema estocástico ou aleatório, existem várias maneiras do mesmo evoluir no tempo (como oposto, por exemplo, a solução de uma equação diferencial ordinária), havendo indeterminações nos estados futuros do sistema, descritas por distribuições de probabilidade. Isto significa que mesmo se a condição inicial (ou ponto de partida) é conhecido, existem muitas possibilidades para o futuro do sistema, mas alguns caminhos podem ser mais prováveis e outros menos.

Nesse contexto, a Verificação Probabilística de Modelos (do inglês, *Probabilistic Model Checking - PMC*) se apresenta como uma técnica variante de verificação de modelos

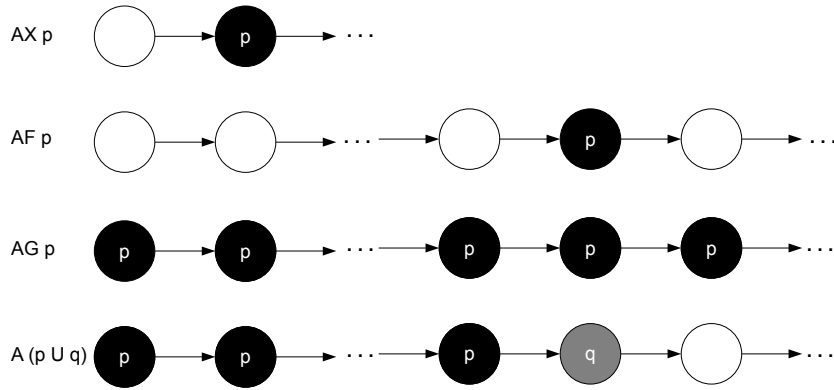


Figura 3.7. Operadores LTL básicos.

utilizada para modelar e analisar sistemas que exibem comportamento estocástico ou aleatório [Hartonas-Garmhausen et al., 1999]. PMC requer, assim como a versão não probabilística da técnica, dois argumentos: a descrição do sistema em algum formalismo de alto nível e a especificação de uma ou mais propriedades desejadas do sistema em lógica temporal. Entretanto, diferente da versão não-probabilística, agora o modelo é estocástico e as propriedades de interesse são expressas de uma maneira quantitativa e não mais somente qualitativa. Por exemplo, ao invés de perguntar 'A espécie A eventualmente reage com a espécie B ?' o interesse é saber 'Qual é a probabilidade da espécie A reagir com a espécie B ?'.

A descrição estocástica apropriada para sistemas biológicos é usualmente uma Cadeia de Markov de Tempo Contínuo (do inglês, *Continuous Time Markov Chain - CTMC*) [Kwiatkowska et al., 2008], em que há um grafo do sistema e taxas positivas e reais são atribuídas às transições entre os estados (nodos do grafo). Esses valores são interpretados como taxas de distribuições exponenciais negativas.

Formalmente, seja $\mathbb{R}_{\geq 0}$ o conjunto dos valores reais não negativos e AP um conjunto finito de proposições atômicas usadas para nomear os estados com propriedades de interesse. Um modelo CTMC é uma tupla (S, S_0, R, L) onde:

1. S é o conjunto finito de todos os estados do sistema;
2. $S_0 \in S$ é o conjunto dos estados iniciais do sistema;
3. $R : (S \times S) \rightarrow \mathbb{R}_{\geq 0}$ é uma matriz de taxas de transição;
4. $L : S \rightarrow 2^{AP}$ é uma função que associa cada estado com o conjunto de proposições atômicas verdadeiras naquele estado.

A matriz R de taxas de transição atribui taxas para cada par de estados. Uma transição pode ocorrer entre os estados s e s' se $R(s, s') > 0$. Pode existir uma condição

de corrida, dado que pode haver um estado s em que há mais que um estado s' para o qual $R(s, s') > 0$. Neste caso, a probabilidade de ir de um estado s para um estado s' dentro de t unidades de tempo é definida como Clarke et al. [2008]:

$$P(s, s', t) = \frac{R(s, s')}{E(s)} \times (1 - e^{-E(s) \times t}) \quad (3.1)$$

onde $E(s) = \sum_{s' \in S} R(s, s')$. Na verdade, $E(s)$ é a taxa da distribuição exponencial do tempo gasto no estado s antes de uma transição ocorrer, ou seja, a probabilidade de deixar s dentro de t unidades de tempo é $1 - e^{-E(s) \times t}$. Consequentemente, a probabilidade da transição do estado s para o estado s' ocorrer é dada por:

$$P(s, s') = \frac{R(s, s')}{E(s)} \quad (3.2)$$

Um caminho em um modelo CTMC a partir de um estado inicial s_0 é uma sequência infinita $\pi = s_0 t_0 s_1 t_1 s_2 t_2 \dots$, onde $s_i \in S$, $t_i \in \mathbb{R}_{\geq 0}$ é o tempo gasto no estado s_i e $R(s_i, s_{i+1}) > 0$ para todo $i \geq 0$.

A Figura 3.8 mostra o grafo de um modelo CTMC em que:

1. $S = \{\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2\}$;
2. $S_0 = \{\mathcal{S}_0\}$;
3. $R(\mathcal{S}_0, \mathcal{S}_1) = \lambda_1$, $R(\mathcal{S}_1, \mathcal{S}_2) = \lambda_2$, $R(\mathcal{S}_2, \mathcal{S}_0) = \lambda_3$ e $R(\mathcal{S}_1, \mathcal{S}_0) = \lambda_4$;
4. $L(\mathcal{S}_0) = \{A, B\}$, $L(\mathcal{S}_1) = \{A, \neg B\}$ e $L(\mathcal{S}_2) = \{\neg A, B\}$.

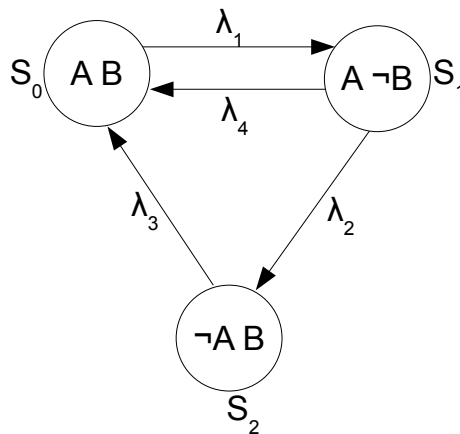


Figura 3.8. Exemplo de modelo CTMC.

Empregando a Equação (3.1), é possível obter para o modelo CTMC da Figura 3.8 a probabilidade de ir do estado \mathcal{S}_1 para o estado \mathcal{S}_2 dentro de t unidades de tempo:

$$P(\mathcal{S}_1, \mathcal{S}_2, t) = \frac{\lambda_2}{(\lambda_2 + \lambda_4)} \times (1 - e^{-(\lambda_2 + \lambda_4) \times t}) \quad (3.3)$$

3.3 Ferramentas Utilizadas

Neste trabalho foram empregadas duas ferramentas principais na modelagem e análise dos sistemas de transporte de íons em membranas: PRISM [Hillston et al., 2006] e BIOLAB [Clarke et al., 2008]. Enquanto a primeira pertence ao grupo de algoritmos que utilizam métodos numéricos para computar a probabilidade exata do sistema satisfazer a propriedade desejada, a segunda é integrante de um outro grupo de PMC aproximados ou estatísticos, que trabalham com amostragens de caminhos do modelo e com a aplicação da estatística para decidir se a propriedade especificada é satisfeita ou não. Embora consiga gerar resultados mais precisos que os algoritmos aproximados, os algoritmos exatos geralmente requerem uma maior quantidade de tempo e espaço de memória, sendo um desafio lidar com grandes sistemas. Segundo Jha et al. [2009] estes algoritmos podem tratar sistemas entre 10^6 e 10^7 estados.

A ferramenta PRISM, desenvolvida na Universidade de Birmingham e Oxford, vem sendo bastante empregada na modelagem de sistemas biológicos [Hillston et al., 2006; Pronk et al., 2007]. Possui um grande conjunto de recursos disponíveis, sendo bastante completa, além de contar com um suporte satisfatório em www.prismmodelchecker.org.

Por outro lado, a motivação para a utilização da ferramenta BIOLAB, desenvolvida na Universidade de Carnegie Mellon, é por ser uma ferramenta ótima no sentido de gerar o menor número possível de caminhos do modelo para verificar a propriedade desejada, quando comparada com outras abordagens de verificação aproximadas. Além disso ela utiliza uma linguagem de descrição do modelo bem mais intuitiva do ponto de vista biológico.

Por fim, a comparação entre as duas ferramentas é outro aspecto motivador, dado que ambas empregam metodologias distintas para decidir se a propriedade de interesse é satisfeita ou não, uma vez que o modelo do sistema está especificado.

As seções seguintes descrevem detalhadamente como os modelos devem ser descritos e as propriedades de interesse especificadas em cada uma delas, bem como brevemente introduz os algoritmos de verificação empregados por elas. Para ilustrar o processo de análise e modelagem utilizando cada uma das ferramentas, um simples conjunto de reações químicas mostrado na Figura 3.9 será modelado. Na primeira reação química, a molécula A e a molécula B podem formar um complexo com taxa r_1 e a separação do complexo formado pode ocorrer com taxa r_2 . Além disso, a molécula A pode se degradar com taxa r_3 .

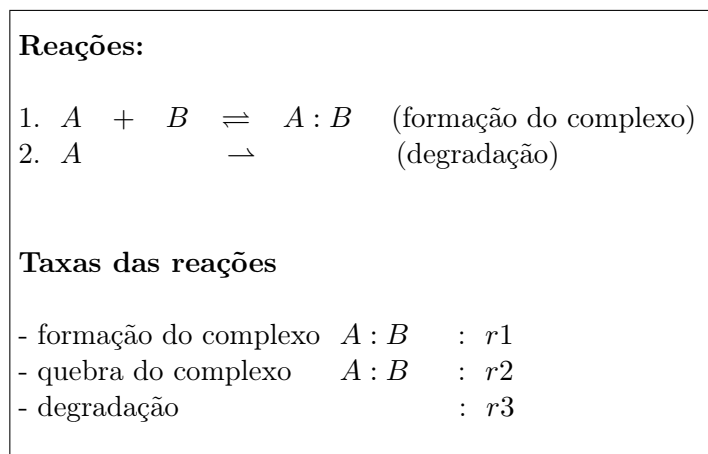


Figura 3.9. Exemplo de um sistema de reações químicas

3.3.1 PRISM

3.3.1.1 Descrição do Modelo

Uma das formas de modelar o conjunto de reações descrito anteriormente na ferramenta PRISM é apresentado no código da Figura 3.10.

A descrição de um modelo CTMC no PRISM, como pode ser observado, deve iniciar com a palavra chave **ctmc** e compreende a um conjunto de módulos, cujos estados são representados por um conjunto de variáveis que assumem valores finitos. No exemplo considerado, há um módulo para cada substrato do conjunto de reações: **m_A**, **m_B** e **m_AB**. Para cada módulo existe uma variável que descreve se aquele substrato está presente (1 - neste primeiro modelo pode existir, no máximo, uma molécula de cada um dos três substratos) ou não (0). Inicialmente, somente as moléculas *A* e *B* estão presentes. Adicionalmente, o módulo **start** é responsável por definir as constantes de taxas para cada uma das três reações possíveis no sistema: **bind** (ligação entre as moléculas *A* e *B*), **rel** (quebra do complexo *AB*) e **deg** (degradação da molécula *A*).

O comportamento dos módulos, ou seja, as transições entre os estados, são especificadas pelos comandos do tipo $[action]g \rightarrow r : u$, que indica que se o predicado *g* for verdadeiro, o sistema será atualizado segundo *u*, que, por sua vez, compreende a uma ou mais declarações da forma $(x' = \dots)$ indicando como o valor da variável *x* é atualizado (*x'* é o valor da variável *x* no próximo estado). O valor da taxa em que as alterações irão ocorrer é determinado por *r*. PRISM também suporta sincronização entre módulos, através dos rótulos dos comandos que devem estar entre os colchetes iniciais do mesmo. Transições em módulos diferentes e com o mesmo rótulo ocorrem simultaneamente. A taxa resultante, neste caso, é igual ao produto das taxas individuais de cada comando nos diferentes módulos que sincronizam. No exemplo, a formação

```

ctmc

const double r1=1;
const double r2=1;
const double r3=0.1;

module m_A
a: [0..1] init 1; // 0 - degradada ou ligada, 1 - livre

[bind] a=1 -> 1 : (a'=0);
[rel] a=0 -> 1 : (a'=1);
[deg] a=1 -> 1 : (a'=0);

endmodule

module m_B
b: [0..1] init 1; // 0 - ligada, 1 - livre

[bind] b=1 -> 1 : (b'=0);
[rel] b=0 -> 1 : (b'=1);

endmodule

module m_AB
ab: [0..1] init 0; // 0 - complexo não existe, 1 - complexo formado

[bind] ab=0 -> 1 : (ab'=1);
[rel] ab=1 -> 1 : (ab'=0);

endmodule

module start
[bind] true -> r1 : true;
[rel] true -> r2 : true;
[deg] true -> r3 : true;

endmodule

```

Figura 3.10. Modelo no PRISM para o sistema de reações da Figura 3.9.

do complexo AB (dada pela reação $A + B \xrightarrow{r_1} AB$) é representada pelos comandos rotulados com **bind**. Para que a reação ocorra, as moléculas A e B devem estar presentes (terem o valor 1) e o complexo AB estar ausente (ter o valor 0). A taxa final quando esta reação acontece é $r_1 * 1 * 1 * 1$ e é dada pelo produto das taxas dos quatro comandos no modelo rotulados com **bind**, um em cada um dos módulos. De maneira similar, a quebra do complexo AB , liberando as moléculas A e B , é representada pelos comandos rotulados com **rel** e a degradação da molécula A pelo comando rotulado com **deg**.

Além disso, PRISM permite estender modelos CTMCs com *rewards*, estruturas que possibilitam a marcação de estados ou transições com valores reais que podem ser utilizados pelas propriedades para contar quantidades. Existem dois tipos de *rewards*, como já mencionado: os associados aos estados, que são acumulados proporcionalmente ao tempo gasto nos mesmos; ou os associados às transições, que são acumulados cada

vez que a transição ocorre. Através desse mecanismo é possível saber, por exemplo, o tempo esperado para que um dado evento ocorra. Cada *reward* normalmente tem um nome associado a ele, que é utilizado na especificação das propriedades de interesse, e possui a seguinte sintaxe:

$$\mathbf{rewards} \text{ 'reward_nome' } reward_corpo;^+ \mathbf{endrewards}$$

onde *reward_corpo* tem a forma $g : v$, quando o *reward* é associado a um estado, ou $[action]g : v$, quando ele é associado a uma transição. No primeiro caso, todo estado que satisfaz o predicado g é marcado com o valor real v . No segundo caso, toda transição rotulada com *action*, a partir de estados que satisfaçam o predicado g , é marcada com o valor v .

O código da Figura 3.11 apresenta a definição de três estruturas de *reward* para o exemplo do sistema de reações químicas sendo considerado. A primeira delas, nomeada **time**, associa o valor 1 a todo estado no sistema. A segunda (**freeA**) associa o valor 1 somente aos estados em que a molécula *A* está livre. Por fim, o *reward* **bind** associa o valor 1 a toda transição rotulada com **bind**, ou seja, toda vez que a molécula *A* se liga à *B*.

```

rewards 'time'
true : 1;
endrewards

rewards 'freeA'
a=1 : 1;
endrewards

rewards 'bind'
[bind] true : 1;
endrewards

```

Figura 3.11. Rewards para modelo PRISM da Figura 3.10.

3.3.1.2 Especificação de Propriedades

As propriedades para verificar modelos CTMCs em PRISM devem ser especificadas na lógica estocástica de tempo contínuo (do inglês, *Continuous Stochastic Logic - CSL*) [Kwiatkowska et al., 2008], que é uma lógica temporal baseada em CTL e em PCTL (do inglês, *Probabilistic Computation Tree Logic*). Segue a sintaxe das fórmulas CSL:

$$\Phi ::= true \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{P}_{\leq\theta}[\phi] \mid \mathcal{S}_{\leq\theta}[\phi]$$

$$\phi ::= \mathbf{X}\Phi \mid \Phi_1 \mathbf{U}^I \Phi_2$$

onde a é uma proposição atômica, $\theta \in [0, 1]$ corresponde a uma probabilidade e I é um intervalo de $\mathbb{R}_{\geq 0}$ (números reais não negativos) em que a propriedade deve ser satisfeita. Os operadores \neg e \wedge são operadores lógicos e \mathbf{X} e \mathbf{U} são operadores temporais. O símbolo $\trianglelefteq \in \{>, <, \geq, \leq\}$ representa o limite que a probabilidade fornecida deve satisfazer. Por exemplo, se \trianglelefteq é $>$, a probabilidade real da propriedade ser satisfeita deve ser maior que θ .

Há dois tipos básicos de propriedades em CSL: transiente ($\mathcal{P}_{\trianglelefteq\theta}$) e em estado de equilíbrio ($\mathcal{S}_{\trianglelefteq\theta}$). Descreveremos unicamente a semântica das propriedades transientes porque somente as mesmas serão utilizadas ao longo deste trabalho. A fórmula $\mathcal{P}_{\trianglelefteq\theta}[\phi]$ é verdade no estado s se a probabilidade que ϕ seja satisfeito por um caminho iniciando no estado s case com o limite $\trianglelefteq\theta$. Fórmulas de caminho são construídas usando os operadores \mathbf{X} (*next*) e \mathbf{U}^I (*time-bounded until*). A fórmula de caminho $\mathbf{X}\Phi$ é verdade somente se Φ é satisfeito no próximo estado, enquanto $\Phi_1\mathbf{U}^I\Phi_2$ é verdade somente se Φ_2 é satisfeito em algum instante no intervalo I e em todos os instantes de tempo anteriores Φ_1 é satisfeito. Outros operadores podem ser derivados deste conjunto mínimo de operadores CSL. Dois deles, que serão comumente utilizados no trabalho, são os operadores $\mathbf{F}^I\Phi$ (*time-bounded eventually*), que é verdade se Φ é satisfeito em algum instante de tempo no intervalo I , e o operador $\mathbf{G}^I\Phi$ (*time-bounded always*), que é verdade se Φ é satisfeito em todo instante do tempo no intervalo I . Vale a pena lembrar que o intervalo I pode ser omitido nos operadores \mathbf{U} , \mathbf{F} e \mathbf{G} , o que significa que $I = [0, \infty)$. Adicionalmente, se ao invés de especificar o limite $\trianglelefteq\theta$ a expressão $=?$ for utilizada ($\mathcal{P}_{=?}[\phi]$), a ferramenta PRISM calcula a probabilidade da propriedade ϕ ser satisfeita.

Seguem alguns exemplos de propriedades transientes considerando o modelo PRISM, definido na seção anterior, para o sistema de reações apresentado na Figura 3.9:

- $\mathcal{P}_{=?}[\mathbf{F}^{[0,T]}ab = 1]$: a probabilidade que a molécula A se ligue à B nas primeiras T unidades de tempo. Aqui, a propriedade é do tipo $\mathbf{F}^I\Phi$ (*eventually*), sendo Φ a proposição atômica $ab = 1$;
- $\mathcal{P}_{=?}[ab = 0\mathbf{U}(a = 0 \wedge ab = 0)]$: a probabilidade que a molécula A se degrade antes de se ligar a molécula B . A propriedade é do tipo $\Phi_1\mathbf{U}^I\Phi_2$ (*time-bounded until*), sendo $I = [0, \infty)$.

Além disso, PRISM também permite verificar o valor esperado de *rewards* definidos no modelo. Algumas das propriedades deste tipo que serão usadas ao longo do trabalho têm a forma $\mathcal{R}_{\trianglelefteq r}[I = t]$, $\mathcal{R}_{\trianglelefteq r}[F\Phi]$ e $\mathcal{R}_{\trianglelefteq r}[C \leq t]$, com r e $t \in \mathbb{R}_{\geq 0}$. A primeira propriedade é verdade, partindo de um estado s , se o *reward* de estado no instante

t satisfaz o limite $\leq r$. A segunda propriedade, por sua vez, é verdade, partindo de s , se o *reward* acumulado ao longo de um caminho até o ponto onde Φ é verdade satisfaz o limite $\leq r$. Por fim, a terceira propriedade é verdade, partindo de s , se o *reward* acumulado ao longo de um caminho até o instante t satisfaz o limite $\leq r$. Dado a definição de um *reward*, o seu valor acumulado ao longo de um caminho em um modelo CTMC é a soma dos *rewards* de estado ao longo do caminho, mais a soma dos *rewards* de transição entre estes estados, ambos definidos no corpo da mesma estrutura do *reward*. O *reward* de estado atribuído a cada estado é $v \times t$, onde t é o tempo gasto no estado e v o *reward* de estado atribuído ao estado. Da mesma forma que para propriedades transientes, se o limite $\leq r$ não for especificado (utiliza-se a expressão $=?$) a ferramenta PRISM calcula o valor esperado do *reward*.

Alguns exemplos de propriedades que obtêm valores de *rewards* para o exemplo sendo considerado, dado os *rewards* definidos no código da Figura 3.11, seguem:

- $\mathbb{R}\{freeA'\}_{=?}[C \leq T]$: o tempo esperado que a molécula A gasta no estado livre (ela não está associada a molécula B , nem se degradou) durante as T primeiras unidades de tempo;
- $\mathbb{R}\{bind'\}_{=?}[F(a = 0 \wedge ab = 0)]$: o número esperado de vezes que a molécula A se liga à B antes de se degradar;
- $\mathbb{R}\{time'\}_{=?}[F(a = 0 \wedge ab = 0)]$: o tempo esperado para que a molécula A se degrade.

3.3.1.3 Implementação

As técnicas que são implementadas em PRISM para verificar as propriedades de modelos CTMCs com *rewards* incluem algoritmos da teoria de grafos e computação numérica. Os primeiros operam sobre a estrutura do grafo que representa a cadeia de Markov especificada na ferramenta para determinar, por exemplo, o conjunto de estados alcançáveis ou para verificar propriedades qualitativas. Neste caso, os algoritmos são executados sobre BDDs como ocorre na versão não-probabilística da técnica de verificação de modelos.

A computação numérica é requerida para resolver a cadeia de Markov e calcular as probabilidades e valores de *rewards* (propriedades quantitativas). Métodos iterativos como Jacobi e Gauss-Siedel são utilizados para resolver sistemas de equações lineares e verificar propriedades do tipo $\mathcal{S}_{\leq \theta}$, $\mathcal{P}_{\leq \theta}[\Phi_1 \mathbf{U} \Phi_2]$ e $\mathcal{R}_{\leq r}[F \Phi]$. Por outro lado, o método iterativo conhecido como uniformização, muito eficiente para análise transiente de modelos Markovianos, é usado para computar *rewards* e probabilidades

para as propriedades que envolvem um intervalo de tempo I ou um tempo específico t : $\mathcal{P}_{\leq \theta} [\Phi_1 \mathbf{U}^I \Phi_2]$, $\mathcal{R}_{\leq r} [I = t]$ e $\mathcal{R}_{\leq r} [C \leq t]$. Maiores detalhes sobre essas técnicas para resolver cadeias de Markov podem ser encontradas em [de Souza et al., 1992].

Além disso, para determinar as propriedades quantitativas utilizando computação numérica, o PRISM permite a utilização de três representações de dados:

1. uma generalização de BDDs, conhecida como *Multi-terminal BDDs* (MTBDDs) para representar funções com valores reais, dado que matrizes e vetores reais são requeridos. Essas estruturas de dados permitem representações compactas e manipulação eficiente de grandes modelos por explorarem suas regularidades. Entretanto, a computação é frequentemente lenta;
2. representação explícita, como matriz esparsa, que permite uma computação mais direta e rápida, entretanto, não consegue lidar com grandes modelos;
3. uma abordagem híbrida, que estende os MTBDDs, permitindo computações mais rápidas quando comparadas à representação que usa a versão original de MTBDDs. Nesse caso, MTBDDs são empregados para representar somente a matriz de transição, sendo os vetores soluções dos métodos iterativos representados por vetores reais tradicionais.

A matriz de adjacência a seguir ilustra a representação explícita para as transições do modelo CTMC da Figura 3.12 (a):

$$\begin{pmatrix} 2 & 5 & - & - \\ 2 & 5 & - & 7 \\ - & - & - & - \\ - & 7 & - & - \end{pmatrix}$$

Considere que a primeira linha e a primeira coluna da matriz possuem índices 0. Adicionalmente a entrada $M(l, c)$ da matriz, indica o valor contido na linha l e coluna c . Por exemplo, o valor da entrada $M(0, 0)$ é 2, enquanto o valor da entrada $M(1, 3)$ é 7.

Já a Figura 3.12 (b) mostra um exemplo de um MTBDD que representa as transições do modelo CTMC da Figura 3.12 (a).

Na representação através do MTBDD, as variáveis binárias x_1 e x_2 codificam os índices das linhas da matriz de transição, enquanto y_1 e y_2 codificam os índices das colunas. Por exemplo, para a entrada $M(1, 3)$ da matriz de transição, o índice 1 da linha é codificado através da representação binária $(x_1, x_2) = (0, 1)$ e o índice 3 da coluna é codificado como $(y_1, y_2) = (1, 1)$. Assim, ao seguir o caminho $x_1 = 0, y_1 = 1,$

$x_2 = 1$ e $y_2 = 1$ no MTBDD (note que a ordem das variáveis é $x_1 < y_1 < x_2 < y_2$) chega-se ao nodo terminal 7 que é o valor na matriz de transição para a entrada $M(1, 3)$.

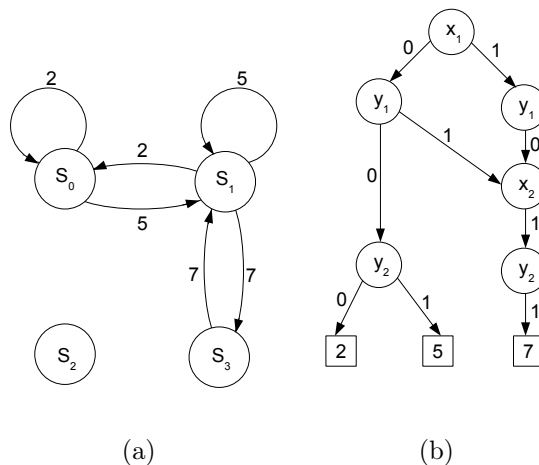


Figura 3.12. (a) Exemplo de um modelo CTMC. (b) MTBDD para o modelo CTMC.

Mais detalhes sobre as técnicas utilizadas pela ferramenta PRISM podem ser encontrados em [Kwiatkowska et al., 2004] e [Kwiatkowska et al., 2007].

3.3.2 BIOLAB

3.3.2.1 Descrição do Modelo

Os modelos a serem verificados na ferramenta BIOLAB devem ser descritos na linguagem BIONETGEN [Faeder et al., 2009], própria para modelagem de sistemas bioquímicos. Um modelo nessa linguagem deve ter cinco blocos básicos: *parameters*, *molecule types*, *seed species*, *reaction rules*, *observables* e *actions*. O código da Figura 3.13 apresenta as definições desses blocos para o sistema de reações da Figura 3.9.

Parâmetros do modelo, tais como os valores para constantes de taxas de transição, para as concentrações iniciais das moléculas e constantes físicas podem ser definidos no bloco *parameters*. No exemplo sendo considerado, os valores das taxas das reações (r_1 , r_2 e r_3) e as quantidades iniciais das moléculas A e B livres (nA e nB) são definidos neste bloco.

O bloco *molecule types* é usado para definir a composição e estados de uma molécula. Moléculas podem ter componentes que representam seus elementos funcionais e que lhes permitem se ligar a outras moléculas ou a outro componente dela mesma. Moléculas também podem ter associadas a elas variáveis de estado, que tem um conjunto finito de valores possíveis e que representam conformações ou estados químicos da molécula.

```

begin parameters
#rate constants
r1 1
r2 1
r3 0.1

#initial amount of free A and free B
nA 1
nB 1
end parameters

begin molecule types
A(n,state~F~C~D)
B(n,state~F~C)
end molecule types

begin seed species
A(n,state~F) nA
B(n,state~F) nB
end seed species

begin reaction rules
A(n,state~F) + B(n,state~F) <-> A(n!1,state~C).B(n!1,state~C) r1,r2
A(n,state~F) -> A(n,state~D) r3
end reaction rules

begin observables
Molecules oA A(n,state~D)
Molecules oB B(n,state~F)
Molecules oAB A(n!1,state~C).B(n!1,state~C)
end observables

generate_network({overwrite=>1});
simulate_ssa({suffix=>ssa,t_end=>10,n_steps=>100});

```

Figura 3.13. Modelo no BIONETGEN para o sistema de reações da Figura 3.9.

No código da Figura 3.13, a molécula A tem dois componentes: o componente \mathbf{n} , que é utilizado para a ligação com outras moléculas, e o componente \mathbf{state} , que é uma variável de estado que pode assumir os valores \mathbf{F} (a molécula A está livre), \mathbf{C} (a molécula A formou um complexo com a molécula B) e \mathbf{D} (a molécula A se degradou). A molécula B , por sua vez, possui os mesmos componentes da molécula A , entretanto, sua variável de estado \mathbf{state} não pode assumir o valor \mathbf{D} , pois a molécula B não pode se degradar. A Figura 3.14 apresenta uma ilustração dos dois tipos de moléculas definidos no modelo.

No bloco *seed species* os valores iniciais das moléculas ou complexos são especificados. Se o valor da espécie não é inicializado neste bloco, ela é considerada como ausente. No exemplo, as moléculas A e B livres ($state \sim F$) são inicializadas com as quantidades nA e nB no bloco *seed species*.

O bloco *reaction rules* define como as moléculas do sistema reagem através de regras. Cada regra pode ser vista como uma reação química, que possui o conjunto de reagentes, uma seta que indica a direção da reação, o conjunto de produtos e a

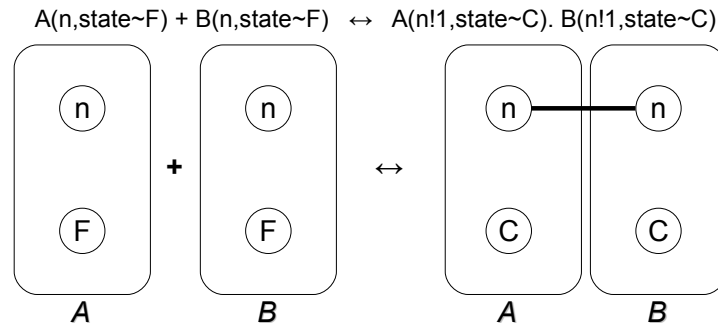
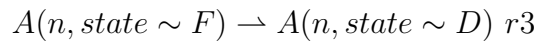


Figura 3.15. Reação de complexação das moléculas A e B .



é irreversível e mostra uma reação em que a molécula A muda o estado do seu componente **state**, que representa a reação em que a molécula A se degrada. A Figura 3.16 ilustra a degradação da molécula A , através da mudança de seu componente **state**.

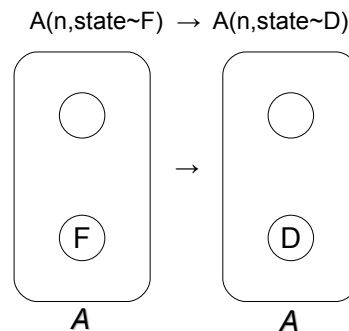


Figura 3.16. Reação de degradação da molécula A .

No bloco *observables* são definidas as saídas do modelo, ou seja, os substratos do modelo que terão suas quantidades calculadas no tempo (serão observados) e que poderão ser utilizados pelo BIOLAB para construir as propriedades de interesse, como é mostrado na próxima seção. Primeiramente, especifica-se o tipo do substrato sendo observado (*Molecules*). Posteriormente, especifica-se o nome do observável que será utilizado na especificação das propriedades a serem verificadas. Finalmente, especifica-se o substrato a ser observado. No nosso exemplo, definimos três observáveis \mathbf{oA} , \mathbf{oB} e \mathbf{oAB} que se referem, respectivamente, a quantidade de moléculas A no estado degradado, a quantidade de moléculas B livres e a quantidade de complexos formados pelas moléculas A e B .

No bloco *actions* são definidas as ações a serem realizadas com o modelo especifi-

cado. Existem dois tipos de ações: a geração da rede de reações químicas (comando `generate_network(overwrite=>1);`) e, após isso, a simulação do modelo, que implica na geração de execuções dos mesmos, ou seja, as concentrações das espécies ao longo do tempo. A simulação pode se dar ou por resolver um conjunto de ODEs usando o comando `simulate_ode`, produzindo uma concentração média das espécies, ou por utilizar o Algoritmo de Simulação Estocástica (SSA) de Gillespie [1977], através do comando `simulate_ssa`. Enquanto o primeiro algoritmo é determinístico, ou seja, produz sempre um mesmo caminho para um mesmo conjunto inicial de parâmetros, o segundo é um processo estocástico e pode gerar caminhos diferentes. Além disso é preciso definir o tempo máximo de simulação (`t_end`) e o número de vezes (`n_steps`) neste intervalo de tempo em que os valores das concentrações das espécies definidas no bloco `observables` serão coletadas ou observadas. No exemplo sendo considerado, serão geradas execuções estocásticas do modelo de 10 unidades de tempo e as concentrações para os observáveis serão coletadas 100 vezes ao longo deste intervalo de tempo. Como é explicado mais detalhadamente na Seção 3.3.2.3, o BIOLAB necessita de execuções estocásticas do modelo, logo o comando de simulação a ser usado é sempre `simulate_ssa`.

3.3.2.2 Especificação de Propriedades

As propriedades no BIOLAB devem ser especificadas na lógica PBLTL (do inglês, *Probabilistic Bounded Linear Temporal Logic*) que é uma extensão da lógica temporal BLTL (do inglês, *Bounded Linear Temporal Logic*) [Finkbeiner e Sipma, 2001]. A gramática da lógica PBLTL é a seguinte:

$$\begin{aligned}\Phi & ::= \mathcal{P}_{\geq\theta} \phi \\ \phi & ::= x \trianglelefteq v \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid \neg\phi \mid \mathbf{X}(\phi) \mid (\phi_1 \mathbf{U}^t \phi_2)\end{aligned}$$

onde $\trianglelefteq \in \{>, <, \leq, \geq, =\}$, $\theta \in [0, 1]$, x pertence ao conjunto de observáveis definidos no bloco `observables` no BIONETGEN (ou uma expressão aritmética envolvendo os observáveis) e $v \in \mathbb{R}_{\geq 0}$. Os operadores \neg , \vee e \wedge são operadores lógicos proposicionais, enquanto \mathbf{X} e \mathbf{U}^t são operadores temporais.

A fórmula $\mathcal{P}_{\geq\theta} \phi$ é verdade no estado s se a probabilidade que ϕ seja satisfeito por um caminho iniciando no estado s seja maior ou igual que θ . A fórmula de caminho $\mathbf{X}(\phi)$ é verdade somente se ϕ é satisfeito no próximo estado, enquanto $\phi_1 \mathbf{U}^t \phi_2$ é verdade somente se ϕ_2 é satisfeito em algum instante tempo menor que t e em todos os instantes de tempo anteriores ϕ_1 é satisfeito.

Também é possível definir os operadores $\mathbf{F}^t \phi$ (*eventually*), que é verdade se ϕ é satisfeito nas primeiras t unidades de tempo, e o operador $\mathbf{G}^t \phi$ (*always*), que é verdade

se ϕ é satisfeito em todo estado dentro de t unidades de tempo.

Seguem alguns exemplos de propriedades considerando o modelo BIONETGEN, definido na seção anterior, para o sistema de reações apresentado na Figura 3.9:

- $\mathcal{P}_{\geq\theta} \mathbf{F} \ oAB = 1$: a probabilidade que a molécula A se ligue à B nas primeiras t unidades de tempo é maior ou igual que θ . Aqui, o tempo t é definido no bloco *actions* do modelo usando o parâmetro t_end . O observável oAB , definido na seção anterior, se refere ao complexo formado pela ligação das moléculas A e B ;
- $\mathcal{P}_{\geq\theta} \ oAB = 0 \ \mathbf{U} \ oA = 1$: a probabilidade que a molécula A se degrade antes de se ligar a molécula B nas primeiras t unidades de tempo é maior ou igual que θ . Novamente, o tempo t é definido no bloco *actions* do modelo e o observável oA se refere a molécula A no estado degradado (ver subseção anterior).

3.3.2.3 Implementação

O BIOLAB é uma técnica de verificação de modelos estatística que utiliza o BIONETGEN para gerar caminhos³ estocásticos e finitos do modelo biológico e verifica se cada uma delas satisfaz a propriedade ϕ de $\mathcal{P}_{\geq\theta} \phi$. Então, a cada nova simulação do modelo e a verificação da propriedade ϕ para ela, o BIOLAB utiliza o algoritmo *Sequential Probability Ratio Test* (SPRT) [Wald, 1945] para gerar novas simulações do modelo BIONETGEN até que se saiba se a propriedade $\mathcal{P}_{\geq\theta} \phi$ em questão é satisfeita ou não para o modelo.

O algoritmo SPRT reduz o problema de verificar a propriedade $\mathcal{P}_{\geq\theta} \phi$ a um problema de teste de hipóteses: ele tem que decidir entre duas hipóteses $H_0 : \mathcal{P}_{\geq\theta}$ contra $H_1 : \mathcal{P}_{<\theta}$. A força desse teste é determinada por dois parâmetros α e β , tais que a probabilidade de aceitar H_1 quando H_0 é verdade, chamado **Erro Tipo I** é menor ou igual à α e a probabilidade de aceitar H_0 quando H_1 é verdade, chamado **Erro Tipo II** é menor ou igual à β .

Para tentar obter uma baixa probabilidade para ambos os erros, o algoritmo SPRT define uma região de indiferença $[\theta_1, \theta_0]$, tais que $\theta_1 = \theta - \Delta$ e $\theta_0 = \theta + \Delta$, onde $0 < \Delta < 1$ e testa a nova hipótese $H'_0 : \mathcal{P}_{\geq\theta_0}$ contra $H'_1 : \mathcal{P}_{\leq\theta_1}$. O algoritmo aceita H'_0 se $f_m \leq B$ e H'_1 se $f_m \geq A$, onde m é o número atual de execuções do modelo, $A = \frac{1-\beta}{\alpha}$ e $B = \frac{\beta}{1-\alpha}$. A razão f_m é dada a seguir:

$$f_m = \prod_{i=1}^m \frac{P(B_i = b_i) | p = \theta_1}{P(B_i = b_i) | p = \theta_0} = \frac{\theta_1^{d_m} (1 - \theta_1)^{m - d_m}}{\theta_0^{d_m} (1 - \theta_0)^{m - d_m}}$$

³Os termos simulação e caminho são usados nesta seção de maneira intercambiável e se referem a geração de um caminho finito na estrutura CTMC do modelo.

onde B_i é uma variável aleatória com distribuição Bernoulli, que pode assumir dois valores 0 ou 1. O resultado de B_i , denotado por b_i , tem o valor 1 se uma simulação satisfaz ϕ e 0 em caso contrário. Além disso $d_m = \sum_{i=1}^m b_i$. Dessa forma, o SPRT calcula f_m para sucessivos valores de m até que ou H'_0 ou H'_1 seja satisfeito.

A Figura 3.17 mostra uma sumarização dos passos do algoritmo de verificação de modelos estatístico BIOLAB.

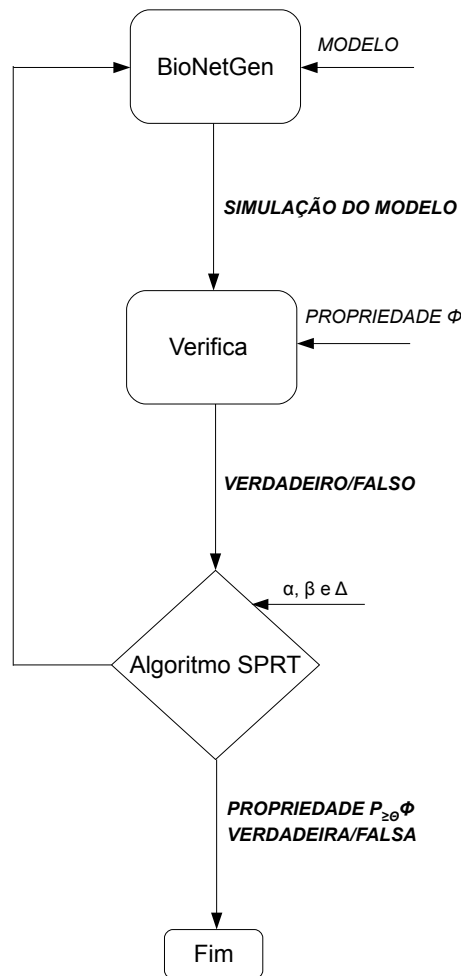


Figura 3.17. Diagrama que mostra arquitetura simplificada do BIOLAB

3.3.3 Comparação entre PRISM e BIOLAB

Embora a lógica PBLTL usada pelo BIOLAB se apresente, a primeira vista, bem similar à lógica CSL usada pelo PRISM, existem diferenças. Uma das mais evidentes é que o BIOLAB não incorpora operadores de *rewards* e também não permite variar o símbolo \geq das propriedades $\mathcal{P}_{\geq\theta}$, como, diferentemente, é permitido pela ferramenta

PRISM. Neste caso, para saber os valores aproximados das probabilidades, é geralmente necessário efetuar uma pesquisa binária variando o valor de θ . Além disso, o tempo t não pode ser omitido, como ocorre com o intervalo I da lógica CSL, porque o BIOLAB trabalha com caminhos finitos do sistema.

Esta subseção apresenta alguns resultados ilustrativos ao utilizar ambas as ferramentas na análise do sistema biológico exemplo utilizado ao longo deste capítulo. Todos os experimentos que serão mostrados nesta seção e nos demais capítulos ao longo deste trabalho foram executados em uma máquina Intel(R) Xeon(R) CPU X3323, 2.50GHz, com 17 GB de memória RAM. Os tempos considerados se referem aos tempos de CPU do usuário (do inglês, *user CPU time*) que retornam o tempo de processador gasto para executar instruções do programa do usuário. O PRISM fornece esses valores tanto para construir um modelo, quanto para verificá-lo. A construção de um modelo corresponde ao mapeamento da sua descrição na linguagem PRISM na sua representação MTBDD e, adicionalmente, à computação dos estados alcançáveis. Todos os estados inalcançáveis, que não são de interesse, são então removidos. A verificação da propriedade na ferramenta PRISM, corresponde à aplicação da solução apropriada (resolver sistema de equações lineares ou aplicar o métodos da uniformização) para verificar a propriedade de interesse. Para o BIOLAB, somente há o tempo de verificação e ele é obtido através do comando **time** do sistema operacional Linux.

A seguir, serão apresentados os resultados para verificar uma propriedade referente à *probabilidade da molécula A se ligar a molécula B* no primeiro segundo de funcionamento do sistema biológico, a partir do estado inicial. O interesse é saber se essa probabilidade é maior que 50%. A propriedade pode ser expressa em CSL (especificação usada pelo PRISM) e PBLTL (especificação usada pelo BIOLAB), respectivamente como:

1. $'init' \Rightarrow \mathcal{P}_{\geq 0,5} [\mathbf{F}^{\leq 1} ab = 1]$

2. $\mathcal{P}_{\geq 0,5} \mathbf{F} \{oAB = 1\}$

O rótulo *'init'* é um identificador do PRISM que se refere ao conjunto de estados iniciais do modelo CTMC. Adicionalmente, para o caso do BIOLAB, é preciso definir o valor do parâmetro t_{end} , tempo de simulação, como 1.

Para a ferramenta PRISM a resposta da propriedade (1) é **verdadeira**, indicando que a probabilidade das moléculas A e B se ligarem no primeiro segundo é maior ou igual a 50%. O tempo total de verificação, incluindo o tempo de construção, foi de 0,0010 segundos.

A tabela a seguir mostra os resultados para 5 execuções do BIOLAB. Em todos os casos o resultado do algoritmo foi **verdadeiro**, indicando que a propriedade

$\mathcal{P}_{\geq\theta_0} \mathbf{F} \{oAB = 1\}$, sendo $\theta_0 = 0,5 + \Delta$, é verdade. Ou seja, a probabilidade das moléculas A e B reagirem no primeiro segundo é maior ou igual a 50,5%.

Tabela 3.1. Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$.

<i>Execução</i>	<i>Número de Caminhos Gerados</i>	<i>Tempo (s)</i>
1	1612	219,126
2	794	108,111
3	1006	137,361
4	1690	229,922
5	1398	190,540

O valor médio do tempo de verificação para o BIOLAB foi de $177,0120 \pm 43,1352$ segundos e o número de caminhos médios gerados foi $1300,0000 \pm 317,8868^4$. Essas diferenças em cada execução do algoritmo se devem ao fato de ele não ser determinístico e se basear em simulações estocásticas.

A tabela seguinte mostra os resultados para 5 execuções do BIOLAB quando os erros α e β são reduzidos de 0,01 para 0,001. Novamente o resultado para cada execução foi **verdadeiro**.

Tabela 3.2. Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,001 e β (Erro Tipo 2) = 0,001. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$.

<i>Execução</i>	<i>Número de Caminhos Gerados</i>	<i>Tempo (s)</i>
1	1410	192,296
2	1488	201,901
3	2382	323,636
4	2474	336,941
5	1582	215,053

O valor médio do tempo de verificação para o BIOLAB com os novos valores de erros foi de $253,9654 \pm 57,6258$ segundos e o número de caminhos médios gerados foi $1867,2000 \pm 423,4565$. Note, portanto, o aumento em relação aos valores médios obtidos anteriormente através da Tabela 3.1. Como esperado, se erros menores são desejáveis, maior será a quantidade de caminhos gerados pelo algoritmo para verificar a propriedade de interesse, refletindo, conseqüentemente, em um maior tempo de execução.

⁴As médias das variáveis são acompanhadas do intervalo de confiança (IC) de 90% e apresentadas no formato *média* \pm *IC*. Isso significa que com 90% de certeza, a média da variável está no intervalo [*média* - *IC*, *média* + *IC*].

A Tabela 3.3, por sua vez, mostra o tempo de verificação e número de caminhos para 5 execuções do BIOLAB quando o valor de Δ da região de indiferença é reduzido de 0,005 para 0,001. O resultado do algoritmo foi **verdadeiro** para todas as execuções.

Tabela 3.3. Resultado da análise do BIOLAB para o modelo apresentado na Figura 3.13. α (Erro Tipo 1) = 0,001 e β (Erro Tipo 2) = 0,001. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,001$.

<i>Execução</i>	<i>Número de Caminhos Gerados</i>	<i>Tempo (s)</i>
1	8967	1220,7760
2	8629	1170,3970
3	9671	1312,7140
4	8491	1152,7520
5	8683	1184,1180

Os novos valores médios para o tempo de verificação e para o número de caminhos gerados pelas 5 execuções do BIOLAB são, respectivamente, $1208,1514 \pm 52,1114$ segundos e $8888,2000 \pm 385,8116$. Novamente, observe o aumento em relação aos valores médios obtidos anteriormente através da Tabela 3.2. Assim, diminuir o tamanho da região de indiferença através da redução do parâmetro Δ aproxima os valores de θ_0 e θ_1 ao valor $\theta = 0,5$ desejado. Em contrapartida, tem-se o preço de aumentar o número de caminhos necessários para o algoritmo SPRT verificar a propriedade de interessante.

Finalmente é importante ressaltar que para o simples sistema de reações mostrado na Figura 3.13 o PRISM se mostrou mais adequado, verificando a propriedade de interesse em um tempo menor. Entretanto, como será mostrado ao longo dos demais capítulos deste trabalho, existem outros contextos em que o BIOLAB se mostra como uma solução mais eficiente, principalmente na verificação de sistemas mais complexos.

Capítulo 4

Sistemas de Transporte de Íons em Membranas Celulares

A existência de sistemas de transporte capazes de mover íons através das membranas celulares é essencial para a vida das células [Sepúlveda e Bezanilla, 2005]. Eles estão envolvidos em diversos processos biológicos, sendo cruciais em cada percepção, movimento, pensamento e batimento do coração no corpo humano. As seções seguintes introduzem alguns conceitos básicos da biologia importantes para o entendimento desses sistemas, bem como o processo de sinalização celular denominado potencial de ação, que depende totalmente desses sistemas. Além disso, as bombas e os canais iônicos, que são dois representantes desses sistemas de transporte de íons a serem estudados neste trabalho, são também apresentados.

4.1 A Célula e a Membrana Celular

O conteúdo desta seção é baseado nos trabalhos de Émerson Flamarion da Cruz [2006], Campos [2008] e Aidley e Stanfield [1996].

A célula é a menor unidade estrutural básica dos seres vivos e é constituída de três elementos básicos: o núcleo, o citoplasma e a membrana celular. O núcleo é o centro de controle da célula, pois contém toda informação sobre seu funcionamento e de todos os organismos a qual pertence. O citoplasma é um meio aquoso onde se encontram pequenas unidades funcionais denominadas organelas, responsáveis por diferentes tarefas na dinâmica celular. A membrana celular é a estrutura que delimita e separa o citoplasma do meio extracelular, controlando o fluxo das substâncias que entram e saem do interior da célula. Ela é formada por duas camadas de moléculas de fosfolípido que possuem uma extremidade globular e hidrofílica e outra extremidade alongada e hidrofóbica, como mostra a Figura 4.1. Existem também proteínas periféricas parcial-

mente mergulhadas na camada de lipídio e outras que atravessam toda a membrana, estabelecendo uma ponte entre o meio intracelular e o extracelular.

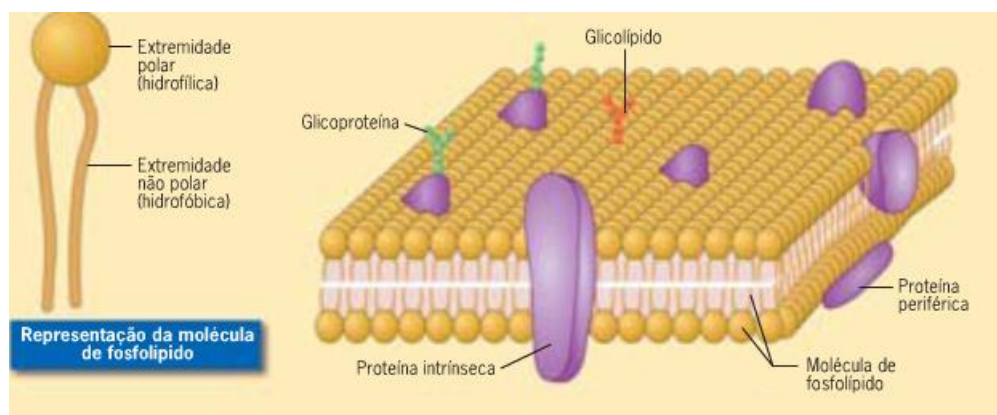


Figura 4.1. Visão esquemática da membrana celular [Cindy e Geisiane, 2010].

Ambos os meios intracelular e extracelular são soluções aquosas compostas por sais dissolvidos. A membrana celular age, dessa forma, como uma barreira separando os conteúdos presentes dentro e fora da célula, permitindo que as concentrações nesses compartimentos sejam diferentes.

As moléculas podem ser transportadas através da membrana por processos ativos, que requerem gasto de energia, ou por meios passivos. A água e outras substâncias como o oxigênio e o dióxido de carbono cruzam a membrana por processos passivos denominados osmose e difusão facilitada, respectivamente. Íons de sódio e potássio, por outro lado, deslocam-se para o meio de menor concentração através de proteínas intrínsecas com arranjos especiais (canais iônicos) mergulhadas na bi-camada fosfolipídica.

As soluções extracelulares geralmente contêm uma concentração relativamente alta de íons de sódio e cloreto, e uma concentração baixa de íons de potássio, cálcio e outros íons. Já o meio intracelular contém uma concentração relativamente alta de potássio e cálcio, e uma baixa concentração de íons de sódio. Essas diferenças são criadas e mantidas por mecanismos ativos (bombas iônicas) que usam energia para bombear íons contra o gradiente de concentração, ou seja, de regiões de baixa concentração para regiões de alta concentração. As bombas também são estruturas proteicas mergulhadas na bi-camada fosfolipídica da célula. A diferença da composição química e elétrica nos fluidos intra e extracelular gera, além de uma diferença de potencial transmembrânica, um gradiente eletroquímico através da membrana para cada espécie iônica. As células fazem uso desses gradientes eletroquímicos em seus processos de controle e sinalização.

No seu estado de repouso a membrana apresenta um potencial negativo (cerca de -80 mV). Isso indica que o exterior da célula possui mais cargas positivas que o

seu interior. Quando a membrana de uma célula excitável é despolarizada além de um limiar, a célula dispara um potencial de ação, que é uma alteração rápida na polaridade da voltagem, de negativa para positiva e de volta para negativa. A duração do potencial de ação varia de célula para célula: nas células nervosas essa duração é de aproximadamente 1 ms, enquanto que nas células musculares cardíacas ela é maior que 200 ms. O potencial de ação pode ser dividido em três fases principais. A fase de repouso, na qual a membrana está polarizada com o potencial de equilíbrio. A fase de despolarização, na qual o interior da célula torna-se mais positiva, o que é causado pela abertura de canais de sódio dependentes de voltagem, provocando o influxo desses íons para dentro da célula (a concentração de sódio é maior fora da célula). E, por fim, a fase de repolarização que é caracterizada pelo fechamento dos canais de sódio e abertura dos canais de potássio, expulsando esses últimos de dentro da célula (a concentração de potássio é maior dentro da célula) e levando a célula novamente ao seu potencial de repouso. Potenciais de ação podem ser gerados por muitos tipos de células, mas são utilizados mais intensamente pelo sistema nervoso, para comunicação entre neurônios e para transmitir informação dos neurônios para outro tecido do organismo, como os músculos ou as glândulas.

4.2 Canais Iônicos

Canais iônicos são complexos proteicos contendo poros aquosos que podem abrir e fechar para permitir o fluxo passivo de íons através da membrana das células. O fluxo de íons é passivo porque ocorre na mesma direção do gradiente de concentração, ou seja, de regiões de alta concentração para regiões de baixa concentração. Um canal iônico é composto de uma ou mais moléculas proteicas que se configuram em um arranjo oligomérico com o seguinte padrão: uma roseta de quatro, cinco ou seis subunidades proteicas similares com um poro aquoso central que pode abrir para permitir o fluxo de íons de um lado para o outro da membrana [Aidley e Stanfield, 1996].

É um elemento fundamental no processo de excitabilidade em células excitáveis (como os neurônios e células musculares) e seu mau funcionamento, devido à mutações genéticas, pode causar várias patologias, como é mostrado na Tabela 4.1. Eles também são frequentemente os principais alvos das toxinas de animais e das pesquisas por novos fármacos, sendo, por esses motivos interessantes na pesquisa biomédica.

Os canais iônicos podem ainda ser classificados de acordo com a natureza de sua abertura (ativados por alterações na diferença de potencial ou por ligantes), os tipos de íons aos quais são permeáveis e o número de poros que possuem.

A Figura 4.2 é uma representação esquemática de um canal iônico. Como pode

Tabela 4.1. Patologias associadas ao mau funcionamento de canais iônicos. Obtido de [Émerson Flamarion da Cruz, 2006]

<i>Canal Iônico Defeituoso</i>	<i>Patologia</i>
Sódio	Paralisia periódica hipercalêmica (Doença de Gamstrop) Paramiotonia congênita (Doença de Eulenburg) Miotonia atípica Síndrome do QT longo (gene LQT2)
Cloreto	Fibrose cística Miotonia congênita (Doença de Thomsen) Miotonia generalizada (Doença de Becker)
Potássio	Síndrome do QT longo (genes LQT1 e LQT3)
Cálcio	Paralisia periódica hipocalêmica Hipertermia maligna

ser visto na figura, quando aberto o canal permite um rápido fluxo de íons através da membrana (cerca de 10^7 íons por segundo) e, por outro lado, quando fechado a passagem dos íons não é permitida.

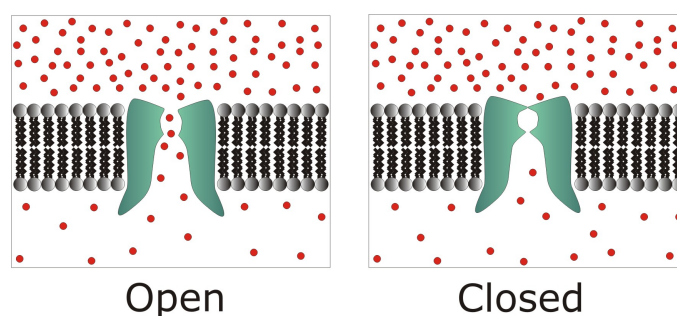


Figura 4.2. Esquema de canal de íons [sophion, 2010].

Um dos aspectos dos canais iônicos amplamente estudado é a descrição de sua cinética, ou seja, como ocorre a abertura dos canais no tempo [Sansom e et al, 1989]. Nesse contexto, canais usualmente existem em dois estados, aberto ou fechado, e podem mudar de um estado para outro aleatoriamente. As mudanças são, portanto, eventos estocásticos e podem ser descritas no tempo somente em termos probabilísticos. Quando um canal está aberto há uma probabilidade constante de ele mudar para o estado fechado no intervalo de tempo δt , independente do tempo que ele tenha ficado no estado aberto e quando ele tenha chegado neste estado. Processos com essas características, em que a probabilidade de uma mudança particular em pequenos intervalos de tempo é constante, são exemplos de processos de Markov. Portanto, a cinética de

canais iônicos estocásticos é usualmente assumida como um processo de Markov de estados discretos e de tempo contínuo [Colquhoun e Hawkes, 1977; Neher e Stevens, 1977] e podem ser descritos utilizando PMC naturalmente através de modelos CTMCs.

Um dos paradigmas tradicionais utilizados para interpretar resultados experimentais de canais iônicos é o diagrama cinético. A Figura 4.3 mostra um diagrama cinético utilizado para sumarizar as trocas de um simples canal iônico de cálcio. Para esse canal existem dois estados, um aberto e outro fechado, sendo k_1 e k_2 constantes de taxa de transição para mudanças de abertura e fechamento, expressas em unidades de frequência (s^{-1}). Encontrar o diagrama cinético que melhor represente o comportamento de um canal de íons é um dos grandes ramos de pesquisa da área de eletrofisiologia como pode ser visto nos trabalhos de Moss et al. [1999] e Magleby e Weiss [1990].

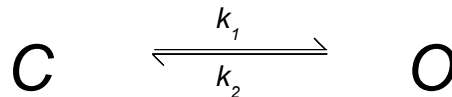


Figura 4.3. Diagrama cinético para um simples canal com dois estados.

4.3 Bombas Iônicas

As bombas iônicas, também conhecidas como sistemas de transporte ativo primário, são proteínas que movem íons através das membranas celulares contra o gradiente de concentração. Para promover esse fluxo ativo de íons, elas utilizam fontes de energia primária como luz, energia redox ou energia derivada da hidrólise do Trifosfato de Adenosina (ATP). Comparadas com os canais iônicos, as bombas são sistemas de transporte de íons extremamente lentos, movendo íons através da membrana em uma taxa entre 10-1000 íons por segundo [Lauger, 1991]. Assim, enquanto um pequeno número de canais iônicos rapidamente altera a diferença de potencial na membrana (o número de íons de sódio e potássio necessários para polarizar ou despolarizar a membrana é relativamente pequeno), um grande número de bombas lentamente e continuamente movem íons para manter as concentrações apropriadas: alta concentração de potássio e baixa concentração de sódio dentro da célula.

A família de bombas *ATPases do tipo P* são transportadores ativos que catalisam a hidrólise do ATP para originar ADP e fosfato inorgânico, com liberação de energia. A bomba de sódio e potássio modelada no Capítulo 6 é um dos representantes dessa

família. Uma ATPase do tipo P é usualmente uma proteína integral que atravessa a membrana plasmática e que apresenta 10 regiões. Algumas vezes ela pode ter uma segunda subunidade [Lehninger et al., 2004].

Uma bomba pode assumir um número diferente de estados E_0, E_1, \dots, E_n , que se diferem pela presença de ligantes, como íons ou fosfato, e pela configuração do sítio de ligação, se está voltado para dentro ou fora da célula [Lauger, 1991]. A Figura 4.4 mostra esse mecanismo da bomba de mudar sua configuração para alternar o sítio de ligação entre os lados da membrana. Essa mudança de configuração é usualmente associada com uma grande mudança de afinidade iônica, devido à mudanças da estrutura proteica. É importante ainda destacar que além das configurações onde o sítio de ligação está voltado para dentro ou fora da membrana, existe um estado intermediário (*occluded state*) em que os íons estão presos dentro da proteína e não podem ser liberados para nenhum dos dois lados da membrana.

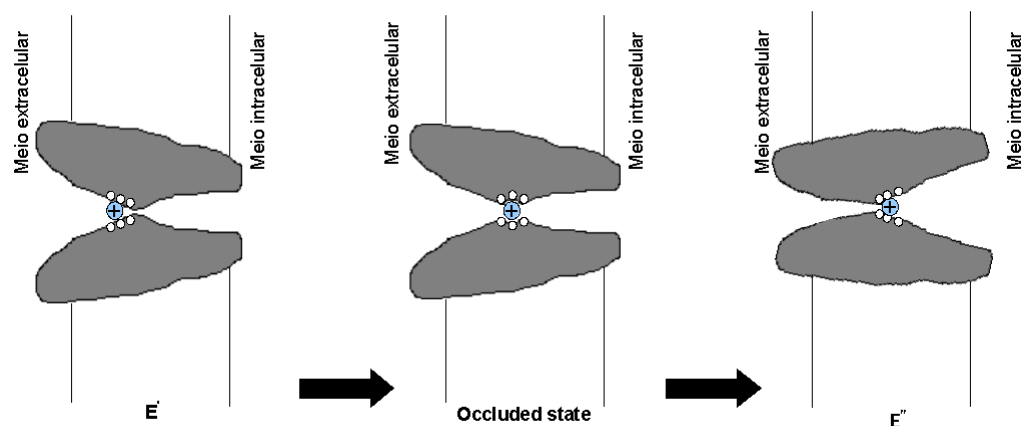


Figura 4.4. Mecanismo de mudança de configuração da bomba. E' e E'' são estados com sítios de ligação voltados para fora e dentro da célula, respectivamente. No *occluded state*, o íon não pode mover nem para dentro, nem para fora da célula. Pequenos círculos brancos representam grupos ligantes da proteína.

As bombas iônicas também podem ser alvos de toxinas que provocam modificações no comportamento das mesmas. Um exemplo é a toxina palitoxina que permite que o sítio de ligação da bomba de sódio e potássio esteja acessível tanto dentro quanto fora da célula, o que leva a destruição do gradiente iônico essencial para a mesma [Rodrigues et al., 2008].

A complexa dinâmica das bombas é usualmente descrita por uma via bioquímica composta por um conjunto de reações elementares que lhes permitam passar pelos estados $E_0 \rightarrow E_1 \rightarrow \dots \rightarrow E_n$ para que a troca de íons entre os dois lados da membrana celular ocorra. Esse sistema de reações é cíclico, ligando e liberando substratos em uma

sequência ordenada de reações, as quais são associadas constantes de taxas cinéticas [Ciobanu et al., 2002; Chapman et al., 1983; Rodrigues et al., 2008; Lauger, 1991]. Além disso, segundo Kwiatkowska et al. [2008], para modelar vias biológicas o modelo tipicamente apropriado é CTMC e, portanto, o emprego de PMC na modelagem e análise desses sistemas é natural.

Existem ainda condições sob as quais uma bomba funciona no sentido reverso. Por exemplo, para a bomba de sódio e potássio, cuja energia é obtida pela quebra (hidrólise) de ATP, produzindo difosfato de adenosina (ADP) e fosfato (P_i), o funcionamento no sentido reverso conduz a síntese de ATP pela ligação de ADP e P_i . Dessa forma, as condições favoráveis para a reversão dessa bomba seriam baixa concentração de ATP, alta concentração de ADP e P_i e um forte gradiente iônico para íons de sódio e potássio direcionado, respectivamente, para dentro e fora da célula [Lauger, 1991].

Capítulo 5

Metodologias para Modelagem de Sistemas Biológicos

Este capítulo sumariza as principais metodologias para modelar sistemas biológicos através da técnica de verificação de modelos, sendo várias delas empregadas na modelagem dos sistemas de transporte de íons que será apresentada nos próximos capítulos. Elas são aplicadas principalmente quando há necessidade de se ter mais de um representante de uma mesma espécie (um tipo de molécula ou íon, por exemplo). Esse aumento no número de representantes é normalmente associado ao aumento do número de estados do modelo. Dessa forma, algumas das técnicas de modelagem, como as abordagens populacional e a baseada em níveis, visam, sobretudo, contornar essa limitação quando o modelo envolve um grande número de representantes de moléculas ou íons. Destaca-se que essas técnicas são aplicadas aos modelos PRISM, que, por ser um PMC exato, sofre mais com o problema da explosão do espaço de estados que o BIOLAB.

Novamente, para ilustrar a aplicação das técnicas de modelagem, o conjunto de reações químicas mostrado na Figura 5.1 será considerado ao longo deste capítulo.

5.1 Abordagem Individual

A abordagem individual modela o estado de cada novo representante da espécie individualmente através de um módulo adicional na ferramenta PRISM. A Figura 5.3 ilustra o procedimento para estender o modelo da Figura 5.2, elaborado no Capítulo 3, de maneira que haja agora dois representantes das moléculas A e B no sistema de reações mostrado na Figura 5.1.

No modelo estendido, um módulo para cada molécula extra é adicionado, usando o recurso de renomeamento da linguagem PRISM que cria uma cópia idêntica do módulo

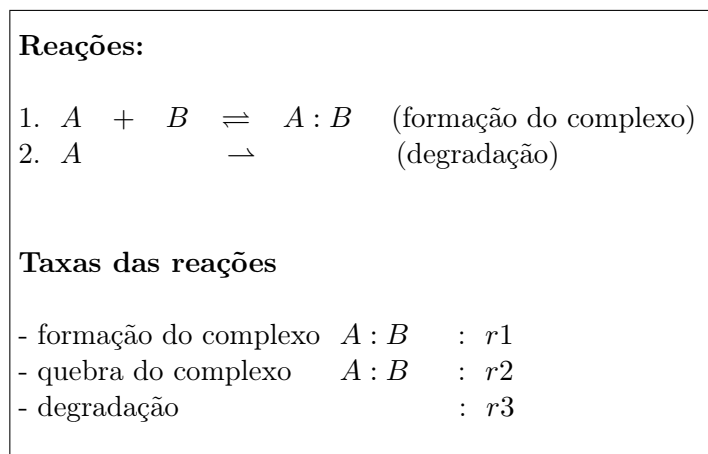


Figura 5.1. Sistema de reações químicas empregado como exemplo ao longo do capítulo.

existente, exceto pelos nomes das variáveis, as quais são dados novos nomes. Posteriormente, a construção **system ... endsystem** é empregada para descrever como os módulos do sistema são compostos para produzir o modelo final. Como as moléculas do tipo A não interagem entre si, elas são compostas usando o operador de paralelismo assíncrono $|||$, que não as permite sincronizar em nenhuma ação. O mesmo é verdade para as moléculas do tipo B . O operador de paralelismo $||$ é então usado para compor esses subsistemas, fazendo-os sincronizar em todas as ações de mesmo rótulo. Se o operador $|||$ não fosse utilizado, seria preciso introduzir diferentes rótulos (e ações) para representar cada uma das possíveis interações entre as moléculas A e B (por exemplo, ações da forma $bind_i_j$ representariam ligações da i -ésima molécula A com a j -ésima molécula B) e impedir que as moléculas de mesmo tipo interajam entre elas mesmas.

A abordagem individual é algumas vezes necessária quando a propriedade de interesse se refere ao comportamento de um representante particular da espécie. Entretanto ela é pouco escalável e sofre do problema da explosão do espaço de estados [Kwiatkowska et al., 2010]. A metodologia alternativa é o emprego da abordagem baseada em população, onde apenas o número de representantes de cada tipo de espécie é modelado através de contadores.

5.2 Abordagem Populacional

A abordagem populacional permite um aumento mais gradual no espaço de estados quando se aumenta o número de moléculas, possibilitando a análise de modelos maiores. De fato, a abordagem baseada em população é considerada como uma técnica de redução exata do espaço de estados por simetria [Heath et al., 2008], que explora a

```

ctmc

const double r1=1;
const double r2=1;
const double r3=0.1;

module m_A
a: [0..1] init 1; // 0 - degradada ou ligada, 1 - livre

[bind] a=1 -> 1 : (a'=0);
[rel] a=0 -> 1 : (a'=1);
[deg] a=1 -> 1 : (a'=0);

endmodule

module m_B
b: [0..1] init 1; // 0 - ligada, 1 - livre

[bind] b=1 -> 1 : (b'=0);
[rel] b=0 -> 1 : (b'=1);

endmodule

module m_AB
ab: [0..1] init 0; // 0 - complexo não existe, 1 - complexo formado

[bind] ab=0 -> 1 : (ab'=1);
[rel] ab=1 -> 1 : (ab'=0);

endmodule

module start
[bind] true -> r1 : true;
[rel] true -> r2 : true;
[deg] true -> r3 : true;

endmodule

```

Figura 5.2. Modelo no PRISM para o sistema de reações da Figura 5.1 considerando uma molécula de cada espécie.

```

...

module m_A2 = m_A [ a=a2 ] endmodule
module m_B2 = m_B [ b=b2 ] endmodule

//definição do sistema (moléculas de mesmo tipo não interagem)
system
(m_A ||| m_A2) || (m_B ||| m_B2) || m_AB || start
endsystem

```

Figura 5.3. Extensão do modelo PRISM da Figura 5.2 de modo a considerar duas moléculas de cada espécie e empregando a abordagem individual.

presença de replicações no modelo. Por ser exata, o modelo reduzido irá produzir resultados idênticos aos produzidos para o modelo original (construído com a abordagem individual).

Considerando o sistema de reações exemplo, uma molécula do tipo A poderá se ligar a uma do tipo B somente se os contadores para o número de representantes dessas duas moléculas forem maiores que 0 e, ainda, se não houver um complexo AB já formado. Além disso, se essa reação ocorre, há um decremento nos contadores das moléculas A e B , e um aumento do contador que representa o complexo AB . A Figura 5.4 mostra um modelo PRISM baseado em população para o exemplo sendo considerado quando existem duas moléculas do tipo A e duas do tipo B . Note que é preciso ajustar as taxas das reações de ligação e degradação para levar em conta as diferentes interações codificadas. Isso é conseguido multiplicando-se as taxas de interação do modelo original apresentado na Figura 5.2, que contém apenas uma molécula de cada espécie, pelo número de representantes de cada espécie envolvida em uma reação no novo modelo populacional. Maiores detalhes de como modificar as taxas das reações para levar em conta a concentração dos reagentes envolvidos nas reações quando a espécie é modelada utilizando uma abordagem populacional são fornecidos na próxima seção.

A Tabela 5.1 mostra a variação no tamanho do modelo, no tempo para construí-lo (T_c) e verificar (T_v) a propriedade '*init*' $\Rightarrow \mathcal{P}_{\geq 0,5} [\mathbf{F}^{\leq 1} ab = 1]$ ao variar o número de moléculas (N) do tipo A e B usando tanto a modelagem populacional, quanto a modelagem individual. O rótulo '*init*' é um identificador do PRISM que se refere ao conjunto de estados iniciais do modelo CTMC. A resposta do algoritmo de verificação é **verdadeiro** para todas as entradas da tabela.

Tabela 5.1. Variação do *Tamanho* do modelo em termos do número de nós da representação MTBDD, dos tempos de construção do modelo (T_c) e dos tempos de verificação do modelo (T_v) devido às variações no número de moléculas do modelo (N).

N	Abordagem Populacional			Abordagem Individual		
	<i>Tamanho</i>	T_c (s)	T_v (s)	<i>Tamanho</i>	T_c (s)	T_v (s)
2	52	0,0110	0,0020	72	0,0110	0,0020
4	124	0,0120	0,0020	182	0,0150	0,0030
6	155	0,0120	0,0020	290	0,0150	0,0050
8	286	0,0130	0,0020	398	0,0190	0,0180
10	272	0,0130	0,0020	506	0,0220	0,1220

Note que, mesmo para o simples modelo biológico sendo considerado, a abordagem populacional escala melhor. Além disso, vale a pena enfatizar que a construção de um modelo individual quando o valor de N aumenta torna-se cada vez mais repetitiva e sujeita a erros, dada a necessidade de fazer vários renomeamentos de módulos, cada um representando uma nova molécula. Dessa forma, quando não há interesse individual em cada molécula, a abordagem populacional se mostra mais adequada.

```
ctmc

const int N=2; //número de cada tipo de molécula

const double r1=1;
const double r2=1;
const double r3=0.1;

module m_A
a: [0..N] init N;

[bind] a>0 -> a : (a'=a-1);
[rel] a<N -> 1 : (a'=a+1);
[deg] a>0 -> a : (a'=a-1);

endmodule

module m_B
b: [0..N] init N;

[bind] b>0 -> b : (b'=b-1);
[rel] b<N -> 1 : (b'=b+1);

endmodule

module m_AB
ab: [0..1] init 0;

[bind] ab=0 -> 1 : (ab'=1);
[rel] ab=1 -> 1 : (ab'=0);

endmodule

module start
[bind] true -> r1 : true;
[rel] true -> r2 : true;
[deg] true -> r3 : true;

endmodule
```

Figura 5.4. Novo modelo baseado na abordagem populacional para o sistema de reações da Figura 5.1 quando há duas moléculas de cada espécie.

5.3 Lei da Ação das Massas

Quando se emprega a abordagem populacional, conforme mencionado anteriormente, é preciso levar em conta as diferentes interações possíveis dada as quantidades dos reagentes que participam da reação, o que é implicitamente feito pela abordagem individual quando há a composição dos módulos de cada representante.

Nos modelos ao longo desse trabalho, caso não seja explicitado, será assumido que as reações seguem a cinética das ações das massas [Guldberg, 1984; Waage, 1984; Ullah et al., 2006], a qual afirma que a taxa final f_i de uma reação i é proporcional às concentrações de seus reagentes e é dada pela seguinte equação:

$$f_i = r_i \times \prod_{j=1}^{n_i} \langle X_j \rangle^{\kappa_{i,j}} \quad (5.1)$$

onde r_i é a taxa base ou constante de taxa da reação i e $\langle X_j \rangle$ é a concentração do reagente X_j , sendo que há n_i reagentes distintos na reação i . Por fim, $\kappa_{i,j}$ é a molecularidade do reagente X_j na reação i . Molecularidade de um reagente é o número de representantes do reagente envolvidos na reação. Por exemplo, na reação $A + B \rightarrow AB$, a molecularidade de A é 1. Por outro lado, na reação $2A + B \rightarrow A_2B$, a molecularidade de A é 2. No modelo da Figura 5.4, a taxa final da reação de ligação das moléculas $A + B \xrightarrow{r1} AB$, dada pelos comandos com rótulo **bind**, é $r1 * a * b$. Note que ambas moléculas têm molecularidade 1 nesta reação.

Vale a pena ressaltar também que tanto r_i como $\langle X_j \rangle$ podem ser dados em termos de molaridade (M ou $\frac{mols}{l}$) ou em termos de número de moléculas ou íons. No primeiro caso, a abordagem baseada em níveis é empregada, enquanto no segundo caso a abordagem baseada na química discreta é utilizada, sendo ambas abordagens descritas nas seções seguintes. No modelo da Figura 5.4 estamos falando em termos de número de moléculas, portanto $r1$ é a constante de taxa da reação em termos de moléculas e $\langle A \rangle$ e $\langle B \rangle$ são os números de moléculas de A ($\#A$) e de B ($\#B$), respectivamente.

Para os modelos descritos na ferramenta PRISM, a lei de ação das massas deve ser explicitada na modelagem, enquanto nos modelos descritos na ferramenta de descrição do BIOLAB, própria para o contexto biológico, as leis cinéticas são automaticamente incorporadas pela ferramenta.

5.4 Abordagem Populacional Baseada na Química Discreta

Assim como a simulação estocástica, a abordagem de modelagem e análise empregada neste trabalho que utiliza PMC é uma abordagem estocástica e discreta, em que a concentração dos representantes de cada espécie é modelada como uma quantidade discreta (número de íons, de moléculas) e não através de funções contínuas (caso da abordagem determinística).

Entretanto, as concentrações iniciais dos substratos dos sistemas e constantes de taxas de transição são comumente encontradas na literatura em termos da unidade molaridade (M ou $\frac{mols}{l}$). Dessa forma, é preciso converter esses valores para que considerem contagem de moléculas ou íons no processo de modelagem (tanto no PRISM, quanto no BIOLAB).

Para converter a quantidade inicial de um substrato X dada em molaridade ($[X]$)

para contagem de íons ou moléculas ($\#X$), a seguinte definição é empregada:

$$\#X = [X] \times V \times N_A \quad (5.2)$$

onde V é o volume onde os substratos se encontram e N_A é a constante de Avogadro ($6,02 \times 10^{23} \text{ mols}^{-1}$). Além disso, para converter as constantes de taxas dadas em uma química contínua para a química discreta é utilizada a conversão de Gillespie [Wolkenhauer et al., 2004], definida como segue:

$$r'_i = \frac{r_i}{(N_A V)^{\kappa-1}} \quad (5.3)$$

onde κ é a molecularidade da reação e r_i é a constante de taxa da reação i dada em termos de M ($M^{-1} \text{ sec}^{-1}, M^{-2} \text{ sec}^{-1}$). Molecularidade de uma reação é o número de entidades reagentes envolvidas na mesma. Por exemplo, na reação $A + B \rightarrow AB$, os reagentes são as moléculas A e B e o valor de κ é $1 + 1 = 2$. Por outro lado, na reação $2A + B \rightarrow A_2B$, embora os reagentes sejam os mesmos, o valor de κ é $2 + 1 = 3$.

Para fins de ilustração, considere o sistema de reações exemplo e suponha que as concentrações iniciais das moléculas A e B sejam ambas $0,5 \text{ mM}$. Considerando também que as moléculas estejam em um compartimento com volume de 10^{-18} litros, as quantidades iniciais de moléculas A e B são obtidas como segue:

$$\#A = \#B = (0,5 \times 10^{-3}) \times (10^{-18}) \times (6,02 \times 10^{23}) = 301 \quad (5.4)$$

Além disso, suponha $r_1 = 10^{11} M^{-1} s^{-1}$, $r_2 = 1 s^{-1}$ e $r_3 = 0,1 s^{-1}$. Os valores das taxas r_2 e r_3 na química discreta são os próprios valores de r_2 e r_3 fornecidos, dado que as reações de degradação da molécula A e de quebra do complexo AB têm molecularidade 1. Sendo assim, ao aplicar a Equação 5.3 o valor do denominador se iguala a 1. Já a conversão de r_1 , que representa a constante de taxa da reação de ligação entre as moléculas A e B , para a química discreta é dada por:

$$\frac{10^{11}}{((6,02 \times 10^{23}) \times 10^{-18})^{2-1}} \approx 170000 s^{-1} \quad (5.5)$$

A Figura 5.4 mostra o novo modelo PRISM para o sistema de reações da Figura 5.1 considerando esses novos parâmetros.

Já a Figura 5.6 apresenta o modelo BIOLAB considerando os novos parâmetros definidos anteriormente. Note, como mencionado anteriormente, que no modelo BIOLAB, diferentemente dos modelos PRISM, não há necessidade de explicitar que as taxas das reações são proporcionais às concentrações dos reagentes.

```

ctmc

const int N=301; //número de cada tipo de molécula

const double r1=170000;
const double r2=1;
const double r3=0.1;

module m_A

a: [0..N] init N;

[bind] a>0 -> a : (a'=a-1);
[rel] a<N -> 1 : (a'=a+1);
[deg] a>0 -> a : (a'=a-1);

endmodule

module m_B

b: [0..N] init N;

[bind] b>0 -> b : (b'=b-1);
[rel] b<N -> 1 : (b'=b+1);

endmodule

module m_AB

ab: [0..N] init 0;

[bind] ab<N -> 1 : (ab'=ab+1);
[rel] ab>0 -> 1 : (ab'=ab-1);

endmodule

module start

[bind] true -> r1 : true;
[rel] true -> r2 : true;
[deg] true -> r3 : true;

endmodule

```

Figura 5.5. Modificações no modelo populacional PRISM dado na química discreta para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5 \text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11} \text{ M}^{-1}\text{s}^{-1}$, $r2 = 1 \text{ s}^{-1}$, $r3 = 0,1 \text{ s}^{-1}$ e um volume de 10^{-18} litros.

5.5 Abordagem Populacional Baseada em Níveis

Modelos CTMC com níveis [Ciocchetta e Guerriero, 2009; Andrei e Calder, 2010] são modelos estocásticos mais abstratos que modelos CTMC discretos descritos na seção anterior. Em um modelo CTMC com níveis, os estados do sistema são caracterizados por intervalos de concentrações, discretizados em um dado número de níveis para cada espécie. A grande vantagem desta abordagem é que o espaço de estados é mais reduzido, possibilitando a análise de modelos maiores quando se emprega algoritmos de verificação de modelos exatos, como o PRISM. Dessa forma, se o número de

```

begin parameters
#rate constants
r1 170000
r2 1
r3 0.1

#initial amount of free A and free B
nA 301
nB 301
end parameters

begin molecule types
A(n,state~F~C~D)
B(n,state~F~C)
end molecule types

begin seed species
A(n,state~F) nA
B(n,state~F) nB
end seed species

begin reaction rules
A(n,state~F) + B(n,state~F) <-> A(n!1,state~C).B(n!1,state~C) r1,r2
A(n,state~F) -> A(n,state~D) r3
end reaction rules

begin observables
Molecules oA A(n,state~D)
Molecules oB B(n,state~F)
Molecules oAB A(n!1,state~C).B(n!1,state~C)
end observables

generate_network({overwrite=>1});
simulate_ssa({suffix=>ssa,t_end=>1,n_steps=>100});

```

Figura 5.6. Modificações no modelo populacional BIOLAB dado na química discreta para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5 \text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11} \text{ M}^{-1}\text{s}^{-1}$, $r2 = 1 \text{ s}^{-1}$, $r3 = 0,1 \text{ s}^{-1}$ e um volume de 10^{-18} litros.

representantes das espécies no modelo baseado na química discreta, apresentado na seção anterior, é muito grande, tornando impraticável a verificação usando a ferramenta PRISM, a abordagem baseada em níveis é uma alternativa de modelagem para reduzir o tamanho do espaço de estados. A proposta da abordagem é lidar com grupos de moléculas ao invés de lidar com cada molécula individualmente.

Informalmente, em um modelo CTMC com níveis cada espécie X é caracterizada por um número de níveis equidistantes, de 0 (a espécie está ausente) até um número máximo N_X . A distância de um nível para o outro é chamada de tamanho do passo h , determina o intervalo de concentração e expressa a granularidade do sistema. Quanto menor h , maior o número de níveis e de estados do modelo. Neste trabalho discutiremos apenas o caso em que o valor de h é comum a todas as espécies (elas estão em um mesmo compartimento ou em compartimentos de mesmo volume). Dada uma máxima

concentração M_X para a espécie X em molaridade, o nível máximo é dado por $N_X = \lceil M_X/h \rceil$. Se l_X é o nível atual da espécie X , sua concentração em molaridade ($[X]$) é dada por:

$$[X] = l_X \times h \quad (5.6)$$

Além disso, a constante de taxa de uma reação química pode ser calculada como segue:

$$r_i'' = \frac{r_i}{h} \quad (5.7)$$

onde r_i é a constante de taxa da reação i dada em termos de molaridade. Ciocchetta e Guerriero [2009] discutem em detalhes como derivar essa equação e como lidar com as situações em que as espécies estão em compartimentos distintos e de tamanhos diferentes.

Para fins de ilustração, considere o sistema de reações exemplo e novamente os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5 \text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11} \text{ M}^{-1}\text{s}^{-1}$, $r2 = 1 \text{ s}^{-1}$, $r3 = 0,1 \text{ s}^{-1}$ e $h = 2,5 \times 10^{-5} \text{ M}$. Os níveis máximos de A e B são ambos $N_A = N_B = \frac{0,5 \times 10^{-3}}{2,5 \times 10^{-5}} = 20$. As novas constantes de taxa para as três reações, dadas na abordagem baseada em níveis são, respectivamente:

$$r1'' = \frac{10^{11}}{2,5 \times 10^{-5}} = 4,0 \times 10^{15} \quad (5.8)$$

$$r2'' = \frac{1}{2,5 \times 10^{-5}} = 4,0 \times 10^4 \quad (5.9)$$

$$r3'' = \frac{0,1}{2,5 \times 10^{-5}} = 4,0 \times 10^3 \quad (5.10)$$

Além disso, é importante destacar que $\langle X_j \rangle$ na Equação 5.1 que determina como aplicar a lei cinética de ação das massas, corresponde à concentração em termos de molaridade obtida pela Equação 5.6.

A Figura 5.7 apresenta um modelo populacional baseado em níveis para o sistema de reações exemplo sendo considerado, dado os parâmetros listados anteriormente.

5.5.1 Imprecisão nos Modelos CTMC com Níveis

Aqui vale a pena discutir a imprecisão introduzida ao discretizar as concentrações das espécies nos modelos CTMC em tamanhos de passos h . Todos os eventos que somente ocorram em intervalos menores que o passo sendo considerado estarão possi-

```

ctmc

const int N=20; //número máximo de níveis para moléculas

const double r1=4*pow(10,16);
const double r2=40000;
const double r3=4000;
const double h=0.000025;

module m_A
a: [0..N] init N;

[bind] a>0 -> (a*h) : (a'=a-1);
[rel] a<N -> 1 : (a'=a+1);
[deg] a>0 -> (a*h) : (a'=a-1);

endmodule

module m_B
b: [0..N] init N;

[bind] b>0 -> (b*h) : (b'=b-1);
[rel] b<N -> 1 : (b'=b+1);

endmodule

module m_AB
ab: [0..N] init 0;

[bind] ab<N -> 1 : (ab'=ab+1);
[rel] ab>0 -> (ab*h) : (ab'=ab-1);

endmodule

module start
[bind] true -> r1 : true;
[rel] true -> r2 : true;
[deg] true -> r3 : true;
endmodule

```

Figura 5.7. Modelo populacional baseado em níveis para o sistema de reações da Figura 5.1 considerando os seguintes parâmetros: $[A]_{inicial} = [B]_{inicial} = 0,5 mM$, $[AB]_{inicial} = 0$, $r1 = 10^{11} M^{-1}s^{-1}$, $r2 = 1 s^{-1}$, $r3 = 0,1 s^{-1}$ e $h = 2,5 \times 10^{-5} M$.

velmente ocultos. De fato, o exercício de determinar um tamanho de passo h suficiente para que não haja nenhuma propriedade de interesse que tenha seu valor alterado caso se assuma um valor menor que h , é ainda um problema em aberto e corresponde a imprecisão introduzida quando consideramos o modelo CTMC com níveis. Entretanto, essa imprecisão não é exclusiva da técnica de verificação de modelos CTMC com níveis. Na verdade essa questão é recorrente em várias técnicas de análise de modelos. Por exemplo, na simulação estocástica a variável tempo é que é discretizada. Nesse caso, eventos que ocorram em intervalos de tempo inferiores ao tamanho de passo sendo considerado estarão também ocultos. Além disso, diferente da abordagem determinís-

tica que só disponibiliza análises em termos de valores médios, os modelos CTMC com níveis apresentam a grande vantagem de serem baseados em um modelo estocástico e, por isso, as análises fornecem diferentes visões introduzidas pelos diversos caminhos do modelo.

5.6 Comparação dos Modelos Baseados em População usando a Química Discreta e Abordagem em Níveis

Nesta seção os modelos BIOLAB e PRISM baseados na química discreta e o modelo PRISM baseado em níveis descritos, respectivamente, nas figuras 5.5, 5.6 e 5.7 são usados para verificar a propriedade referente à *probabilidade da molécula A se ligar a molécula B*.

O interesse inicial é saber se essa probabilidade é maior que 50% no primeiro segundo de funcionamento do sistema, a partir do estado inicial. Essa propriedade pode ser expressa em CSL (especificação usada pelo PRISM) e PBLTL (especificação usada pelo BIOLAB), respectivamente como:

1. $'init' \Rightarrow \mathcal{P}_{\geq 0,5} [\mathbf{F}^{\leq 1} ab \geq 1]$
2. $\mathcal{P}_{\geq 0,5} \mathbf{F} \{oAB \geq 1\}$

O rótulo *'init'* é um identificador do PRISM que se refere ao conjunto de estados iniciais do modelo CTMC. Adicionalmente, para o caso do BIOLAB, é preciso definir o valor do parâmetro t_{end} , tempo de simulação, como 1. Destaca-se também que o termo *oAB* foi definido no bloco *observables* do modelo BIOLAB apresentado na Figura 3.13 e se refere ao número de complexos entre as moléculas *A* e *B* que foram formados.

A Tabela 5.2 apresenta os resultados da verificação. Para os modelos PRISM discreto e baseado em níveis o tempo de verificação é separado do tempo de construção do modelo. Além disso, para esses modelos é também apresentado o tamanho da matriz de taxas representada em MTBDD, o que fornece uma ideia dos recursos de memória empregados. Para o modelo BIOLAB, a tabela apresenta o tempo de verificação e o número de caminhos gerados para 5 execuções. Para todas as execuções os parâmetros empregados foram: α (Erro Tipo 1) = 0,01, β (Erro Tipo 2) = 0,01, n_{steps} (número de passos do simulador) = 100 e $\Delta = 0,005$. Adicionalmente, nas 5 execuções a resposta para a propriedade (2) foi **verdadeiro**.

Primeiramente, note que para todas as execuções do BIOLAB o número de caminhos do modelo gerados para que o algoritmo SPRT verifique a propriedade é sempre

Tabela 5.2. *Tamanho* do modelo em termos do número de nós da representação MTBDD, tempos de construção (T_c) e tempos de verificação (T_v) para modelos PRISM baseados na química discreta e em níveis. *Número de Caminhos* e tempo de verificação (T_v) para modelo BIOLAB baseado na química discreta.

<i>Modelo</i>	T_c (s)	T_v (s)	<i>Tamanho do Modelo</i>	<i>Número de Caminhos</i>
PRISM Discreto	7,6340	> 7 dias	1.314.057	-
PRISM Níveis	0,0270	1712,8810	5.186	-
BIOLAB Discreto 1	-	46,9830	-	346
BIOLAB Discreto 2	-	47,0910	-	346
BIOLAB Discreto 3	-	47,6310	-	346
BIOLAB Discreto 4	-	47,7190	-	346
BIOLAB Discreto 5	-	47,6030	-	346

o mesmo e em todos eles $oAB \geq 1$. Isso se deve ao fato da probabilidade real da molécula A se ligar a molécula B no primeiro segundo de funcionamento do sistema ser bem maior que o limiar 0,5 considerado. Além disso, destaca-se a maior eficiência do BIOLAB para verificar a propriedade de interesse, tendo um tempo médio de verificação de $47,4054 \pm 0,2797$ segundos. A pouca eficiência do PRISM para verificar a propriedade (1) se deve ao fato dela envolver intervalos de tempo. Para o modelo PRISM discreto a entrada > 7 dias na tabela indica que a propriedade não pôde ser verificada em um tempo menor que 7 dias. Para verificar esse tipo de propriedade, a ferramenta PRISM emprega a técnica de uniformização [de Souza et al., 1992]. Nesse caso o custo do algoritmo é quadrático no número de estados do modelo e linear em relação ao tempo usado no intervalo. Note, na tabela a seguir, que para verificar propriedades que não envolvam intervalos de tempo, a ferramenta rapidamente encontra a solução¹.

Tabela 5.3. Tempos de verificação (T_v) para modelos PRISM baseados na química discreta e em níveis. *Propriedade (3)*: $'init' \Rightarrow \mathcal{P}_{\geq 0,5} [\mathbf{F} ab \geq 1]$ e *Propriedade (4)*: $\mathcal{P}_{=?} [\mathbf{F} ab \geq 1]$.

Modelo	Propriedade (3)		Propriedade (4)	
	T_v (s)	<i>Resultado</i>	T_v (s)	<i>Resultado</i>
PRISM Discreto	5,4000	verdade	0,7960	1
PRISM Níveis	0,0120	verdade	0,0040	1

A propriedade (3) $\mathcal{P}_{\geq 0,5} [\mathbf{F} ab \geq 1]$ questiona se a probabilidade de haver uma ligação entre as moléculas A e B considerando todos os estados possíveis do sistema é maior ou igual a 0,5. Já a verificação da propriedade (4) $\mathcal{P}_{=?} [\mathbf{F} ab \geq 1]$ retorna a

¹Os tempos de construção do modelo e o tamanho do modelo foram omitidos na Tabela 5.3 porque são os mesmos que os apresentados na Tabela 5.2

probabilidade das moléculas A e B se ligarem. Destaca-se o fato do modelo PRISM baseado em níveis apresentar menores custos de tempo e memória, uma vez que é um modelo mais abstrato que o modelo PRISM baseado na química discreta. Como no modelo baseado em níveis considerado há 20 níveis, cada nível representa aproximadamente um grupo de 15 moléculas. Dessa forma, a granularidade de mudanças de 1 em 1 molécula são omitidas nesse modelo, diferentemente do que ocorre com o modelo baseado na química discreta.

Capítulo 6

Modelagem da Bomba de Sódio-Potássio

Este capítulo introduz e discute alguns modelos da Na,K-ATPase que empregam as técnicas descritas no Capítulo 5. Os modelos foram construídos utilizando as linguagens de descrição formal das ferramentas de verificação de modelos BIOLAB e PRISM. A construção dos vários modelos é um resultado do avanço metódico do trabalho em busca de novas soluções para tentar superar as limitações da modelagem em questão. Além disso, como sistemas biológicos em geral tendem a ser muito complexos, múltiplas descrições de um mesmo sistema em diferentes níveis de abstração são interessantes [Pinto et al., 2007].

6.1 A Bomba de Sódio e Potássio e seu Funcionamento

A bomba de sódio-potássio (Na,K-ATPase) é encontrada na membrana plasmática de praticamente todas as células animais e é responsável pelo transporte ativo de sódio e potássio através da membrana. Ela é uma proteína integral com duas subunidades que atravessam a membrana plasmática. A Na,K-ATPase é crucial no processo de criar e manter as diferenças de concentração existentes no citoplasma (baixas concentrações de sódio e altas concentrações de potássio) que são essenciais para as funções celulares básicas, tais como regulação do volume, transporte ativo secundário e excitabilidade [Lauger, 1991]. O papel central da bomba é refletido na energia investida nas reações que compõem seu mecanismo de funcionamento: cerca de 25% da energia total consumida de um ser humano em repouso [Lehninger et al., 2004]. A existência da Na,K-ATPase na membrana plasmática de animais foi primeiramente proposta por

Dean [1941] e muitas propriedades sobre ela têm sido esclarecidas, entretanto questões básicas sobre seu modo de funcionamento ainda permanecem abertas [Lauger, 1991].

A Figura 6.1 ilustra o mecanismo básico de funcionamento da Na,K-ATPase, que usa uma ATPase da membrana para mover dois íons de potássio de fora da célula (baixa concentração de potássio) para dentro da célula (alta concentração de potássio) e três íons de sódio de dentro (baixa concentração de sódio) para fora da célula (alta concentração de sódio). Esse movimento de íons contra o gradiente de concentração é permitido pela mudança de configuração estrutural da ATPase e quebra da molécula de ATP.

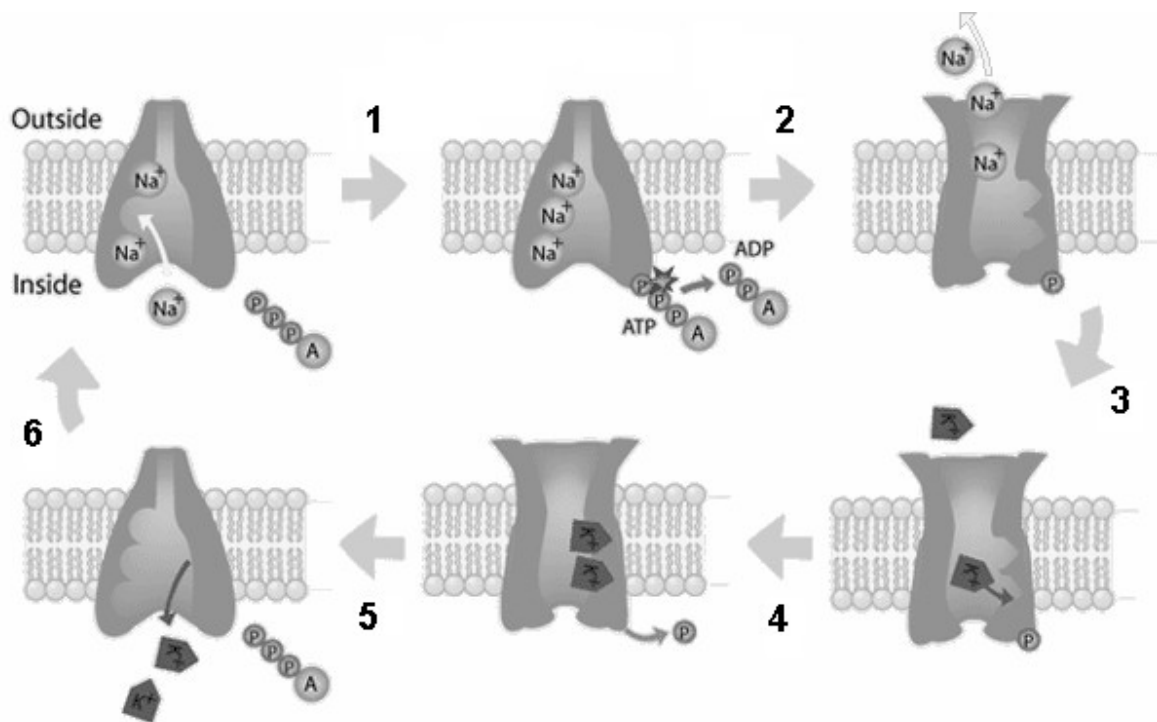


Figura 6.1. Mecanismo básico de funcionamento da bomba de sódio-potássio [Karp, 2008].

O ciclo de Albers-Post [Albers, 1967], mostrado na Figura 6.2, é um esquema de reações proposto pelos biólogos e é consistente com um grande número de observações experimentais. Note que o mapeamento entre o ciclo e a Figura 6.1 possui algumas imprecisões, dado que o ciclo é uma modelagem abstrata para o complexo funcionamento da bomba de sódio-potássio. No ciclo, as formas intracelular e extracelular de íons de sódio e potássio são explicitamente identificadas como Na_{in}^+ , Na_{out}^+ , K_{in}^+ e K_{out}^+ . P_i é um grupo de fosfato inorgânico e f_i e b_i são as constantes de taxas no sentido direto (funcionamento da bomba em condições normais) e reverso para o i – ésimo passo do

ciclo. Por exemplo, f_1 é a constante de taxa para a primeira reação no sentido direto do ciclo ($E_1 \cdot \text{ATP} + 3\text{Na}_{\text{in}}^+ \rightarrow \text{Na}_3 \cdot E_1 \cdot \text{ATP}$). Além disso, $A \cdot B$ significa que os substratos A e B são ligados por uma ligação não covalente e $A \sim B$ indica que o substrato A está covalentemente ligado ao substrato B .

De acordo com o ciclo de Albers-Post, a proteína da bomba pode assumir duas conformações principais, E_1 e E_2 , com os sítios de ligação voltados para dentro da célula e para fora da célula, respectivamente. Quando no estado E_1 , a bomba tem alta afinidade com os íons de sódio (Na^+) e, portanto, tende a se ligar a eles. Por outro lado, quando no estado E_2 , a bomba tem alta afinidade com os íons de potássio (K^+).

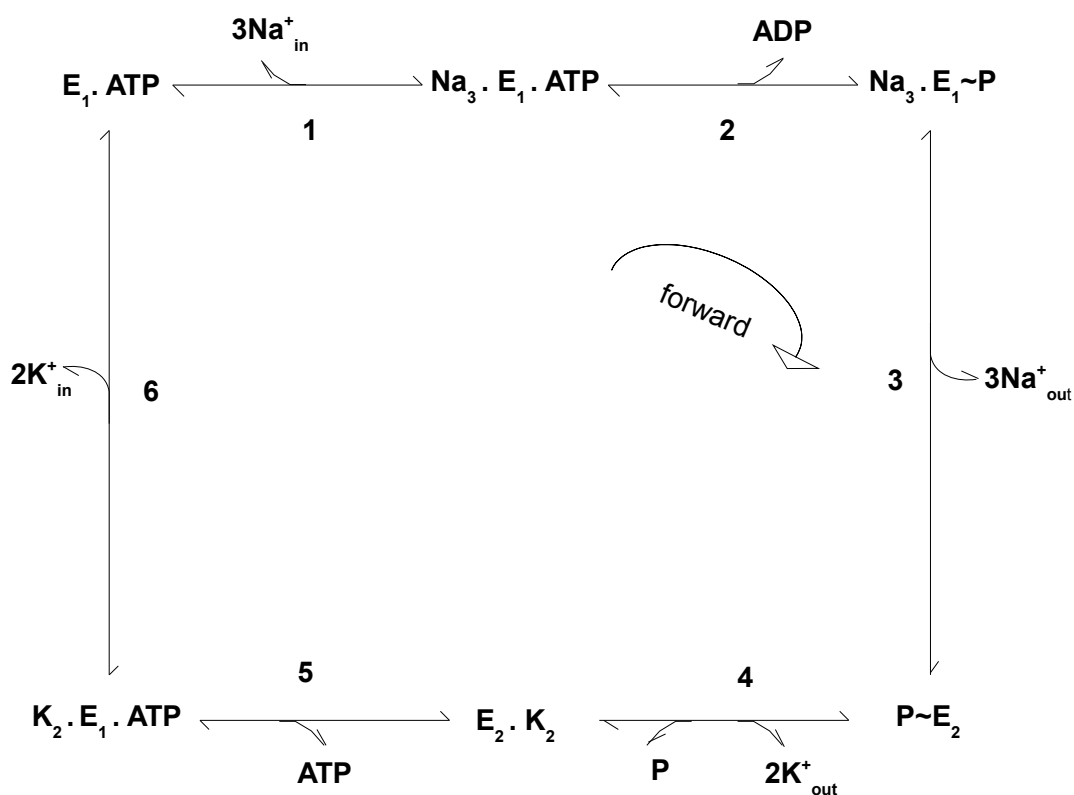
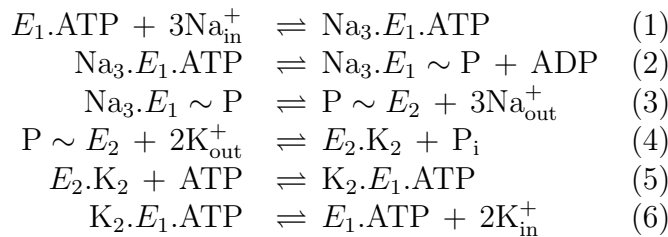


Figura 6.2. Esquema de reações para a Na,K-ATPase baseado no modelo de Albers-Post, retirado de Chapman et al. [1983]

O ciclo da Figura 6.2 pode ainda ser representado através de um conjunto de seis reações reversíveis, como mostra a Tabela 6.1. Um funcionamento da bomba no sentido normal (ou direto) é iniciado com a proteína da bomba em sua conformação E_1 ligada ao ATP. Então, três íons de sódio dentro da célula se ligam a essa estrutura (passo 1). Essa reação, por sua vez, estimula a hidrólise da molécula de ATP e, então, a liberação de ADP dentro da célula, formando uma proteína fosforilada intermediária

(passo 2). A liberação dos três íons de Na^+ fora da célula é finalizada pela mudança de conformação da proteína (E_2) (passo 3). Nessa nova configuração, a proteína da bomba têm alta afinidade com íons de potássio. Então, dois desses, localizados fora da célula, se ligam à proteína da bomba e, devido a essa reação, a proteína é desfosforilada (passo 4). Uma nova mudança de conformação em que a proteína da bomba se liga à molécula de ATP (passo 5) é seguida pela liberação dos dois íons de potássio dentro da célula (passo 6). Finalmente, a bomba volta a sua forma original que é capaz de reagir com Na^+ dentro da célula.

Tabela 6.1. Reações do modelo de Albers-Post



6.2 Parâmetros Iniciais dos Modelos e Considerações

Dados referentes às constantes de taxa das reações e às concentrações iniciais dos substratos envolvidos no ciclo de funcionamento da Na,K-ATPase, apresentado anteriormente, são listados na Tabela 6.2. As concentrações dos substratos ($[\text{Na}_{\text{in}}^+]$, $[\text{Na}_{\text{out}}^+]$, $[\text{K}_{\text{in}}^+]$, $[\text{K}_{\text{out}}^+]$, $[\text{ATP}]$, $[\text{P}_i]$ e $[\text{ADP}]$) e das constantes de taxas das reações foram obtidas de Chapman et al. [1983], enquanto o volume celular é de Hernández e Chifflet [2000]. Esses dados constituem o ponto de partida para explorar a dinâmica da bomba de uma maneira quantitativa usando PMC.

A Na,K-ATPase troca íons entre dois compartimentos: o intracelular e o extracelular. Neste trabalho é assumido que esses dois compartimentos têm o mesmo tamanho. Essa premissa é geralmente introduzida para simplificar a modelagem e as análises, principalmente quando dados precisos sobre os volumes dos compartimentos não estão disponíveis [Ciocchetta e Guerriero, 2009], como ocorre neste trabalho em relação ao volume extracelular. Chapman et al. [1983] e Hernández e Chifflet [2000], que fornecem os dados da Tabela 6.2, não mencionam o tamanho do compartimento externo usado nos experimentos.

Além disso, para fins de nomenclatura, nos modelos que serão apresentados, duas espécies iônicas (Na^+ e K^+), três espécies moleculares (ATP, P_i , ADP) e a proteína da bomba interagem através de seis reações (veja Tabela 6.1).

Tabela 6.2. Parâmetros fisiológicos normais associados ao esquema da Figura 6.2.

<i>Parâmetro</i>	<i>Valor</i>	<i>Unidade</i>
$[Na_{in}^+]$	0,02200	M
$[Na_{out}^+]$	0,14000	M
$[K_{in}^+]$	0,12700	M
$[K_{out}^+]$	0,01000	M
$[ATP]$	0,00500	M
$[P_i]$	0,00495	M
$[ADP]$	0,00006	M
f_1	$2,5 \times 10^{11}$	$M^{-3}s^{-1}$
f_2	10^4	s^{-1}
f_3	172	s^{-1}
f_4	$1,5 \times 10^7$	$M^{-2}s^{-1}$
f_5	2×10^6	$M^{-1}s^{-1}$
f_6	$1,15 \times 10^4$	s^{-1}
b_1	10^5	s^{-1}
b_2	10^5	$M^{-1}s^{-1}$
b_3	$1,72 \times 10^4$	$M^{-3}s^{-1}$
b_4	2×10^5	$M^{-1}s^{-1}$
b_5	30	s^{-1}
b_6	6×10^8	$M^{-2}s^{-1}$
volume celular	10^{-12}	l
temperatura	310	K

Para fins de comparação, duas propriedades foram verificadas para cada modelo apresentado. A primeira delas é a propriedade CSL seguinte, e os tempos para verificá-la serão referenciados nas seções seguintes como T_{v1} :

$$P_{\leq 0} [F((atp = 0) \& !('naInOver') \& !('kOutOver'))] \quad (6.1)$$

A propriedade questiona se o ATP acaba antes que os outros substratos necessários para o funcionamento da bomba (sódio dentro da célula e potássio fora da célula) se esgotem. Se a propriedade é verdadeira, em todos os caminhos do modelo sendo considerado não é possível que o ATP acabe e interrompa o funcionamento da bomba no sentido normal. Caso contrário, é possível que a falta de ATP seja o responsável pelo não funcionamento da bomba. Os termos 'kOutOver' e 'naInOver' são rótulos para, respectivamente, $kOut = 0 | kOut = 1$ e $naIn = 0 | naIn = 1 | naIn = 2$ e expressam as condições necessárias para que o potássio fora da célula e o sódio dentro da célula estejam esgotados. A necessidade de $kOut = 1$ para representar o esgotamento do potássio fora da célula, além do termo intuitivo $kOut = 0$, se deve ao fato que,

dependendo de certos parâmetros do modelo (volume celular ou tamanho do passo h), a concentração inicial de potássio fora da célula pode ter um valor ímpar e, dado que as mudanças na concentração desse substrato sempre ocorrem de duas unidades, o valor 0 nunca será atingido. Assim $kOut = 1$ também representará uma condição de esgotamento do potássio fora da célula, pois não há quantidade suficiente desse substrato para o funcionamento normal da bomba. O mesmo raciocínio é utilizado para justificar os termos $naIn = 1$ e $naIn = 2$ para representar o esgotamento do sódio dentro da célula, exceto pelo fato que as mudanças na concentração desse substrato sempre ocorrem de três unidades. Esses dois rótulos serão correntemente empregados ao longo deste capítulo e do seguinte.

Destaca-se que a propriedade (6.1) é uma propriedade de alcançabilidade e considerada qualitativa na ferramenta PRISM (por envolver $P_{\leq 0}$). Dessa forma, a verificação da mesma é completamente baseada na estrutura dos BDDs e, por isso, normalmente é rápida e eficiente, levando relativamente pouco tempo para ser verificada, como poderá ser observado nas próximas seções. Além disso, essa propriedade não pode ser diretamente verificada no BIOLAB, dado que não envolve intervalos de tempo, parâmetro essencial para essa ferramenta que se baseia em simulações finitas.

A segunda propriedade a ser verificada segue, sendo os tempos de verificação da mesma ao longo desse capítulo referenciados como T_{v2} :

$$P_{\leq 0,1} [F^{\leq 1} 'kOutOver'] \quad (6.2)$$

A propriedade questiona se a probabilidade do potássio fora da célula se esgotar no primeiro segundo de funcionamento da bomba é menor que 10%. Como será mostrado ao longo do capítulo, a verificação de propriedades que envolvam intervalos de tempo na ferramenta PRISM é normalmente mais custosa quando comparada com a propriedade (6.1). Além disso, justamente por envolver um intervalo de tempo, essa propriedade pode também ser verificada no BIOLAB.

6.3 Modelo baseado na Química Discreta no PRISM

O primeiro modelo da Na,K-ATPase foi construído na ferramenta PRISM utilizando a química discreta descrita na Seção 5.4, em que todos os substratos envolvidos no funcionamento da Na,K-ATPase são modelados usando a abordagem populacional, dado que o número de íons e moléculas nesse sistema é normalmente grande, dificultando a modelagem através da abordagem individual. O modelo é parcialmente apresentado na Figura 6.3, estando completamente disponível no Apêndice A.1. Como observado na Tabela 6.2 as concentrações iniciais dos substratos e as constantes de taxa são dadas em

termos de molaridade (M ou $\frac{mols}{l}$). Dessa forma, é preciso, primeiramente, converter todos esses parâmetros para a química discreta utilizando as equações 5.2 e 5.3.

```
ctmc

//constants
const double AV=6.022*pow(10,23);//avogadro number
const double V;//cell volume
const int NI=ceil(0.022*AV*V);//initial number of Na inside the cell
const int NO=ceil(0.14*AV*V);//initial number of Na outside the cell
const int naFlow=3;
...

//Modules
module na
  naIn : [0..(NI+NO)] init NI;//Number of Na ions inside the cell
  naOut : [0..(NI+NO)] init NO;//Number of Na ions outside the cell

  [r1] naIn>=naFlow -> pow(naIn,3) : (naIn'=naIn-naFlow);
  [rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);

  [r3] naOut<=NO+NI-naFlow -> 1 : (naOut'=naOut+naFlow);
  [rr3] naOut>=naFlow -> pow(naOut,3) : (naOut'=naOut-naFlow);
endmodule

...

module pump
  E1ATP : [0..1] init 1;
  E1ATPNa : [0..1] init 0;
  E1PNa : [0..1] init 0;
  E2P : [0..1] init 0;
  E2K : [0..1] init 0;
  E1ATPK : [0..1] init 0;

  //reaction1: 3 Na ions bind to pump enzyme
  [r1] E1ATP=1 & E1ATPNa=0 -> 1 : (E1ATP'=0) & (E1ATPNa'=1);
  [rr1] E1ATP=0 & E1ATPNa=1 -> 1 : (E1ATP'=1) & (E1ATPNa'=0);
  ...
endmodule

// base rates
const double r1rate = 2.5*pow(10,11)/(pow((V*AV),3));
const double r2rate = 10000;
...

// module representing the base rates of reactions
module base_rates
  [r1] true -> r1rate : true;
  [rr1] true -> r1rate : true;
  ...
endmodule
```

Figura 6.3. Primeiro modelo discreto da Na,K-ATPase no PRISM

Nesse modelo há um módulo para cada espécie envolvida no processo de funcionamento da bomba: **na**, **k**, **atp**, **adp**, **p** e **pump** (a proteína da bomba). No módulo **na**, por exemplo, existem duas variáveis finitas, *naIn* e *naOut*, que descrevem, respectivamente, o número de íons de sódio dentro e fora da célula. Já no módulo **pump** existem

seis variáveis discretas que representam os seis possíveis estados da proteína, de acordo com o ciclo da Figura 6.2. As concentrações iniciais de sódio dentro e fora da célula na química discreta são representadas pelas constantes NI e NO , respectivamente. Os comandos com rótulo **r1** representam as condições e atualizações necessárias para que a primeira reação do ciclo aconteça. Nesse caso, há uma redução no número de íons de sódio dentro da célula e a proteína da bomba muda para um estado que os íons de sódio estão a ela ligados. A taxa final dessa reação é $r1rate * 1 * pow(naIn, 3)$, onde $r1rate$ é a constante de taxa obtida após o processo de discretização. A multiplicação pelos valores 1 (há apenas uma bomba) e $pow(naIn, 3) \equiv naIn^3$ se deve a Lei de Ação das Massas descrita na Seção 5.3.

Vale a pena ressaltar que a concentração inicial das moléculas e as constantes de taxas na química discreta dependem do volume celular representado no modelo pela constante V . Se este parâmetro muda, esses valores irão também ser alterados. Ao utilizar o volume celular real (10^{-12} l), como listado na Tabela 6.2, o modelo se tornará intratável na ferramenta PRISM, devido às grandes quantidades de íons e moléculas e, conseqüentemente, do tamanho do modelo CTMC resultante. A Tabela 6.3 mostra a variação do tamanho do modelo em termos do número de nós da representação MTBDD da matriz de transição e dos tempos necessários para construí-lo e verificá-lo ao se variar o volume celular. Para fins ilustrativos, a tabela também apresenta a quantidade de íons de sódio dentro e fora da célula, dado o volume celular. Como esperado, quanto maior o volume celular, maior o tamanho do modelo e maior o tempo para construí-lo e verificá-lo. Para o volume 10^{-18} l, o modelo não pôde ser construído em um tempo menor que 7 dias.

Tabela 6.3. Variação do *tamanho* do model (número de nós do MTBDD que representa a matriz de transição), do número de íons de sódio dentro e fora da célula ($\#Na_{in}^+$ e $\#Na_{out}^+$), do tempo para contruir o modelo (T_c) e verificá-lo (T_{v1} e T_{v2}) devido às variações no volume celular.

<i>Volume (l)</i>	<i>Tamanho</i>	T_c (s)	T_{v1} (s)	T_{v2} (s)	$\#Na_{in}^+$	$\#Na_{out}^+$
10^{-20}	26881	48,5324	0,0050	43,6490	133	844
10^{-19}	351409	6745,7930	0,0460	977,1300	1325	8431
10^{-18}	?	> 7 dias	?	?	13249	84308

A propriedade (6.1), utilizada para obter os tempos de verificação T_{v1} é **verdadeira** quando o volume considerado é 10^{-20} ou 10^{-19} , o que significa que o ATP não se esgota. A propriedade (6.2), utilizada para obter os tempos de verificação T_{v2} , também é **verdadeira** para os volumes 10^{-20} ou 10^{-19} , indicando que a probabilidade do potássio fora da célula se esgotar no primeiro segundo de funcionamento da bomba é menor que 0,1.

Note também que no modelo apresentado nesta seção existe apenas uma proteína da bomba funcionando no sistema. As subseções seguintes apresentam a aplicação da abordagem individual e populacional caso haja necessidade de se modelar mais que uma bomba no sistema.

6.3.1 Abordagem Individual

A Figura 6.4 mostra uma extensão do modelo anterior de maneira a permitir duas bombas de sódio-potássio usando a abordagem individual, discutida na Seção 5.1. O recurso de renomeamento e a construção *system ... endsystem* da ferramenta PRISM são empregadas para esse fim.

```

module pump2=pump [
  E1ATP=E1ATP2,
  E1ATPNa=E1ATPNa2,
  E1PNa=E1PNa2,
  E2P=E2P2,
  E2K=E2K2,
  E1ATPK=E1ATPK2
]
endmodule

//system definition (The pumps do not interact with each other)
system
(pump ||| pump2) || na || k || p || adp || atp || base_rates
endsystem

```

Figura 6.4. Extensão do modelo discreto da Na,K-ATPase no PRISM para permitir duas bombas usando abordagem individual.

6.3.2 Abordagem Populacional

Nesta subseção a abordagem baseada em população é empregada na modelagem de um sistema com duas bombas de sódio-potássio e comparada com a metodologia apresentada na subseção anterior, que utiliza a abordagem individual. As diferenças nos modelos ocorrem, sobretudo, no módulo **pump**, como mostra a Figura 6.5. Uma constante NP é introduzida para representar o número de bombas no modelo. É preciso também considerar a Lei de Ação das Massas, levando em conta a concentração das bombas. Isso é conseguido multiplicando as taxas de interação pelo número de bombas em cada estado que pode participar da reação. Além disso, é importante ressaltar que a construção *system ... endsystem* usada anteriormente na abordagem individual não é mais necessária.

A Tabela 6.4 lista o número de nós da representação MTBDD da matriz de transição, os tempos de construção do modelo e os tempos de verificação variando o número

```

...
const int NP;

...
module pump
  E1ATP : [0..NP] init NP;
  E1ATPNa : [0..NP] init 0;
  E1PNa : [0..NP] init 0;
  E2P : [0..NP] init 0;
  E2K : [0..NP] init 0;
  E1ATPK : [0..NP] init 0;

  //reaction1: 3 Na ions bind to pump enzyme
  [r1] E1ATP>0 & E1ATPNa<NP -> E1ATP : (E1ATP'=E1ATP-1) &
    (E1ATPNa'=E1ATPNa+1);
  [rr1] E1ATP<NP & E1ATPNa>0 -> E1ATPNa : (E1ATP'=E1ATP+1) &
    (E1ATPNa'=E1ATPNa-1);
...
endmodule

```

Figura 6.5. Extensão do modelo discreto da Na,K-ATPase no PRISM para permitir duas bombas usando abordagem populacional.

de bombas, tanto para a abordagem individual quanto para a abordagem populacional. Como esperado, a abordagem populacional é mais escalável em termos de recursos de memória (tamanho do modelo). Em termos de tempo de construção do modelo, a abordagem populacional também é mais escalável, o que é nitidamente percebido com o aumento do número de bombas (a partir de 8 bombas).

Para todas as configurações apresentadas na tabela a verificação da propriedade (6.1) resultou no valor **verdadeiro**. Em termos do tempo para verificar essa propriedade, entrada T_{v1} da tabela, não há diferenças significativas entre as duas abordagens. Entretanto, os tempos para verificar a propriedade (6.2) se diferem significativamente, indicando mais uma vez os ganhos da abordagem populacional (entradas na tabela com **> 7 dias** indicam que a propriedade não pôde ser verificada em um tempo menor que 7 dias).

Tabela 6.4. Variação do *tamanho* do modelo, em termos do número de nós da representação MTBDD, dos tempos de construção do modelo (T_c) e dos tempos de verificação do modelo (T_{v1} e T_{v2}) devido às variações do número de bombas (NB). O volume celular utilizado nos modelos foi de $10^{-20} l$.

NB	Abordagem Populacional				Abordagem Individual			
	<i>Tamanho</i>	T_c (s)	T_{v1} (s)	T_{v2}	<i>Tamanho</i>	T_c (s)	T_{v1} (s)	T_{v2}
1	26.881	49,6440	0,0050	44 s	26.881	47,0190	0,0050	44 s
2	107.720	63,0870	0,0100	11 min	118.763	45,8160	0,0100	18 min
4	707.736	87,4430	0,0390	3 h	1.077.634	64,8940	0,0370	40 h
6	2.449.574	137,9450	0,0930	23 h	5.580.492	120,3400	0,0990	> 7 dias
8	7.694.589	284,4660	0,2480	107 h	20.822.865	321,6320	0,3180	> 7 dias
10	17.607.709	783,4790	0,5310	> 7 dias	62.242.251	1047,9040	0,5190	> 7 dias

6.3.3 Discussões

Conforme apresentado, o modelo da Na,K-ATPase baseado na química discreta e construído na ferramenta PRISM é intratável quando se considera o volume celular real 10^{-12} litros (ver Tabela 6.3). Dessa forma, a solução adotada é realizar as análises considerando um volume celular reduzido, como por exemplo 10^{-20} litros, estratégia comum na modelagem de sistemas biológicos [Sauro et al., 2006]. Apesar dessa restrição, o modelo ainda preserva a proporção original das moléculas listada na Tabela 6.2. Além disso, a redução do volume celular pode ser vista como a redução das análises para um subespaço celular, como mostra a Figura 6.6, e espera-se que o estudo do mecanismo da bomba em uma célula virtualmente menor ajude a compreender melhor o seu funcionamento no espaço real.

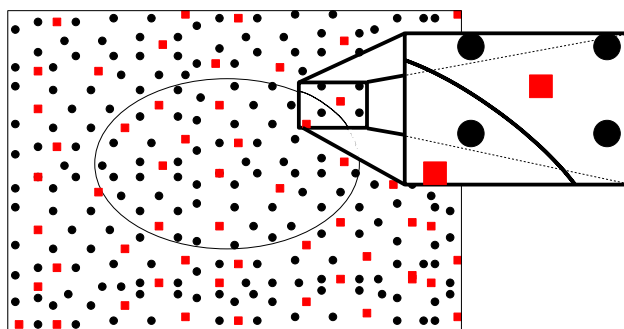


Figura 6.6. Ilustração da redução do volume celular no modelo discreto.

6.4 Abordagem Baseada em Níveis no PRISM

Um modelo alternativo no PRISM para a Na,K-ATPase que emprega a abordagem baseada em níveis e, portanto, não enfrenta o problema da redução do volume celular presente no modelo baseado na química discreta é discutido nesta seção. Dado que foi assumido que todas as espécies envolvidas no funcionamento da bomba de sódio-potássio estão em compartimentos de mesmo tamanho, é possível trabalhar com um modelo CTMC com níveis cujas espécies têm o mesmo tamanho do passo h [Ciocchetta e Guerriero, 2009].

A Figura 6.7 apresenta parte do modelo CTMC com níveis da Na,K-ATPase, cuja versão completa se encontra no Apêndice A.4. Como no caso discreto, há um módulo para cada espécie envolvida no funcionamento da bomba. Entretanto, agora as variáveis em cada módulo descrevem os níveis dos substratos. Por exemplo, as variáveis $naIn$ e $naOut$ descrevem, respectivamente, o nível atual de sódio dentro e fora da célula. Por sua vez, as constantes NI e NO descrevem os níveis iniciais de sódio dentro e

fora da célula. Novamente, os comandos rotulados como **r1** representam as ações necessárias para que a reação 1 mostrada na Tabela 6.1 aconteça no sentido direto. A taxa final desta reação é agora dada por $r1rate \times (E1ATP * h) \times pow(naIn * h, 3)$, onde $r1rate$ é a constante de taxa obtida após aplicar a Equação 5.7, necessária para construir um modelo CTMC com níveis. A multiplicação pelos valores $(E1ATP * h)$ e $pow(naIn, 3) \equiv naIn^3$ se deve a Lei de Ação das Massas. Note que $naIn * h$ e $E1ATP * h$ são os valores das concentrações atuais (ver Equação 5.6) de sódio dentro da célula e da proteína da bomba em seu estado $E1$ e ligada não covalentemente ao ATP.

Vale ressaltar que a escolha do tamanho do passo h determina tanto a precisão do modelo, quanto o seu tamanho. Se por um lado um h pequeno favorece a representação mais precisa das concentrações das espécies, minimizando os problemas discutidos na Seção 5.5.1, por outro um valor pequeno do passo pode produzir um grande número de níveis. Assim é preciso assumir um compromisso entre precisão (valor de h) e a tratabilidade do modelo. A Tabela 6.5 mostra a variação do tamanho do modelo da Na,K-ATPase, em termos de nós da representação MTBDD, e do tempo necessário para construí-lo e verificá-lo devido às variações no valor de h .

Tabela 6.5. Variação do *tamanho* do modelo, em termos do número de nós da representação MTBDD, do tempo para construir o modelo ($Tempo_c$) e dos tempos de verificação ($Tempo_{v1}$ e $Tempo_{v2}$) devido às variações do tamanho do passo (h).

h em M	<i>Tamanho</i>	T_c (s)	T_{v1} (s)	T_{v2} (s)
0,0003	14810	13,3520	0,0030	27,0970
0,0002	21600	30,0054	0,0050	37,0630
0,0001	46006	162,1990	0,0080	70,4280
0,00009	51176	202,1918	0,0080	96,3080
0,00008	59152	246,5536	0,0090	87,5210
0,00007	69192	361,3644	0,0140	134,8190
0,00006	82709	510,0978	0,0130	121,4810
0,00005	99944	648,9052	0,0190	203,8520

Para todos os valores de h apresentados na tabela, a resposta da verificação da propriedade (6.1), a qual foi utilizada para obter os tempos de verificação T_{v1} , é **verdade**, indicando que o ATP nunca se esgota. De maneira similar, a resposta da verificação da propriedade (6.2), utilizada para obter os tempos T_{v2} , é **verdade**, revelando que a probabilidade do potássio fora da célula se esgotar no primeiro segundo de funcionamento do sistema é menor ou igual a 0,1.

```

ctmc

//constants
const double h;
const int NI=ceil(0.022/h); //initial number of Na's levels inside the cell
const int NO=ceil(0.140/h); //initial number of Na's levels outside the cell
const int naFlow=2; //Na ions that go through membrane
int NP=1; //number of pump levels

module na
naIn : [0..(NI+NO)] init NI; //Number of Na inside of cell
naOut : [0..(NI+NO)] init NO; //Number of Na outside of cell

[r1] naIn>=naFlow -> pow(naIn*h,naFlow) : (naIn'=naIn-naFlow) ;
[rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);
...
endmodule
...

module pump
E1ATP : [0..NP] init NP;
E1ATPNa : [0..NP] init 0;
E1PNa : [0..NP] init 0;
E2P : [0..NP] init 0;
E2K : [0..NP] init 0;
E1ATPK : [0..NP] init 0;

//reaction1: 3 Na ions bind to pump
[r1] E1ATP>0 & E1ATPNa<NP -> E1ATP*h : (E1ATP'=E1ATP-1) &
(E1ATPNa'=E1ATPNa+1);
[rr1] E1ATP<NP & E1ATPNa>0 -> E1ATPNa*h : (E1ATP'=E1ATP+1) &
(E1ATPNa'=E1ATPNa-1);

...
endmodule

// base rates
const double r1rate = 2.5*pow(10,11)/h;
const double rr1rate = 100000/h;
...

// module representing the base rates of reactions
module base_rates
[r1] true -> r1rate : true;
[rr1] true ->rr1rate : true;
...
endmodule

```

Figura 6.7. Modelo baseado em níveis da Na,K-ATPase no PRISM.

6.4.1 Discussões

O modelo CTMC baseado em níveis da Na,K-ATPase não enfrenta a restrição presente no modelo baseado na química discreta, que não consegue lidar com volumes celulares de dimensões reais. Em contrapartida, esta abordagem trabalha com grupos de íons ou moléculas de tamanho h , ao invés de partículas individuais, como é mostrado na Figura 6.8

Dessa forma, conforme já mencionado, o modelo CTMC com níveis pode apresentar

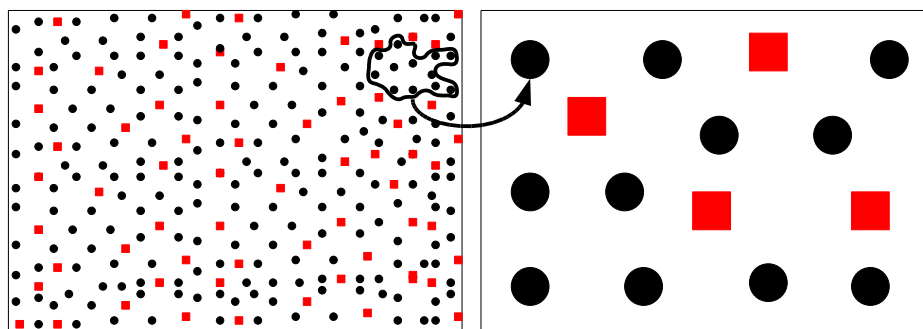


Figura 6.8. Ilustração do agrupamento das moléculas e íons no modelo baseado em níveis.

o problema discutido na Seção 5.5.1, sendo que certos eventos só podem acontecer quando o tamanho do passo h é menor que um dado limiar. Logo, a escolha de um valor de h pode alterar a resposta do algoritmo de verificação e determinar qual o valor de h ideal não é uma tarefa trivial. Entretanto, a proposta de redução do volume celular na abordagem discreta, apresentada anteriormente, também pode levar a escolha de um volume que pode alterar a resposta do algoritmo de verificação.

Nesse contexto, é interessante também discutir a questão multiescalar da Na,K-ATPase: a quantidade de íons dentro e fora da célula é significativamente maior que o número de bombas na membrana da mesma. Assim, se por um lado um h pequeno favorece a representação mais precisa das espécies com concentração baixa, por outro um valor pequeno do passo pode produzir um grande número de níveis para as espécies em maior concentração. Uma solução estudada para lidar com a multiescalaridade da Na,K-ATPase, caracterizada pela coexistência de espécies com níveis de concentração em diferentes escalas, é a utilização de dois tamanhos de passo diferentes: h_1 , o passo menor, e h_2 , o passo maior. Toda vez que uma reação envolver reagentes do grupo de menor concentração, como por exemplo a proteína da bomba, o passo menor h_1 deve ser usado. Caso contrário, o passo maior h_2 é empregado. Acredita-se que com essa metodologia o modelo se torne mais preciso e também mais escalável em termos de tempo de verificação e de memória empregada. Entretanto, ela não produz efeitos significativos para o sistema da Na,K-ATPase, sendo mais conveniente usar somente o tamanho de passo menor para h_1 . Isso se deve ao fato que nesse sistema biológico não há reações que envolvam somente espécies do grupo de maior concentração, ou seja, não envolvam a proteína da bomba. O Capítulo 8 mostra a aplicação desta abordagem em outro sistema de transporte de íons, sendo os resultados, embora preliminares, positivos no que diz respeito a escalabilidade em termos de recursos de memória.

6.5 Modelo no BIOLAB

Um terceiro modelo para a Na,K-ATPase emprega a abordagem da química discreta e é construído no BIOLAB. Parte do modelo é apresentado na Figura 6.9. A versão completa do modelo encontra-se disponível no Apêndice A.5. Como nos modelos discretos construídos na ferramenta PRISM, é preciso também transformar as constantes de taxa e concentrações dos substratos listadas na Tabela 6.2 para a química discreta descrita na Seção 5.4. Para isso as equações 5.2 e 5.3 novamente foram empregadas.

```

begin parameters
Na      6.02214179e23 #avogadro constant
V       1e-20 #cell volume
DIV     Na*V

# concentrations
sNaI    0.022*DIV #initial amount of Na inside cell
sNaO    0.14*DIV #initial amount of Na outside cell
spump   1 #number of pumps
...

# rate constants
sT1     2.5e11 / DIV^3
sR1     100000
...

end parameters

begin molecule types
pump(f~A~V,a,n1,n2,n3,k1,k2)
Na(s~I~0~C,n)
...
end molecule types

begin seed species
pump(f~A,a!1,n1,n2,n3,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4) spump
Na(s~I,n)                sNaI
Na(s~0,n)                sNaO
...
end seed species

begin reaction rules

# E1.ATP + 3Na <-> Na3.E1.ATP
pump(f~A,a!1,n1,n2,n3,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4) +
Na(s~I,n) + Na(s~I,n) + Na(s~I,n) <-> pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).
A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4).Na(s~C,n!5).
Na(s~C,n!6).Na(s~C,n!7) sT1,sR1

...
end reaction rules

...

```

Figura 6.9. Modelo discreto da Na,K-ATPase no BIOLAB.

No modelo do BIOLAB da Na,K-ATPase, no bloco *parameters* são definidos os valores iniciais das quantidades discretas de cada substrato envolvido no funcionamento

da bomba (como, por exemplo, **sNaI**, **sNaO** e **spump**) e também das constantes de taxa de cada reação listada na Tabela 6.1 (**sT1** e **sR1** são as constantes de taxa para a reação 1 no sentido direto e reverso, respectivamente). No bloco *molecule types* são definidas a estrutura de cada espécie no modelo: **pump** (a proteína da bomba), **Na** (íon de sódio), **K** (íon de potássio), **P** (fosfato) e **A** (adenosina). As moléculas do tipo **P** e **A** se ligam para formar ATP e ADP. Como exemplo, a espécie definida como **pump**, pode ter duas formas: **A**, quando seus sítios de ligação estão voltados para dentro da célula (forma E_1), e **V**, quando seus sítios de ligação estão voltados para fora da célula (forma E_2). Além disso, existem três sítios de ligação para íons de sódio (**n1**, **n2** e **n3**), dois sítios para íons de potássio (**k1** e **k2**) e um sítio de ligação **n** para as demais ligações (com ATP ou fosfato, por exemplo). Já a espécie definida como **Na** pode estar em três estados: **I**, dentro da célula, **O**, fora da célula, e **C**, quando ligada à proteína da bomba. Ela também possui um sítio de ligação **n** para se ligar à proteína da bomba. Por fim, as regras para as seis reações que sumarizam o funcionamento da bomba (ver Tabela 6.1) são definidas no bloco *reaction rules*. Como exemplo, é mostrado a regra da primeira reação do ciclo: espécies do tipo **pump**, quando no estado **A** e ligadas ao ATP (ligação de três moléculas do tipo **P** e uma do tipo **A**), se ligam à três representantes da espécie **Na**. Aqui, vale ressaltar que não é preciso multiplicar as taxas pelas concentrações de cada substrato envolvido na reação, pois isso é implicitamente considerado nos modelos construídos na ferramenta BIOLAB.

Da mesma forma que o modelo baseado na química discreta e construído no PRISM, variações no volume celular podem provocar alterações nas concentrações iniciais dos substratos. A Tabela 6.6 lista os tempos médios T_{v2} para verificar a propriedade $P_{\leq 0,1} [F^{\leq 1} \text{ 'kOutOver' }]$ usando o modelo no BIOLAB, bem como o número médio de caminhos gerados pela ferramenta, ao se variar o volume da bomba ¹. Os valores médios são apresentados devido ao não determinismo do algoritmo, que se baseia em simulações estocásticas.

Destaca-se aqui que a propriedade (6.1) definida no início deste capítulo e usada para obter os tempos T_{v1} nos modelos PRISM apresentados anteriormente não pode ser diretamente verificada no BIOLAB, dado que é uma propriedade que não envolve intervalos de tempo e considera todos os estados possíveis do modelo.

A resposta do algoritmo para a propriedade (6.2) é sempre **verdade**, indicando que a probabilidade do potássio fora da célula se esgotar no primeiro segundo de funcionamento da bomba é menor que 10%. Note que não há efeitos significativos na variação do tempo para verificar a propriedade de interesse quando se varia o volume celular.

¹As médias das variáveis (tempo e número de caminhos) foram obtidas considerando cinco execuções do algoritmo BIOLAB e com um intervalo de confiança (IC) de 90% no formato *média ± IC*. Isso significa que com 90% de certeza, a média da variável está no intervalo [*média - IC, média + IC*]

Tabela 6.6. Variação do tempo para verificar o modelo da Na,K-ATPase no BIOLAB devido às variações no volume celular. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$.

<i>Volume (l)</i>	T_{v2} (s)	<i>Número de Caminhos Gerados</i>
10^{-20}	133,2400 \pm 0,0964	414 \pm 0
10^{-19}	133,3288 \pm 0,1613	414 \pm 0
10^{-18}	133,4360 \pm 0,1787	414 \pm 0
10^{-12}	133,5336 \pm 0,2555	414 \pm 0

É possível, inclusive, verificar a propriedade com um volume celular real (10^{-12} l) em tempo hábil. Isso se deve ao fato, como já mencionado, que o BIOLAB não constrói o modelo matemático implícito (grafo) que representa o sistema modelado. Também destaca-se o fato do algoritmo ter verificado a propriedade desejada com um mesmo número de caminhos em todas as execuções e para todos os volumes. Isso ocorre porque o evento $F \leq 1$ 'kOutOver' é, de fato, raro, sendo baixa a probabilidade de recuperar um caminho do modelo em que ele acontece. Para todas as execuções, a propriedade F 'kOutOver' foi falsa nos 414 caminhos gerados do modelo, o que levou o algoritmo a decidir que a propriedade $P_{\leq 0,1} [F^{\leq 1} \text{ 'kOutOver' }]$ é verdadeira com um mesmo número de simulações para cada execução.

Entretanto, ao variar o número de bombas, como mostra a Tabela 6.7, há um efeito significativo no tempo de verificação (T_{v2}), mesmo com o algoritmo produzindo a mesma resposta para a propriedade (6.2) que é **verdadeiro**. Note também que novamente o número de caminhos gerados pelo algoritmo para verificar a propriedade de interesse não variou. Dessa forma, conclui-se que o simulador estocástico que gera os caminhos dos modelos descritos no BIOLAB é que demanda mais tempo quando há um aumento no número de bombas e não o algoritmo SPRT. Isso ocorre porque a possibilidade de reações aumenta com o aumento do número de bombas e a simulação se torna lenta.

6.5.1 Discussões

O modelo na ferramenta BIOLAB, apesar de ser factível mesmo empregando volumes reais, herda alguns dos problemas dos simuladores estocásticos, no qual são baseados. Primeiramente, a sua análise deve ser limitada em um tempo t_{end} finito, não havendo possibilidade de se verificar eventos que persistem indefinidamente, como oscilações. Além disso, dados t_{end} e o número de passos n_{steps} , parâmetros da ferramenta que determinam quando as concentrações dos substratos serão observadas (a cada intervalo $\frac{t_{end}}{n_{steps}}$), podem existir eventos que somente ocorrem em intervalos me-

Tabela 6.7. Variação do tempo para verificar o modelo da Na,K-ATPase no BIOLAB devido às variações no número de bombas (NP). O volume celular considerado é 10^{-20} l. α (Erro Tipo 1) = 0,01 e β (Erro Tipo 2) = 0,01. Além disso, o número de passos do simulador (n_{steps}) é 100 e $\Delta = 0,005$.

NP	T_{v2} (s)	Número de Caminhos Gerados
1	133,2400 ± 0,0964	414 ± 0
2	134,8448 ± 0,2563	414 ± 0
4	137,6754 ± 0,1961	414 ± 0
6	140,1674 ± 0,3398	414 ± 0
8	142,9634 ± 0,1458	414 ± 0
10	145,1922 ± 0,1361	414 ± 0
100	253,8696 ± 0,1822	414 ± 0
1000	1632,0940 ± 3,6001	414 ± 0

nores que $\frac{t_{end}}{n_{steps}}$ e, portanto, podem não ser capturados pelo algoritmo de verificação. Entretanto, a verificação das propriedades que envolvem intervalos de tempos, como a propriedade (6.2), é bem mais rápida no BIOLAB quando comparada com os modelos PRISM baseados na química discreta.

Outro aspecto a se destacar é que o BIOLAB não é um algoritmo de verificação exato como o empregado pelo PRISM, que é baseado em métodos numéricos para obter a probabilidade com que a propriedade está segura com alta precisão. O BIOLAB é um método aproximado que trabalha por amostragem de um conjunto de caminhos do modelo e, portanto, podem não ser tão precisos quanto os modelos construídos na ferramenta PRISM. Apesar disso, seu uso é atrativo porque o objetivo é prover aos biólogos uma maneira barata de determinar aspectos interessantes de um sistema que devam ser verificados experimentalmente. Dessa forma, seu uso é recomendado se uma precisão tão alta não é essencial e o interesse não é saber a probabilidade exata de ocorrência da propriedade, mas sim se ela é maior ou menor que um limiar desejado.

6.6 Comparações

A Tabela 6.8 apresenta um resumo dos resultados da verificação de algumas propriedades, considerando as abordagens de modelagem apresentadas e também dois sistemas biológicos: as reações mostradas na Figura 3.9 e a bomba de sódio-potássio, modelada neste capítulo. Parâmetros para esses sistemas são variados, como volume e número de bombas. A seguir é apresentada uma breve descrição de cada um deles:

- *Reacao₁*: modelo de reações químicas entre as moléculas A e B (ver Figura 3.9) com apenas uma molécula de cada espécie. O estudo desse sistema pode ser

encontrado na Seção 3.3.3 do Capítulo 3;

- *Reacao₂*: modelo de reações químicas entre as moléculas A e B (ver Figura 3.9) em um volume de 10^{-18} litros. Os demais parâmetros considerados são $[A]_{inicial} = [B]_{inicial} = 0,5 \text{ mM}$, $[AB]_{inicial} = 0$, $r1 = 10^{11} \text{ M}^{-1}\text{s}^{-1}$, $r2 = 1 \text{ s}^{-1}$, $r3 = 0,1 \text{ s}^{-1}$. O modelo PRISM baseado em níveis para esse sistema emprega um valor de $h = 2,5 \times 10^{-5} \text{ M}$, que equivale a 15 moléculas, dado o volume de 10^{-18} litros. O estudo detalhado desse sistema pode ser encontrado na Seção 5.6 do Capítulo 5;
- *bomba₁*: modelo da Na,K-ATPase em um volume de 10^{-20} litros e considerando apenas uma bomba. Os demais parâmetros podem ser vistos na Tabela 6.2 deste capítulo. O modelo PRISM baseado em níveis para esse sistema emprega um valor de $h = 5$ íons ou moléculas, dado o volume de 10^{-20} litros;
- *bomba₂*: modelo da Na,K-ATPase em um volume de 10^{-12} litros e considerando apenas uma bomba. Os demais parâmetros podem ser vistos na Tabela 6.2. O modelo PRISM baseado em níveis para esse sistema emprega um valor de $h = 5 \times 10^{-5} \text{ M}$, que equivale a $6,022 \times 10^6$ íons ou moléculas, dado o volume de 10^{-12} litros;
- *bomba₃*: modelo da Na,K-ATPase em um volume de 10^{-20} litros e considerando 8 bombas. Os demais parâmetros podem ser vistos na Tabela 6.2.

A Tabela 6.8 fornece a sumarização dos resultados. O modelo que fornece um menor tempo de verificação, considerando um par constituído por um sistema biológico e uma propriedade de interesse, é destacado em negrito na tabela. Além disso, entradas com o símbolo - indicam que a propriedade de interesse não pôde ser verificada para o sistema biológico em questão com a abordagem de modelagem sendo considerada. Isso pode ocorrer, por exemplo, na verificação de propriedades sem intervalo de tempo empregando modelos BIOLAB ou quando o modelo PRISM baseado na química discreta é intratável.

Note que não existe uma abordagem de modelagem incondicionalmente melhor que a outra. Para cada uma delas, conforme já discutido, existem vantagens e desvantagens. O modelo PRISM baseado na química discreta muitas vezes é intratável, não sendo possível, por exemplo, verificar propriedades para a Na,K-ATPase considerando um volume real (sistema *bomba₂* da tabela). Entretanto, pode ser a melhor alternativa em certas situações, como para verificar a propriedade $P_{\leq 0} [F ((atp = 0) \& !('naInOver') \& !('kOutOver'))]$ no sistema *bomba₃*. Como o BIOLAB é um algoritmo de verificação de modelos baseado em simulações, que por natureza são finitas

Tabela 6.8. Tabela com resumo dos resultados da verificação dos vários modelos para o sistema de reações mostrado na Figura 3.9 e para a bomba de sódio e potássio. A coluna *Tempo* da tabela para modelos PRISM apresenta a soma dos tempos de construção e verificação. Adicionalmente, os valores *T* e *F* na coluna de *Resultado* indicam, respectivamente, se a propriedade sendo verificada é verdadeira ou falsa.

<i>Sistema Biológico</i>	<i>Propriedade</i>	<i>Modelo</i>	<i>Tempo (s)</i>	<i>Resultado</i>
<i>Reacao₁</i> 1 molécula	$\mathcal{P}_{\geq 0,5} [\mathbf{F}^{\leq 1} ab = 1]$	PRISM discreto	0,0010	T
		PRISM níveis	-	-
		BIOLAB	177,0120	T
<i>Reacao₂</i> $10^{-18} l$	$\mathcal{P}_{\geq 0,5} [\mathbf{F}^{\leq 1} ab \geq 1]$	PRISM discreto	-	-
		PRISM níveis	1712,9080	T
	BIOLAB	47,4054	T	
	$\mathcal{P}_{\geq 0,5} [\mathbf{F} ab \geq 1]$	PRISM discreto	8,0340	T
PRISM níveis		0,0390	T	
<i>bomba₁</i> $10^{-20} l$	$P_{\leq 0} [F ((atp = 0) \& !('naInOver') \& !('kOutOver'))]$	PRISM discreto	48,53274	T
		PRISM níveis	1,5810	T
		BIOLAB	-	-
	$P_{\leq 0,1} [F^{\leq 1} 'kOutOver']$	PRISM discreto	92,1814	T
PRISM níveis		14,2070	F	
<i>bomba₂</i> $10^{-12} l$	$P_{\leq 0} [F ((atp = 0) \& !('naInOver') \& !('kOutOver'))]$	PRISM discreto	-	-
		PRISM níveis	648,9242	T
		BIOLAB	-	-
	$P_{\leq 0,1} [F^{\leq 1} 'kOutOver']$	PRISM discreto	-	-
PRISM níveis		852,7572	T	
<i>bomba₃</i> $10^{-20} l$ 8 bombas	$P_{\leq 0} [F ((atp = 0) \& !('naInOver') \& !('kOutOver'))]$	PRISM discreto	284,7140	T
		PRISM níveis	-	-
		BIOLAB	-	-
	$P_{\leq 0,1} [F^{\leq 1} 'kOutOver']$	PRISM discreto	385.484,4660	T
PRISM níveis		-	-	
		BIOLAB	142,9634	T

no tempo, essa propriedade não pode ser verificada por ele, já que não envolve um intervalo de tempo que representa o tempo de simulação. Nesse caso, é preciso construir todo o espaço de estados alcançáveis do sistema, sendo o PRISM o algoritmo de verificação apropriado. Adicionalmente, o modelo *bomba₃* modela um número específico de bombas, sendo assim, pouco apropriado o uso da modelagem baseada em níveis. Além disso, a abstração introduzida por esse modelo pode acarretar no problema da imprecisão, discutido na Seção 5.5.1 do Capítulo 5, conforme pode ser visto na verificação da propriedade $P_{\leq 0,1} [F^{\leq 1} 'kOutOver']$ para o sistema *bomba₁*. Nesse caso, a

abstração, embora produza uma verificação mais rápida, acarreta em uma resposta incorreta. Finalmente, quando a propriedade de interesse envolve intervalos de tempo, a modelagem através do BIOLAB é geralmente indicada, embora, ainda sim, para alguns sistemas simples, o seu tempo de verificação seja superior aos das demais modelagens, como nos casos destacados na coluna *Tempos* da tabela, dado o seu custo fixo.

Assim, a escolha da modelagem mais adequada deve levar em conta o sistema sendo modelado e a propriedade que se deseja verificar.

Capítulo 7

Análise da Bomba de Sódio-Potássio

Esta seção mostra como a técnica de verificação de modelos pode ajudar na melhor compreensão da dinâmica da Na,K-ATPase. A ocorrência de certos eventos, como o esgotamento de substratos e a propriedade de reversibilidade da bomba, podem ser estudados através da técnica de maneira direta, enquanto que com as abordagens tradicionais de análise esses eventos podem não ser identificados. Também um entendimento da dinâmica da bomba em certas configurações em termos de **tendências de reações** é apresentado. Esse estudo permite compreender, por exemplo, porque a reversibilidade da bomba necessita de longos tempos para ocorrer e, mais ainda, possibilita a identificação dos substratos e configurações da proteína da bomba que são os grandes responsáveis pelo lento funcionamento da mesma no seu sentido normal.

7.1 Modelo Empregado

As análises desta seção empregam o modelo baseado na química discreta da Na,K-ATPase construído na ferramenta PRISM, o qual foi apresentado na Seção 6.3 do capítulo anterior. Um dos motivos que levou a escolha desse modelo se deve ao fato de que algumas das propriedades a serem verificadas dizem respeito a eventos que persistem indefinidamente e que não são limitadas a um intervalo de tempo e, portanto, só podem ser tratadas por modelos construídos na ferramenta PRISM. Além disso, o modelo CTMC baseado na química discreta construído na ferramenta PRISM, apesar de ter a restrição de utilizar um volume reduzido, estratégia comum na modelagem de sistemas biológicos [Sauro et al., 2006], permite um estudo cujo comportamento mantém as proporções reais dos substratos ao longo do tempo. Vale ressaltar, entretanto, que o modelo CTMC baseado em níveis poderia ser similarmente empregado nos estudos realizados ao longo deste capítulo e o modelo BIOLAB poderia também ser empregado para verificar algumas propriedades que consideram intervalos de tempo.

O volume celular empregado ao longo do capítulo é 10^{-20} litros. Evidentemente, quanto maior o valor do volume mais acurado será o modelo. Entretanto, esse valor foi escolhido de maneira a não se obter tempos de verificação tão longos, o que permite explorar os modelos através de um número maior de análises, embora mesmo com o valor escolhido a verificação de certas propriedades leve cerca de 17 horas. Adicionalmente, o estudo apresentado aqui considera apenas uma bomba, embora todas as definições sejam extensíveis para modelos que considerem quantidades superiores de bombas.

7.2 Esgotamento do Potássio fora da Célula

As análises iniciais objetivam estudar alguns eventos no funcionamento da Na,K-ATPase que podem ser ocultados pelas abordagens tradicionais. O primeiro evento a ser estudado é o esgotamento de potássio fora da célula (K_{out}^+).

A Figura 7.1 apresenta a quantidade de íons de potássio fora da célula no decorrer do tempo para a abordagem que emprega ODE e um caminho do modelo da Na,K-ATPase obtido através de simulação estocástica¹. Observando a figura, na abordagem determinística o potássio fora da célula diminuiu durante os primeiros segundos de funcionamento da bomba e converge para o valor aproximado de 11 íons. Entretanto, este comportamento médio esconde alguns caminhos importantes, como aquele mostrado na mesma figura, no qual a quantidade de potássio fora da célula, depois de uma rápida diminuição, oscila em torno do valor médio e pode eventualmente se esgotar. Porém, esse caminho é apenas um dentre os diversos possíveis, e outros caminhos gerados pela simulação estocástica podem não capturar esse evento (potássio fora da célula se esgota), dado que ele ocorre com probabilidade considerável somente em tempos relativamente longos (durante os primeiros 10 segundos de funcionamento da bomba a probabilidade do evento ocorrer é $6,33 \times 10^{-3}$).

Assim, a primeira vista, as abordagens tradicionais podem ocultar o fato que o potássio fora da célula pode eventualmente se esgotar no modelo da Na,K-ATPase sendo considerado. Entretanto, a propriedade CSL a seguir

$$P \geq 1 [F \text{ 'kOutOver'}] \quad (7.1)$$

permite verificar se o potássio fora da célula de fato sempre acaba. A propriedade anterior é **verdade** para o modelo da Na,K-ATPase sendo considerado.

É possível ainda saber o tempo esperado para que esse evento aconteça. Para isso, é preciso inicialmente estender os modelos na linguagem PRISM definidos nas seções

¹Os caminhos foram obtidos usando a ferramenta BIONETGEN que pode gerar tanto simulações estocásticas quanto determinísticas do modelo

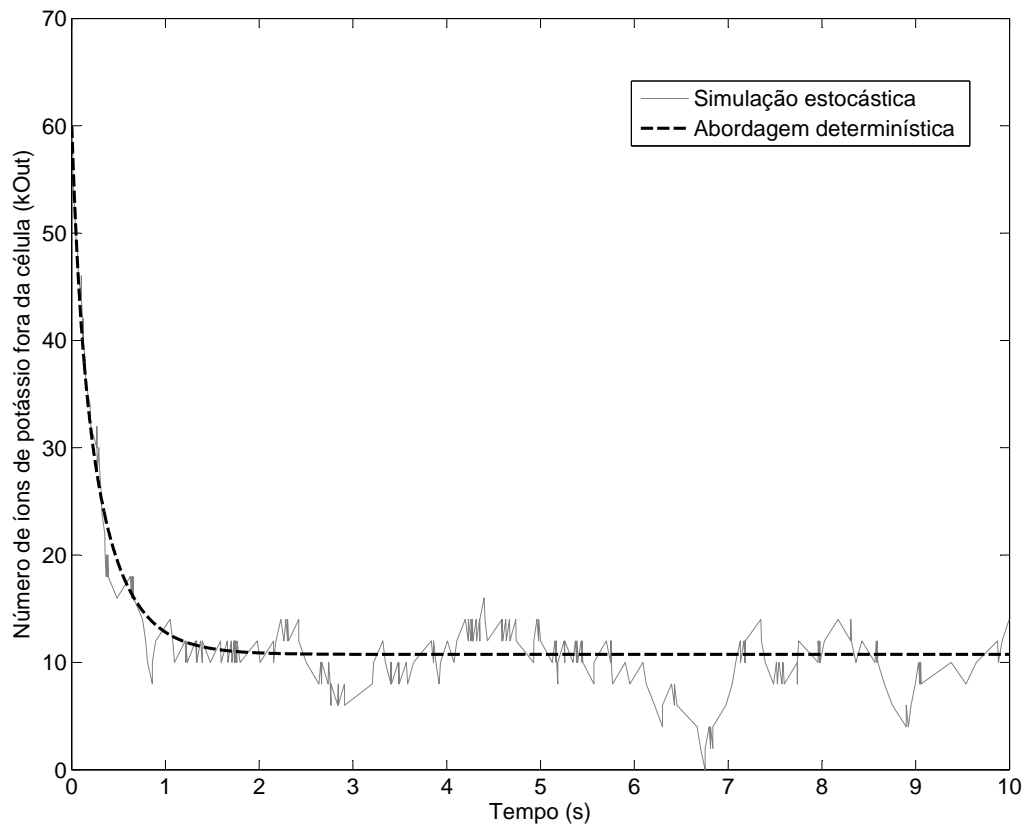


Figura 7.1. Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase.

anteriores, conforme é mostrado na Figura 7.2. O *reward* 'time' associa o valor 1 a cada estado do modelo.

```
//time
rewards "time"
true: 1;
endrewards
```

Figura 7.2. Definição do *reward* 'time' para modelos PRISM da Na,K-ATPase.

A propriedade (7.2) pode, então, ser usada para saber o tempo esperado para que esse o evento *potássio fora da célula se esgota* aconteça:

$$R\{\text{'time'}\} = ? [F \ \text{'kOutOver'}] \quad (7.2)$$

A resposta para essa propriedade é 1.287 segundos e o tempo que o PRISM levou para verificar a propriedade foi cerca de 194 segundos.

A propriedade (7.3), por sua vez, nos permite saber a probabilidade do potássio fora da célula acabar nas primeiras T unidades de tempo:

$$P = ? [F \leq T \text{ 'kOutOver'}] \quad (7.3)$$

Utilizando $T = 1.287$, o valor retornado foi 0,63. Com isso, podemos concluir que durante os primeiros 1.287 segundos a quantidade de potássio se esgotou em 63% dos caminhos do modelo².

Finalmente, as propriedades seguintes são usadas para saber sobre o tempo esperado mínimo e máximo para o sistema ir de um estado onde o valor de potássio fora da célula se esgotou para outro estado onde esse substrato não está mais esgotado:

$$R\{\text{'time'}\} = ? [F \text{ !('kOutOver')} \{ \text{'kOutOver'} \} \{max\}] \quad (7.4)$$

$$R\{\text{'time'}\} = ? [F \text{ !('kOutOver')} \{ \text{'kOutOver'} \} \{min\}] \quad (7.5)$$

Como possivelmente há mais de um estado onde o potássio fora da célula se esgotou, **min** and **max** são usados para retornar o tempo esperado mínimo e máximo para atingir um estado onde $!(kOutOver)$, dado o conjunto de todos os estados iniciais que satisfazem $kOutOver$. As respostas para as propriedades (7.4) e (7.5) são, respectivamente, 111 milisegundos e 14 milisegundos. Isso significa que é esperado que o sistema permaneça em um estado onde o potássio fora da célula está esgotado por no máximo 111 milisegundos e no mínimo 14 milisegundos.

O tempo esperado mínimo (min) é potencialmente útil para determinar o número de passos (n_{steps}) mínimo nos modelos BIOLAB, de forma a garantir que o evento de interesse seja sempre capturado. Nesse caso, sendo t_{end} o tempo de simulação, a relação $\frac{t_{end}}{n_{steps}} < min$ é desejada. A partir disso, a relação $n_{steps} > \frac{t_{end}}{min}$ é obtida. Se o número de passos for insuficiente, o evento de interesse poderá não ser capturado, dado que os momentos de observação da ferramenta podem não coincidir com aqueles em que o evento ocorre.

Esta seção mostrou que o esgotamento do potássio fora da célula, apesar de passível de ocorrência, é um evento raro, ou seja, ocorre com probabilidades consideráveis somente em longos períodos de tempo, tendo baixa probabilidade de ocorrência em curtos períodos de tempo, sendo, justamente por isso, difícil de ser identificado pelas abordagens tradicionais de análise. Entretanto, apesar de raro, esse evento pode ser de grande interesse biológico, caso traga consequências graves ao ser humano. Para fins de ilustração, a redução do potássio no meio extracelular abaixo de certos limia-

²O tempo para verificar a propriedade foi cerca de 17 horas.

res críticos caracteriza a doença conhecida como hipocalemia, que pode causar desde fraqueza muscular e arritmias cardíacas até, em casos mais extremos, a paralisia e a morte [Weiner e Wingo, 1997].

7.3 Retorno do Potássio fora da Célula à sua Quantidade Inicial

Outro evento que pode ser identificado com a técnica de verificação de modelos é se depois que o potássio fora da célula se esgota, sua quantidade irá retornar ao valor inicial KO . A propriedade (7.6) é usada para verificar se esse evento sempre acontece, enquanto a propriedade (7.7) é usada para determinar o tempo esperado para esse evento acontecer:

$$P \geq 1 [G \text{ ('kOutOver' } \Rightarrow P \geq 1 [F \text{ kOut} \geq KO])] \quad (7.6)$$

$$R\{\text{'time'}\} = ? [F \text{ kOut} = KO \text{ {'kOutOver'}}] \quad (7.7)$$

A resposta para a propriedade (7.6) é **verdadeiro**, enquanto o tempo esperado computado na propriedade (7.7) é cerca de 132.515 segundos. Ou seja, o potássio fora da célula, depois de se esgotar, sempre retorna ao seu valor inicial depois de um tempo esperado de 132.515 segundos.

Assim, dois eventos significativos no sistema *potássio fora da célula se esgota e potássio fora da célula retorna ao valor inicial* sempre acontecem e nos levam a pensar sobre suas recorrências a longo prazo, como é discutido na próxima seção.

7.4 Reversibilidade da Na,K-ATPase

Devido ao fato que todas as reações no modelo da Na,K-ATPase são reversíveis (ver Tabela 6.1) e, além disso, muitas das taxas dessas reações dependerem dos substratos presentes dentro e fora da célula, o mecanismo de funcionamento da bomba pode ser considerado automaticamente reversível, ou seja, as reações são tomadas no sentido direto ou reverso, dependendo de mudanças nas concentrações dos substratos. No sentido direto ou normal, a Na,K-ATPase usa uma molécula de ATP para transportar dois íons de potássio de fora para dentro da célula e três íons de sódio de dentro para fora da célula. O funcionamento da bomba nesse sentido fortalece as condições favoráveis para a reversão do sentido da bomba, conforme discutido na Seção 4.3: baixa concentração de ATP, alta concentração de ADP e P_i , além de altas concentrações de

sódio e potássio fora e dentro da célula, respectivamente. No sentido reverso, o ATP é produzido a partir de ADP e P_i , e os íons de potássio são transportados para fora da célula enquanto os de sódio para dentro da célula.

Sem perda de generalidade, o estudo desse comportamento reversível da bomba será realizado neste trabalho em termos da quantidade de potássio fora da célula, que será a espécie a ser observada. Esse comportamento reversível automático pode ser visto como uma oscilação infinita entre dois valores: a quantidade inicial de potássio fora da célula (KO) e a quantidade final desse substrato depois que o ciclo no sentido direto da bomba está completo e o ciclo no sentido reverso ainda não se iniciou. Nessa configuração final, o potássio fora da célula acaba, isto é, atinge seu valor mínimo ($kOutOver$). Esse comportamento reversivo pode ser expresso através da propriedade CSL que segue:

$$P \geq 1 [G((kOut=KO \Rightarrow (P > 0 [F \text{ 'kOutOver'}])) | \text{'kOutOver'} \Rightarrow (P > 0 [F \text{ kOut=KO}])))] \quad (7.8)$$

A propriedade questiona se a oscilação de $kOut$ entre os valores $kOutOver$ e KO nunca termina, ou seja, é verdade a longo prazo. O resultado dessa propriedade é **verdadeiro**, o que prova que os eventos *o potássio fora da célula se esgota e o potássio fora da célula atinge o valor inicial KO* acontecem indefinidamente, ou seja, a propriedade de reversibilidade automática da bomba é **verdade**. Essa propriedade da bomba não pode ser visualizada na Figura 7.1 porque o estudo do comportamento médio da bomba oculta alguns aspectos de sua reversibilidade. Mais ainda, com a técnica de verificação de modelos é possível estudar sobre esse comportamento a longo prazo, o que não pode ser facilmente conseguido através da geração e análise de trajetórias do modelo com simulações estocásticas.

7.5 Entendendo o Comportamento da Bomba através de Tendências

Esta seção investiga os motivos que levam ao lento funcionamento da bomba no sentido normal com o passar do tempo ou mesmo a sua interrupção. Com esse estudo será possível compreender, por exemplo, porque a probabilidade de ocorrência do esgotamento do potássio fora da célula ou da reversibilidade da bomba em curtos períodos de tempo é baixa. O estudo será realizado em termos das taxas das reações do ciclo mostrado na Figura 6.2. O objetivo é identificar as reações com menor probabilidade

de ocorrência (e, conseqüentemente, as com maior probabilidade de ocorrência) e compreender, de maneira detalhada, o comportamento da bomba através das mudanças nessas probabilidades de ocorrência ou tendências. Com isso, será possível encontrar os gargalos que impedem a bomba de funcionar no sentido normal e eliminá-los através do aumento da concentração de certos substratos ou mesmo da adição de novas reações ao modelo da Na,K-ATPase.

Seja s^k um estado do modelo da Na,K-ATPase em que a proteína da bomba se encontra na configuração $k \in \{E_1.ATP, Na_3.E_1.ATP, Na_3.E_1.ATP, P \sim E_2, E_2.K_2, K_2.E_1.ATP\}$ (ver Figura 6.2). A **tendência positiva ou ascendente** de uma reação r_i a partir do estado corrente s^k é definida como:

$$\uparrow r_i(s^k) = \begin{cases} 1, & \text{se } rate(s^k, s^k + r_i) / exit(s^k) \geq \xi \quad \text{e} \quad exit(s^k) \neq 0 \\ 0, & \text{caso contrário} \end{cases}$$

onde $exit(s^k) = \sum_{j \in M} rate(s^k, s^k + r_j)$, sendo M o conjunto finito de todas as reações do modelo possíveis a partir de s^k , e $0 < \xi < 1$ o limiar que indica a tendência positiva da reação. Além disso, $s^k + r_i$ representa um estado depois da reação r_i ocorrer a partir do estado s^k . Informalmente, uma tendência positiva para uma reação r_i a partir de um estado s^k indica que há uma probabilidade superior da reação r_i acontecer ao invés de qualquer outra reação r_j ($j \neq i$) de M , cuja taxa $rate(s^k, s^k + r_j) > 0$.

Dado que as quantidades dos demais substratos que participam do funcionamento da Na,K-ATPase, além da proteína da bomba, podem assumir diferentes valores ao longo da evolução do sistema, existem vários estados em que a proteína da bomba se encontra na configuração k , cada um dos quais se diferenciando pelos valores das variáveis que representam os demais substratos. Dessa forma, a seguinte propriedade poderia ser utilizada para verificar se a tendência da reação r_i acontecer é positiva, a partir de *qualquer* estado em que a configuração da proteína da bomba é k :

$$P \geq 1 \quad [G \quad (\bigwedge_{s_j^k \in S^k} \uparrow r_i(s_j^k))] \quad (7.9)$$

S^k é o conjunto de todos os estados do sistema em que a configuração da proteína da bomba é k .

Note, entretanto, que embora uma reação r_i possa sempre exibir tendência positiva quando a configuração corrente é k , se essa configuração muda, a tendência da reação r_i será 0. Isso porque, como observado na Figura 6.2 nem toda reação r_i pode acontecer dependendo da configuração k da bomba. Por isso, ao invés de verificar se uma reação r_i sempre exibe uma tendência positiva a partir de qualquer estado onde a configuração da bomba é k , é mais conveniente verificar se ela nunca exibirá uma tendência positiva,

exibindo, portanto, sempre uma **tendência negativa** a partir da configuração proteica k , como mostra a propriedade a seguir:

$$P \geq 1 \quad [G \quad \neg(\bigvee_{s_j^k \in S^k} \uparrow r_i(s_j^k))] \quad (7.10)$$

Assumindo um valor para o limiar ξ igual a 0,5, quando verdadeira para uma reação r_i e uma configuração proteica k , a propriedade anterior indica que a probabilidade da reação r_i ocorrer, sendo k a configuração atual da proteína da bomba, é sempre menor que 0,5, independente dos demais substratos envolvidos no funcionamento da Na,K-ATPase. Caso contrário, se a propriedade é falsa, para alguns (ou todos) estados onde k é verdade existe uma **tendência positiva** que a reação r_i ocorra.

Além disso, observando novamente o ciclo da Figura 6.2 é possível perceber que para cada uma das seis configurações k que a bomba pode estar, apenas duas reações podem ser tomadas: uma no sentido direto r_k e outra no sentido reverso rr_l , sendo $l = k - 1$ quando $k \neq 1$, ou $l = 6$, quando $k = 1$. Por exemplo, se a configuração da bomba é $\text{Na}_3.E_1.ATP$ ($k = 2$), somente as reações r_2 e rr_1 podem ser tomadas, levando a proteína da bomba para as configurações $\text{Na}_3.E_1 \sim P$ e $E_1.ATP$, respectivamente. Dessa forma, podem ocorrer somente três situações possíveis para as tendências dos pares de reações r_k e rr_l , a partir de uma configuração k da proteína da bomba, de acordo com as definições feitas até então:

1. a propriedade (7.10) é verdadeira somente para a reação r_k . Essa reação, portanto, apresenta uma tendência sempre negativa e rr_l exibe sempre uma tendência positiva;
2. a propriedade (7.10) é verdadeira somente para a reação rr_l . Essa reação, portanto, apresenta uma tendência sempre negativa e r_k exibe sempre uma tendência positiva;
3. a propriedade (7.10) é falsa tanto para r_k quanto para rr_l , o que significa que ambas reações apresentam tendências positivas e negativas ao longo da evolução do sistema, ou seja, dependendo da configuração dos demais substratos que participam do funcionamento da Na,K-ATPase.

7.5.1 Tendências nos Modelos PRISM

Para o modelo discreto do PRISM, fórmulas de tendência foram adicionadas para todas as reações do modelo da Na,K-ATPase utilizando recursos da ferramenta PRISM chamados de **labels** e **formulas**. Uma fórmula compreende a um nome (um identi-

ficador) seguida por uma expressão e é usada para evitar duplicação de código. Isso porque o nome da fórmula pode ser usado como um atalho para a expressão que representa em qualquer lugar do modelo onde uma expressão é aceita. Já um *label* permite rotular um conjunto de estados de interesse particular e podem somente ser utilizados na especificação das propriedades. Adicionalmente, diferentemente das fórmulas, os *labels* tem tipo booleano.

O código da Figura 7.3 ilustra o procedimento para computar as tendências para a primeira reação no ciclo da Figura 6.2 (r_1):

```
const double threshold;

//trend formulas
formula rate_r1 = ((E1ATP>0 & E1ATPNa<NP) ? r1rate*pow(naIn,naFlow)*E1ATP : 0);
...

formula exit_rate = rate_r1 + rate_r2 + rate_r3 + ... rate_r6 + rate_rr1 + rate_rr2 +
rate_rr3 + ... rate_rr6;

//trend label r1
formula rate_r1_d = (exit_rate=0 ? 0:rate_r1/exit_rate);

label "trend_r1_up" = (rate_r1_d>=threshold);
```

Figura 7.3. Tendências para a reação r_1 considerando o modelo PRISM discreto da Na,K-ATPase.

O valor da taxa final da reação r_1 é obtida pela fórmula **rate_r1** e é diferente de 0 somente quando a configuração corrente da bomba é E_1 .ATP. Nesse caso, o valor da taxa final para r_1 é determinado do mesmo modo como aquele descrito na Seção 6.3. Os valores das demais taxas são computados de maneira similar e a fórmula **exit_rate** representa a soma de todos os valores de taxas das reações possíveis a partir da configuração corrente da proteína da bomba. A probabilidade que r_1 seja tomada no estado corrente é dada pela fórmula **rate_r1_d**, enquanto o *label* **trend_r1_up** indica se r_1 tem uma tendência ascendente, ou seja, $\uparrow r_1$ é 1.

No caso do modelo CTMC com níveis, a diferença se dá simplesmente no cálculo das taxas de transição (fórmula **rate_r1**). Nesse caso, elas são obtidas conforme descrito na Seção 6.4. Já para o modelo BIOLAB, a verificação das tendências das reações é totalmente incorporada na especificação da propriedade. As versões completas das definições de tendências de taxas para os modelos da Na,K-ATPase se encontram no Apêndice B.

A propriedade CSL (7.11) pode ser utilizada para identificar no modelo PRISM quais as reações que sempre exibem uma tendência negativa durante a evolução do sistema:

$$P \geq 1 \quad [G \quad \neg('trend_ri_up')] \quad (7.11)$$

onde 'trend_ri_up' deve ser substituído pelo rótulo de tendência positiva da reação r_i , conforme apresentado no código 7.3. Por exemplo, a tendência positiva para a primeira reação da bomba na direção direta é 'trend_r1_up', enquanto o *label* 'trend_rr1_up' refere-se a tendência positiva para a primeira reação no sentido reverso.

A propriedade (7.12) pode ainda ser utilizada para identificar quando uma reação r_i , que pode exibir tanto uma tendência positiva quanto uma tendência negativa, passa a exibir a tendência positiva:

$$KO + 2 * ((R\{'plusKout'\} =? [F \quad 'trend_ri_up']) - (R\{'minusKout'\} =? [F \quad 'trend_ri_up'])) \quad (7.12)$$

Essa propriedade nos revela um limiar do valor de potássio fora da célula quando a reação r_i passa a exibir uma tendência positiva. Para verificá-la é preciso também estender os modelos no PRISM para definir os *rewards* de transição **plusKout** e **minusKout**, usados na propriedade (7.12), conforme é apresentado na Figura 7.4.

```
rewards "plusKout"
[rr4] true : 1 ;
endrewards

rewards "minusKout"
[r4] true : 1 ;
endrewards
```

Figura 7.4. Extensão dos modelos CTMCs no PRISM para definição dos *rewards* **plusKout** e **minusKout**.

O *reward* **plusKout** atribui o valor 1 toda vez que a reação rr_4 acontece, resultando na mudança de configuração da proteína da bomba de $E_2.K_2$ para $P \sim E_2$, ou seja, ocorre a fosforilação da proteína da bomba e a liberação de dois íons de potássio fora da célula. Por outro lado, o *reward* **minusKout** atribui 1 toda vez que a reação r_4 acontece, resultando na desfosforilação da proteína da bomba e no consumo de dois íons de potássio de fora da célula ($P \sim E_2 \rightarrow E_2.K_2$).

7.5.2 Estudo das Tendências Considerando os Parâmetros Iniciais

Usando as propriedades (7.11) e (7.12) é possível descobrir quais reações exibem tendências positivas ao longo da evolução do sistema e quanto elas passam a exibir essa tendência em termos da concentração de potássio fora da célula (K_{out}^+). O resultado da verificação dessas duas propriedades permite dividir o processo de funcionamento da bomba em três etapas principais, como é apresentado nas figuras 7.5 e 7.6. Note que a Figura 7.5 mostra apenas setas e rótulos para as reações com tendência positiva.

Na primeira etapa (a), há uma tendência inicial da proteína da bomba oscilar entre as configurações $E_1.\text{ATP}$ e $K_2.E_1.\text{ATP}$ dada as tendências positivas das reações rr_6 e r_6 . Se a proteína da bomba se encontra no estado $E_1.\text{ATP}$, existe uma probabilidade superior da reação rr_6 acontecer (ao invés da reação r_1), pois há uma maior concentração de potássio (ao invés de sódio) no interior celular. Isso favorece a ligação de dois íons de potássio no meio intracelular à proteína da bomba, mudando a configuração da bomba para $K_2.E_1.\text{ATP}$. Nessa nova configuração, a bomba tende a voltar para a configuração anterior, devido a tendência positiva da reação r_6 , que apresenta uma constante de taxa bem superior à da reação rr_5 (ver valores f_6 e b_5 na Tabela 6.2).

Assim, para que a proteína da bomba atinja a configuração $\text{Na}_3.E_1 \sim P$, é preciso, além de vencer essa tendência inicial de ocupar as configurações $E_1.\text{ATP}$ e $K_2.E_1.\text{ATP}$, que a hidrólise do ATP ocorra, levando à fosforilação da bomba e à liberação de ADP no meio intracelular. Esse último processo ocorre quando a proteína da bomba está na configuração $\text{Na}_3.E_1.\text{ATP}$ e a reação r_2 acontece, mudando a configuração proteica para $\text{Na}_3.E_1 \sim P$. Note, adicionalmente, que r_2 exibe uma tendência negativa, havendo uma probabilidade maior da reação rr_1 acontecer quando a configuração é $\text{Na}_3.E_1.\text{ATP}$ (segundo Cruz [2011], a fosforilação da bomba é de fato um gargalo conhecido da Na,K-ATPase). Entretanto, uma vez que esse processo acontece, o sistema completa facilmente o ciclo no sentido direto, devido às tendências positivas de r_3 , r_4 , r_5 e r_6 .

A fase (b) inicia quando o K_{out}^+ atinge o valor limite 21, e a reação r_3 passa a ter tendência negativa, e seu par no sentido reverso rr_2 é que passa a exibir uma tendência positiva. Essa reversão é causada pelo aumentando da quantidade de ADP dentro da célula com o funcionamento da bomba no sentido normal, fortalecendo a tendência positiva de rr_2 . Nessa fase a bomba ainda continua funcionando no sentido direto, conforme pode ser visto pelas curvas da Figura 7.6.

A terceira etapa (c) inicia quando o K_{out}^+ atinge o valor 7, e ocorre a reversão das tendências do par de reações r_4 e rr_3 e também do par r_5 e rr_4 . A primeira reversão se deve ao aumentando da quantidade de sódio fora da célula, que favorece a reação rr_3 , e a redução de potássio fora da célula, que enfraquece r_4 . A segunda reversão

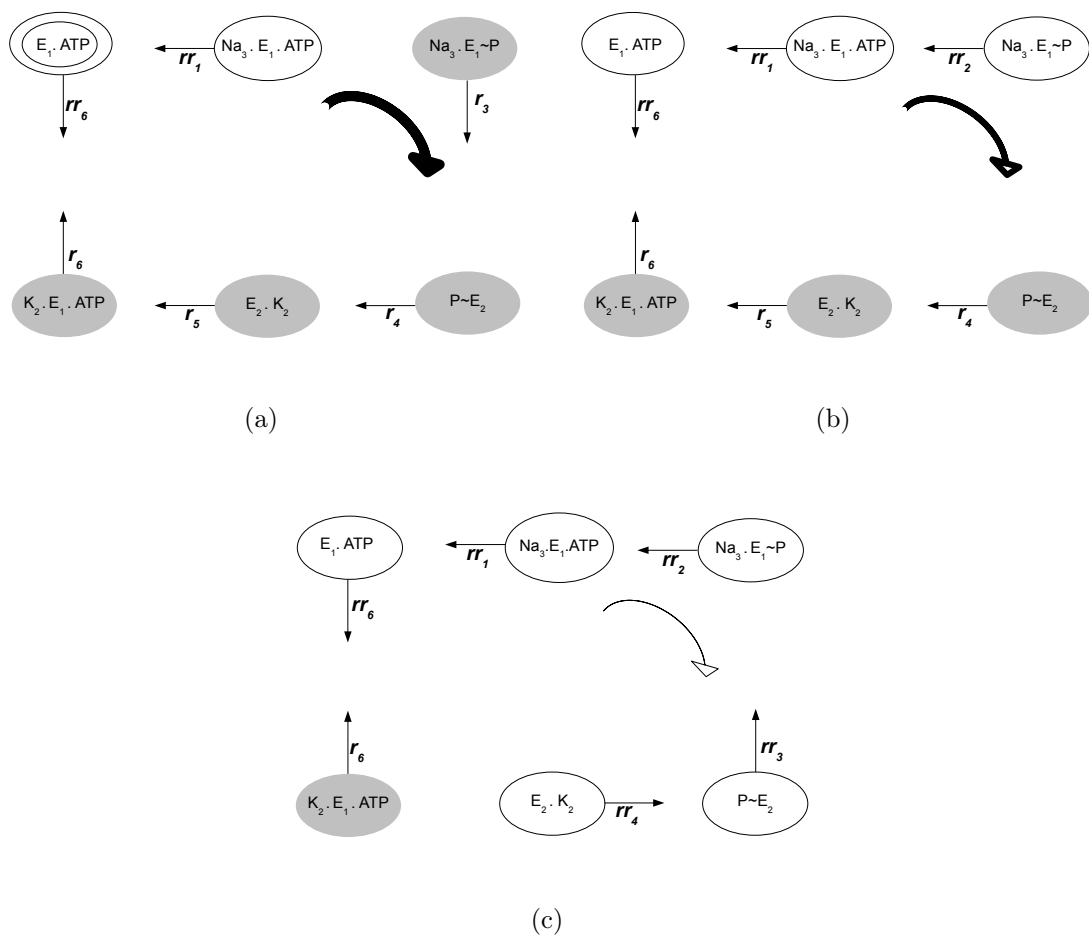


Figura 7.5. Etapas de funcionamento da Na,K-ATPase para o modelo PRISM discreto. As setas são mostradas apenas para as reações que exibem tendência positiva. Configurações da bomba em cinza são aquelas cujas reações no sentido direto são as que exibem tendências positivas, enquanto aquelas na cor branca têm reações no sentido reverso com tendências ascendentes. A grossura da seta central no sentido direto de funcionamento da bomba indica a força da tendência geral de funcionamento da bomba nesse sentido. Assim, quanto mais grossa a seta, maior é a tendência geral da bomba de funcionar no sentido direto. (a) Etapa inicial. O círculo duplo representa a configuração inicial do sistema. (b) Etapa intermediária. (c) Etapa final.

é devida ao acúmulo de P_i dentro da célula, que favorece a reação rr_4 , e a redução de ATP, que desfavorece r_5 . Na etapa (c) é pouco provável, embora não impossível, que a bomba continue seu funcionamento no sentido normal, dado que a única reação no sentido direto que possui tendência positiva é r_6 , tardando, assim, o esgotamento do potássio fora da célula. Há sim uma forte tendência geral para o funcionamento da bomba no sentido reverso nesse momento, o que favorece o retorno da bomba à configuração da etapa intermediária, onde o sistema se concentra na maior parte do

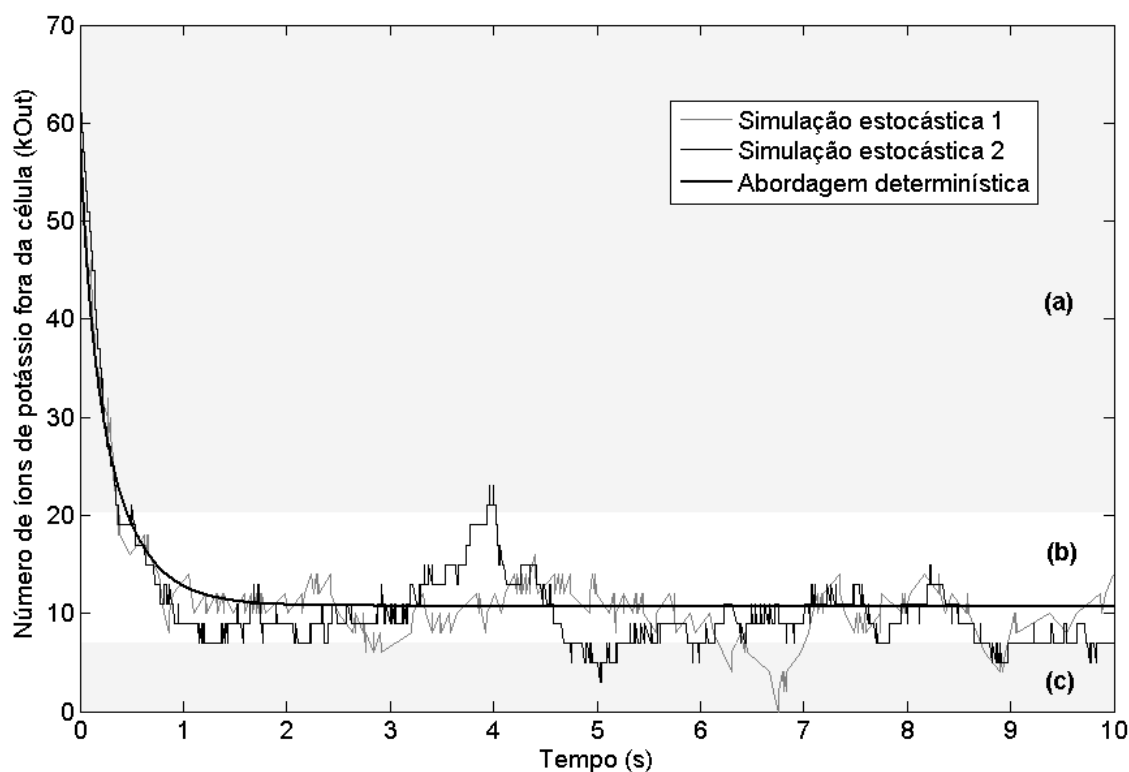


Figura 7.6. Divisão ilustrativa do gráfico que representa a concentração de potássio fora da célula. O gráfico mostra a concentração de potássio fora da célula dada pela abordagem determinística e por dois caminhos gerados por simulação estocástica. As divisões (a), (b) e (c) do gráfico correspondem as respectivas etapas (a), (b) e (c) da Figura 7.5.

tempo. Adicionalmente, a bomba pode funcionar no sentido reverso e voltar a sua configuração inicial. Entretanto, esse processo demora tempos relativamente longos conforme já apresentado. Isso porque é preciso vencer todas as reações r_3 , r_4 , r_5 e r_6 que apresentam tendências positivas na etapa inicial. Para fins ilustrativos, a probabilidade da bomba ir de um estado onde $K_{out}^+ \leq 11$ para outro onde $K_{out}^+ > 21$ (ver caminho resultante da simulação estocástica 2 no gráfico da Figura 7.6), passando da etapa intermediária para a etapa inicial, no tempo menor que 10 segundos é 0,05% (menos que 1%)³.

É possível, dessa forma, sumarizar aqui alguns dos gargalos que levam ao lento funcionamento da bomba com o passar do tempo:

- a baixa concentração de potássio fora da célula, que enfraquece a reação no sentido direto r_4 ;

³ A propriedade utilizada para obter a probabilidade é $P = ? [F \leq 10 \quad kOut > 21 \quad \{kOut \leq 11\}]$

- a alta concentração de sódio fora da célula, que fortalece a reação no sentido reverso rr_3 ;
- o acúmulo de P_i e ADP no meio intracelular que favorecem, respectivamente, as reações no sentido reverso rr_4 e rr_2 ;
- a baixa concentração de ATP que enfraquece a reação no sentido direto r_5 .

As próximas seções mostram como eliminar esses gargalos, tentando, assim, fortalecer o funcionamento da bomba no sentido direto.

7.5.3 Aumentando a Concentração de Potássio fora da Célula

Nesta seção novos estudos do funcionamento da Na,K-ATPase em termos de tendências de reações são realizados considerando um aumento da concentração externa de íons de potássio de $0,01 M$ para $0,1 M$. Isso equivale a aumentar a concentração no modelo discreto de 61 íons para 603 íons.

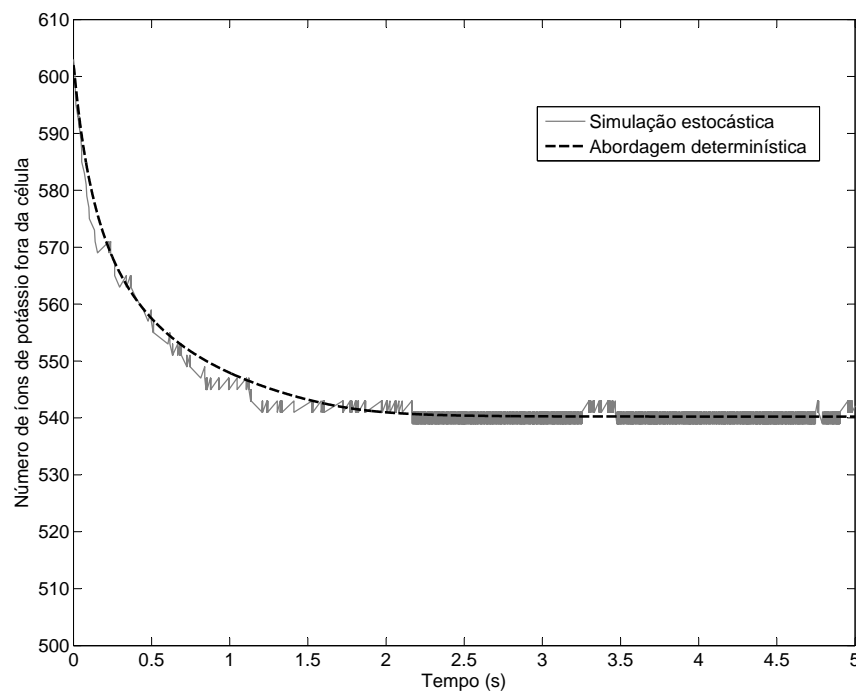


Figura 7.7. Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com alta concentração de potássio fora da célula.

O estudo da seção anterior permitiu identificar que a reação r_4 , inicialmente com tendência positiva, passa a exibir tendência negativa contribuindo para que a tendência

geral da bomba no sentido direto seja reduzida com o passar do tempo. Essa mudança de tendência de r_4 ao longo do funcionamento da Na,K-ATPase no sentido direto se deve à redução de potássio fora da célula e ao fortalecimento da reação que representa seu par no sentido reverso rr_3 . Dessa forma, uma opção para evitar que a reação r_4 exiba uma tendência negativa seria o aumento de potássio fora da célula, como proposto nesta seção. Entretanto, como pode ser visto na Figura 7.7, que apresenta as quantidades de íons de potássio fora da célula no tempo, obtidas através da abordagem determinística e da simulação estocástica utilizando a nova configuração, a bomba não consegue evoluir no sentido direto após a quantidade de K_{out}^+ se aproximar de 541 íons. O objetivo é, então, entender porque essa situação acontece analisando as novas tendências das reações.

Com a verificação das propriedades (7.11) e (7.12) é possível dividir o funcionamento da bomba na nova configuração em 3 etapas, conforme é mostrado nas figuras 7.8 e 7.9. Inicialmente (a), os ciclos no sentido direto da bomba se completam facilmente devido as tendências positivas das reações no sentido direto (r_3 , r_4 , r_5 e r_6). Depois de algum tempo de funcionamento da bomba, mais especificamente quando a quantidade de potássio fora da célula decresce para o valor de 563, a reação r_3 passa a exibir uma tendência negativa, enquanto rr_2 exibe uma tendência positiva. Isso acontece pelo acúmulo de ADP dentro da célula. Nessa nova etapa (b), embora mais difícil, o funcionamento da bomba no sentido normal predomina. Por fim (c), quando a quantidade de potássio fora da célula atinge o valor 549, a reação rr_4 passa a exibir uma tendência positiva, ao mesmo tempo que r_5 passa a ter tendência negativa, devido à redução de ATP dentro da célula e conseqüente acúmulo P_i , o que dificulta ainda mais o funcionamento da bomba no sentido direto. Entretanto, como pode ser visto nas curvas da Figura 7.9, devido as tendências positivas de r_4 e r_6 , a bomba continua seu funcionamento no sentido direto, que é interrompido quando o valor de K_{out}^+ se aproxima de 541. Na verdade, o funcionamento da bomba no sentido normal torna-se impossível, pois neste instante, o ATP dentro da célula (energia necessária para que a bomba funcione normalmente) se esgota. Isso faz com que a reação r_5 , que já possui uma tendência negativa, se torne nula.

A propriedade a seguir retorna o valor da concentração de potássio fora da célula (541) quando o ATP se esgota:

$$KO + 2 * ((R\{plusKout'\} =? [F \ atp=0]) - (R\{minusKout'\} =? [F \ atp=0])) \quad (7.13)$$

A bomba até tenta mover mais dois íons de potássio para fora da célula, como pode

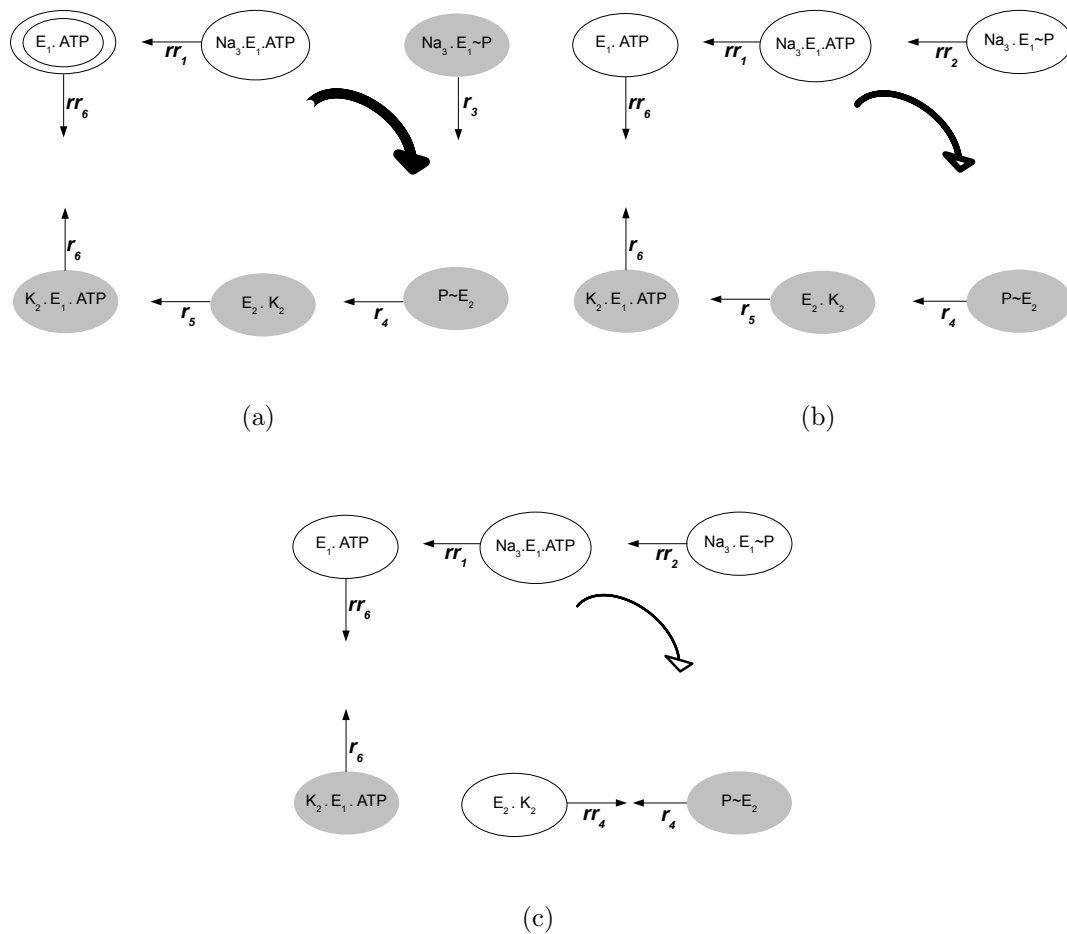


Figura 7.8. Etapas de funcionamento da Na,K-ATPase para o modelo PRISM discreto quando o potássio fora da célula é $0,1 M$. As setas são mostradas apenas para as reações que exibem tendência positiva. Configurações da bomba em cinza são aquelas cujas reações no sentido direto são as que exibem tendências positivas, enquanto aquelas na cor branca têm reações no sentido reverso com tendências ascendentes. A grossura da seta central no sentido direto de funcionamento da bomba indica a força da tendência geral de funcionamento da bomba nesse sentido. Assim, quanto mais grossa a seta, maior é a tendência geral da bomba de funcionar no sentido direto. (a) Etapa inicial. O círculo duplo representa a configuração inicial do sistema. (b) Etapa intermediária. (c) Etapa final.

ser visto no caminho da simulação estocástica apresentado na Figura 7.9 quando K_{out}^+ atinge o valor 539. Entretanto, esses íons ficam ligados à bomba, sem poder atingir o meio intracelular, dado que a configuração $K_2 \cdot E_1 \cdot ATP$ da bomba é impossível de ser alcançada devido à ausência de ATP dentro da célula.

Dessa forma, apesar de eliminar um dos gargalos que dificultavam o funcionamento da bomba no sentido normal (deficiência de potássio no meio extracelular), o funcionamento da mesma no sentido direto é ainda interrompido pelos seguintes motivos:

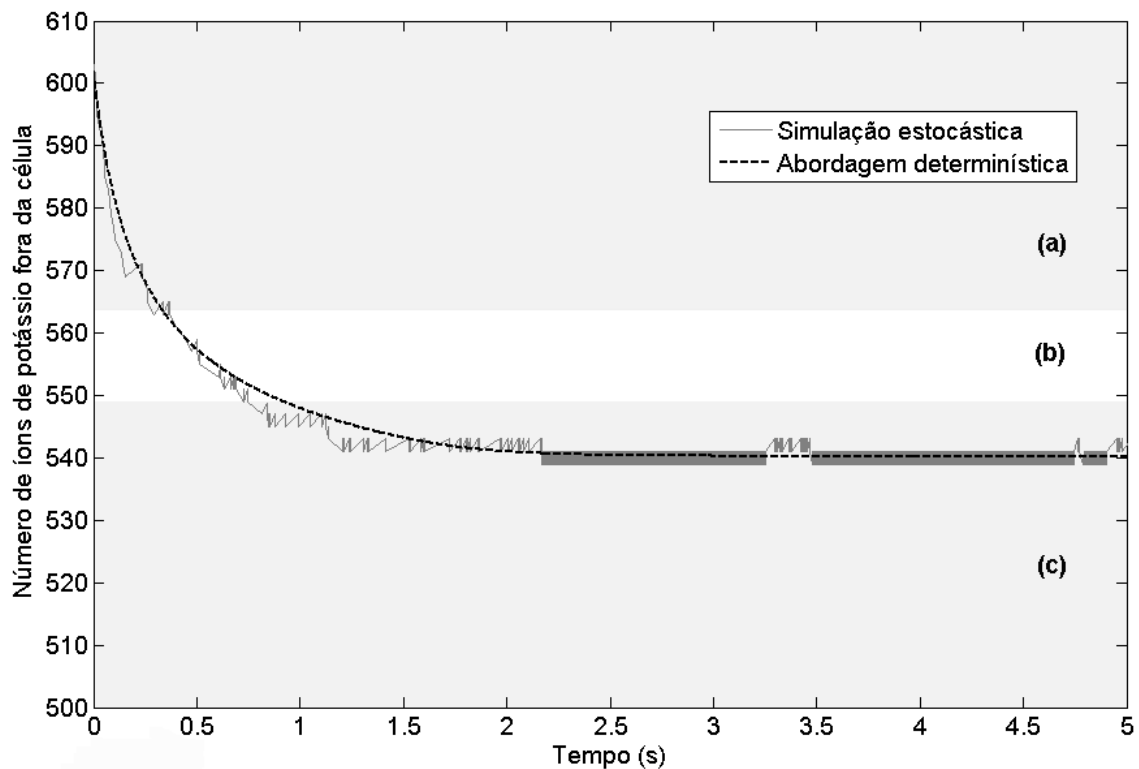


Figura 7.9. Divisão ilustrativa do gráfico que representa a concentração de potássio fora da célula. O gráfico mostra a concentração de potássio fora da célula dada pela abordagem determinística e por um caminho gerado por simulação estocástica. As divisões (a), (b) e (c) do gráfico correspondem as respectivas etapas (a), (b) e (c) da Figura 7.8.

- o acúmulo de P_i e ADP no meio intracelular que favorecem, respectivamente, as reações no sentido reverso rr_4 e rr_2 ;
- o esgotamento de ATP que enfraquece a reação no sentido direto r_5 .

A próxima seção novamente tenta eliminar esses gargalos, a fim de favorecer o funcionamento da bomba no sentido direto.

7.5.4 Adicionando Reações de Síntese de ATP

Finalmente, esta seção considera uma nova configuração em que a concentração inicial de potássio fora da célula ainda permanece elevada ($0,1 M$) e, adicionalmente, o ATP dentro da célula não pode se esgotar, pois uma nova reação $ADP + P_i \xrightarrow{r^{ATP}} ATP$ simula de maneira bastante simplificada a fosforilação oxidativa dentro da célula. A fosforilação oxidativa é um dos processos mais utilizados em muitos dos organismos para sintetizar ATP a partir da energia de oxidação, produzida pela degradação de

carboidratos, gorduras e aminoácidos [Lehninger et al., 2004]. A nova reação impede o esgotamento da energia necessária para o funcionamento da Na,K-ATPase no sentido normal e, conseqüentemente, o acúmulo de ADP e P_i dentro da célula. A Figura 7.10 é uma extensão do modelo discreto do PRISM que adiciona a nova reação.

```

module p
...

[rATP] p>0 -> p : (p'=p-1);

endmodule

module atp
...

[rATP] atp<N -> 1 : (atp'=atp+1);

endmodule

module adp
...

[rATP] adp>0 -> adp : (adp'=adp-1);

endmodule

const double rATP = 1000.0/pow(V*AV,1);

module start
...

[rATP] true -> rATP : true;

endmodule

```

Figura 7.10. Extensão do modelo discreto no PRISM para simular a síntese de ATP dentro da célula.

O estudo da seção anterior mostrou que o esgotamento do ATP era o grande motivo que impedia a bomba de funcionar em seu sentido normal. Com adição das reações que sintetizam ATP, como proposto nesta seção, espera-se que a bomba possa continuar seu funcionamento normalmente no sentido direto. A Figura 7.11 mostra a concentração de potássio fora da célula no decorrer do tempo para o novo modelo. Nota-se, comparando com o gráfico da Figura 7.7, que a bomba consegue avançar no seu funcionamento após a adição das reações de síntese de ATP.

O resultado da verificação da propriedade (7.11), que mostra as tendências para todas as reações do ciclo da bomba, considerando esse novo modelo é sumarizado na Figura 7.12.

Como pode ser observado, nessa nova configuração as reações r_3 e r_5 deixam de exibir uma tendência negativa durante a evolução do sistema, diferente do que acontecia no modelo da seção anterior. Além disso, não existem mais reações cujas tendências

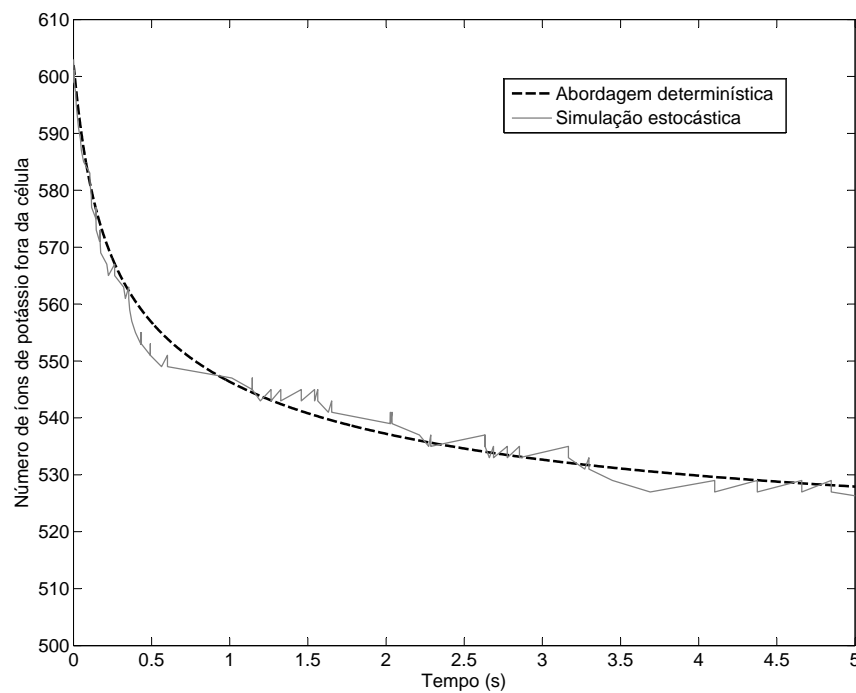


Figura 7.11. Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com reações de síntese de ATP.

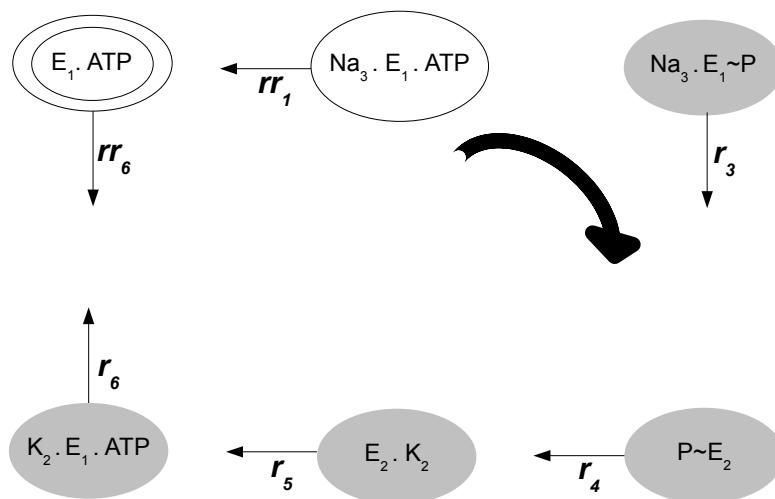


Figura 7.12. Sumarização das tendências das reações no ciclo da Na,K-ATPase apresentado na Figura 6.2 com uma concentração externa de potássio equivalente à $0,1 M$ e com síntese de ATP.

podem ser alteradas ao longo da evolução do sistema. As reações no sentido direto r_3 , r_4 , r_5 , r_6 e as no sentido reverso rr_1 e rr_6 sempre exibem tendências positivas, enquanto

as demais reações do modelo sempre exibem tendências negativas. Assim, a maioria das reações com tendência ascendente favorecem o funcionamento da bomba no sentido direto, justificando o avanço no funcionamento normal da bomba. Entretanto, com o passar do tempo o funcionamento nesse sentido torna-se cada vez mais difícil, como pode ser visto na Figura 7.13, que apresenta as quantidades de íons de potássio fora da célula no tempo obtidas através da abordagem determinística que emprega ODE e através da simulação estocástica (um caminho do modelo). Isso se deve, sobretudo, porque outro substrato (sódio dentro da célula) necessário para o funcionamento da bomba no sentido normal tende a se esgotar. Ao mesmo tempo que a tendência positiva da reação rr_6 se fortalece, devido ao aumento da concentração interna de potássio com o passar do tempo, a tendência da reação r_1 é enfraquecida, dada a redução da concentração interna de sódio.

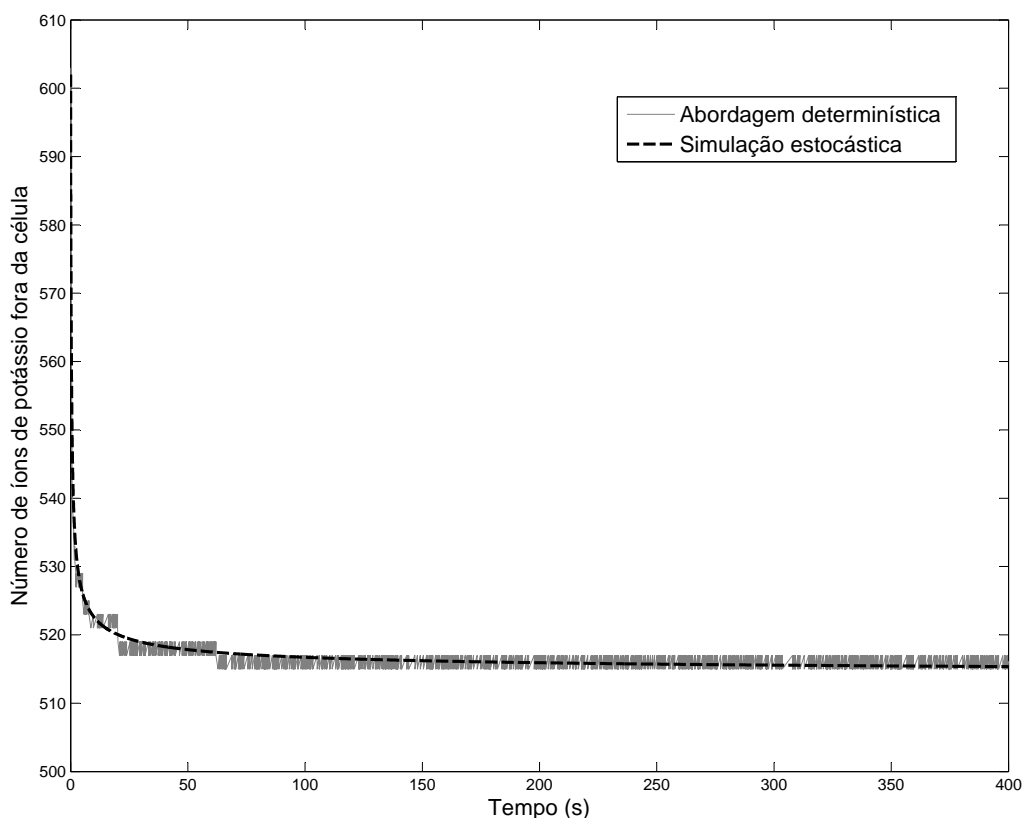


Figura 7.13. Soluções tradicionais para a concentração de potássio fora da célula no modelo da Na,K-ATPase com reações de síntese de ATP, considerando tempos longos.

O esgotamento do sódio no meio intracelular pode ser verificado com a seguinte

propriedade:

$$P \geq 1 [F \text{ 'naInOver'}] \quad (7.14)$$

que para esse novo modelo é **verdade**. Com a propriedade (7.15) é possível, ainda, saber a quantidade de potássio dentro da célula quando o sódio se esgota:

$$KO + 2 * ((R\{\text{'plusKout'}\} =? [F \text{ 'naInOver'}]) - (R\{\text{'minusKout'}\} =? [F \text{ 'naInOver'}])) \quad (7.15)$$

A resposta dessa propriedade é 517, ou seja, quando o número de íons de potássio fora da célula é 517, o sódio dentro da célula se esgota e a reação r_1 , que já exibía uma tendência negativa, passa a ter o valor 0, impossibilitando o funcionamento da bomba no sentido normal de funcionamento.

7.6 Discussões

No estudo realizado neste capítulo foi possível confirmar que a reversibilidade da Na,K-ATPase demora tempos relativamente longos para acontecer em condições normais de funcionamento da bomba. Mesmo o esgotamento de certos substratos, como o potássio fora da célula devido ao funcionamento normal da bomba, é demorado. Isso ocorre porque certas reações passam a ter baixa probabilidade (ou mesmo probabilidade 0) com o funcionamento da bomba no sentido normal, o que é de certa forma esperado, dado que o modelo empregado possui aspectos estáticos, como a não realimentação de certos substratos por outros sistemas. As tendências das reações permitiram identificar essas reações diretamente ligadas à lentidão ou mesmo à interrupção do funcionamento normal da bomba e, conseqüentemente, os substratos cujas concentrações eram insuficientes e impactavam essas reações.

A cada novo possível gargalo identificado, um novo modelo foi proposto de maneira a eliminá-lo, obtivando permitir o funcionamento normal da bomba. Neste trabalho, por questões de escopo, esse processo foi interrompido na identificação da insuficiência de sódio no meio intracelular. Para eliminar esse gargalo, um novo modelo poderia ser proposto de maneira a evitar o esgotamento do sódio no meio intracelular, considerando, por exemplo, o aumento da concentração inicial desse substrato, ou mesmo, a adição de um novo sistema responsável por sua reposição.

Esse tipo de estudo pode ser útil na identificação de novos fármacos. Por exemplo, no caso da bomba ser afetada pela palitoxina, uma potente toxina extraída de corais do

gênero *Palythoa*. Os efeitos dessa toxina incluem contrações de músculos lisos, constrições de vasos sanguíneos, despolarização celular e mesmo a letalidade. Em termos do modelo de Albers-Post a ação da PTX é traduzida pela abertura simultânea das duas portas da bomba transformando-a num canal não-seletivo para cátions monovalentes [Rodrigues et al., 2008]. Dessa forma, dado o comportamento da bomba quando afetada por uma toxina, a identificação dos gargalos no funcionamento da mesma através das tendências das reações poderia ser um primeiro passo para apontar como uma nova droga poderia agir para evitar o comportamento indesejado.

Capítulo 8

Estudos Adicionais

Este capítulo apresenta alguns estudos adicionais sobre a detecção de comportamento periódico em sistemas CTMCs descritos no PRISM e utilizando o algoritmo de verificação aproximado BIOLAB. Além disso, uma abordagem para lidar com a característica multiescalar de certos sistemas biológicos é também introduzida. Ambas metodologias são aplicadas ao sistema de oscilação intracelular de cálcio, que tem papel importante no controle de muitos processos celulares, como secreção e ativação ovular em períodos de fertilização e vários modelos investigando esse comportamento oscilatório têm sido propostos [Borghans et al., 1997]. Os resultados são ainda bem preliminares e estudos mais aprofundados são essenciais para melhor compreendê-los.

8.1 Oscilação Intracelular de Cálcio

A oscilação intracelular de cálcio é observada em uma variedade de tipos celulares em resposta aos estímulos de certas moléculas extracelulares. Como o próprio nome indica, esse sistema é caracterizado por exibir um comportamento oscilatório, ou seja, o conjunto de reações que o modelam resulta em uma flutuação do nível de certos substratos em torno de um dado valor.

A Figura 8.1 ilustra um modelo de oscilação básico proposto no trabalho de Borghans et al. [1997]. Nele há três compartimentos sendo considerados: extracelular, citosólico e o retículo sarcoplasmático (SR). A reação 1f, mostrada na figura, provoca o fluxo de íons de cálcio do meio extracelular para o meio citosólico. Em contrapartida, a reação 1b move íons de cálcio do meio citosólico para o meio extracelular. Nesse modelo é assumido que a concentração extracelular de íons de cálcio é sempre constante. Por sua vez, as reações 2f e 2b da figura permitem o fluxo de íons de cálcio entre o meio citosólico e o SR se o canal de cálcio receptor de IP_3 se encontra ativado. Já a reação 3f permite o movimento de íons de cálcio do SR para o meio citosólico. Finalmente, as

reações 4f e 4b (não são mostradas na Figura 8.1) são adicionadas para representar as mudanças do canal receptor de IP_3 entre os estados ativo e inativo. Esse canal pode mudar seu estado devido à concentração de cálcio no meio citosólico e também pela aumento dos níveis intracelulares de IP_3 , que estimula os receptores do canal. O IP_3 é um mensageiro secundário envolvido na transdução de sinal em células biológicas. Nesse modelo é assumido que a concentração de IP_3 dentro da célula é suficiente para estimular o canal.

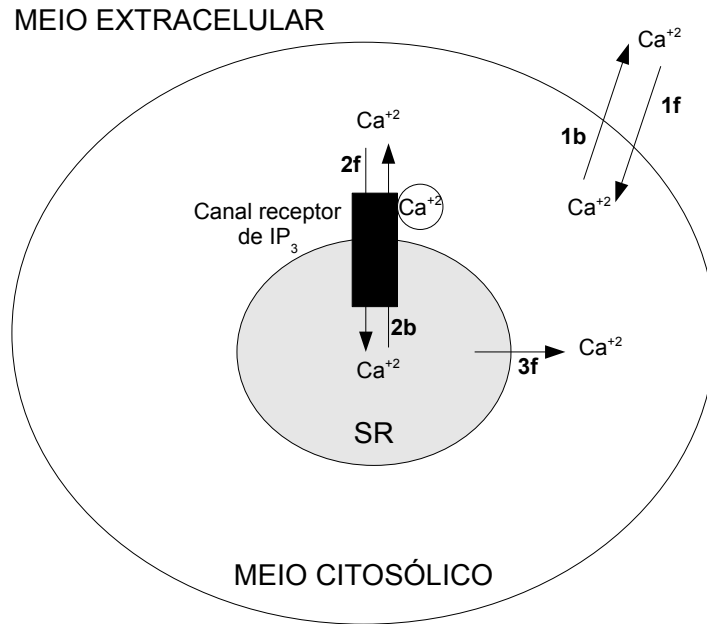


Figura 8.1. Representação esquemática para oscilações do Ca^{2+} . Os fluxos de Ca^{2+} são representados por setas.

A seguir são apresentadas as taxas das reações que regem o comportamento desse sistema, obtidas de [Borghans et al., 1997; Ciocchetta e Guerriero, 2009]:

$$\begin{aligned}
 r_{1f} &= v_0 + (v_1 \cdot \beta) \\
 r_{1b} &= k \cdot [Ca_{\text{cyt}}] \\
 r_{2f} &= V_{M2} \cdot \frac{[Ca_{\text{cyt}}]^2}{(K_2^2 + [Ca_{\text{cyt}}]^2)} \\
 r_{2b} &= \beta \cdot \frac{\frac{a}{d} \cdot [Ca_{\text{cyt}}]^4 \cdot [C_{Ac}]}{(1 + \frac{a}{d} \cdot [Ca_{\text{cyt}}]^4)} \cdot V_{M3} \cdot \frac{[Ca_{\text{SR}}]^2}{K_{\text{CaSR}}^2 + [Ca_{\text{SR}}]^2} \\
 r_{3f} &= k_f \cdot [Ca_{\text{SR}}] \\
 r_{4f} &= k_d \cdot [Ca_{\text{cyt}}]^4 \cdot [C_{Ac}]
 \end{aligned}$$

$$r_{4b} = k_r \cdot [C_{In}]$$

A taxa r_X refere-se a taxa final da reação X descrita anteriormente. Por exemplo, a taxa r_{1f} refere-se a taxa final da reação 1f mostrada na Figura 8.1. $[Ca_{cyt}]$ e $[Ca_{SR}]$ representam, respectivamente, as concentrações molares de cálcio no meio citosólico e no SR, enquanto $[C_{Ac}]$ e $[C_{In}]$ representam, respectivamente, as concentrações molares de canais de cálcio receptores de IP_3 ativos e inativos. As constantes e concentrações iniciais dos substratos envolvidos nas definições das taxas de transições do modelo da oscilação intracelular de cálcio, obtidos de [Borghans et al., 1997], são listadas na Tabela 8.1.

Tabela 8.1. Parâmetros usados nas definições das taxas que regem a cinética do sistema de oscilação intracelular de cálcio.

<i>Parâmetro</i>	<i>Valor</i>	<i>Unidade</i>
$[Ca_{cyt}]$	0,5	μM
$[Ca_{ext}]$	0,1	μM
$[Ca_{SR}]$	1,9	μM
$[C_{Ac}]$	0,1	μM
$[C_{In}]$	0,9	μM
v_0	1	$\mu M \cdot min^{-1}$
v_1	1	$\mu M \cdot min^{-1}$
β	0,5	-
k	10	min^{-1}
k_f	1	min^{-1}
k_r	5	min^{-1}
k_d	5000	$\mu M^{-4} \cdot min^{-1}$
K_2	0,1	μM
$K_{Ca_{SR}}$	0,2	μM
V_{M2}	6,5	$\mu M \cdot min^{-1}$
V_{M3}	50	$\mu M \cdot min^{-1}$
a	40000	$\mu M^{-4} \cdot min^{-1}$
d	100	min^{-1}

8.1.1 Modelo Empregado

A abordagem com níveis discutida na Seção 5.5 do capítulo 5 foi empregada para construir o modelo de oscilação intracelular de cálcio. Essa escolha foi motivada pelo fato de não haver em [Borghans et al., 1997] referências a respeito do volume celular empregado nos experimentos, sendo, portanto, a abordagem baseada na química discreta pouco apropriada, dado que a mesma depende do volume celular. Além disso, a construção de um modelo desse sistema ainda não é possível no BIOLAB porque a

sua linguagem de descrição BIONETGEN não permite funções de taxas cinéticas tão complexas como as do sistema em questão.

A Figura 8.2 mostra parte do modelo construído no PRISM para o sistema de oscilação intracelular de cálcio. A versão completa do modelo encontra-se no Apêndice C.1. Nesse modelo há dois módulos: **ca**, que representa a espécie cálcio, e **ip3** que representa o canal de cálcio receptor de IP_3 . No módulo **ca** existem três variáveis finitas, $caIn$, $caOut$ e $caSR$ que descrevem, respectivamente, os níveis de cálcio no citosol, fora da célula e no SR. Já no módulo **ip3** existem duas variáveis discretas que representam os estados possíveis do canal receptor de IP_3 : rAc , canal ativo, e rIn , canal inativo. Os comandos com rótulo r_x , $x \in \{1f, 1b, 2f, 2b, 3f, 4f, 4b\}$, representam as condições e atualizações necessárias para que a reação x aconteça. Por exemplo, os comandos rotulados com r_{4f} representam os comandos relacionados com a reação $4f$, que é responsável pela mudança do canal receptor de IP_3 de ativo para inativo. A taxa final dessa reação é $(kd/h) * pow(caIn * h, 4) * (rAc * h)$, onde $pow(caIn * h, 4) \equiv (caIn * h)^4$, e é similar àquela apresentada anteriormente ($r_{4f} = k_d \cdot [Ca_{cyt}]^4 \cdot [C_{Ac}]$). As diferenças são devidas à modelagem baseada em níveis, sendo h o tamanho do passo (ver Seção 5.5).

8.1.2 Influência do Tamanho do Passo

Nesta seção será mostrado como o tamanho do passo h no modelo CTMC com níveis pode influenciar no sistema de oscilação intracelular do cálcio. Mais especificamente, quanto menor o tamanho do passo (modelo mais acurado) mais próximo o modelo baseado em níveis se aproxima do comportamento oscilatório e periódico real do sistema. A Figura 8.3 mostra a diferença no comportamento do modelo baseado em níveis para as concentrações de cálcio no meio SR quando se varia o valor de h de $0,1 \mu M$ para $0,0001 \mu M$. Ambas curvas são caminhos gerados pela simulação estocástica na ferramenta PRISM. A curva preta representa o comportamento do sistema dado pela resolução das ODEs para o sistema de oscilação intracelular de cálcio (ver Apêndice D) na ferramenta [MATLAB, 2011].

Nota-se que, apesar das diferenças no comportamento da concentração do cálcio ao longo do tempo devido às variações no tamanho do passo h , os modelos ainda preservam uma característica essencial do sistema: o seu comportamento oscilatório.

8.1.3 Detectando Comportamento Oscilatório

O objetivo desta seção é detectar se o comportamento oscilatório do sistema de oscilação intracelular de cálcio é permanente, ou seja, persiste indefinidamente e não

```

ctmc

const double h;

const int caInI=ceil(0.5/h);
const int caInM=caInI+caOutI+caSRI;
...

const double kd=5000;
...

module ca
caIn : [0..caInM] init caInI;
caOut : [0..caOutM] init caOutI;
caSR  : [0..caSRM] init caSRI;

[r4f] caIn>0 -> pow(caIn*h,4) : true;
...

endmodule

module ip3
rAc : [0..rAcM] init rAcI;
rIn : [0..rInM] init rInI;

[r4f] rIn<rInM & rAc>0 -> rAc*h : (rIn'=rIn+1) & (rAc'=rAc-1);
...

endmodule

module start

[r4f] true -> kd/h : true;
...

endmodule

```

Figura 8.2. Modelo baseado em níveis para a oscilação intracelular de cálcio no PRISM.

pára. A motivação para este estudo é devida ao fato de existirem sistemas biológicos que oscilam apenas temporariamente, como, por exemplo, o *3-way oscillator* [Ballarini et al., 2009; Spieler, 2009].

Spieler [2009] propõe uma maneira simples para verificar se a concentração de um substrato sendo observado μ oscila indefinidamente. Segundo ele, para que isso aconteça é preciso que duas condições sejam satisfeitas:

1. μ não divirja;
2. μ não convirja.

Desde que em modelos CTMCs modelados no PRISM as concentrações dos substratos são finitas, e por isso existe um valor mínimo e máximo para elas, a propriedade (1) é trivialmente satisfeita. Para verificar a propriedade (2), Spieler [2009] propõe o

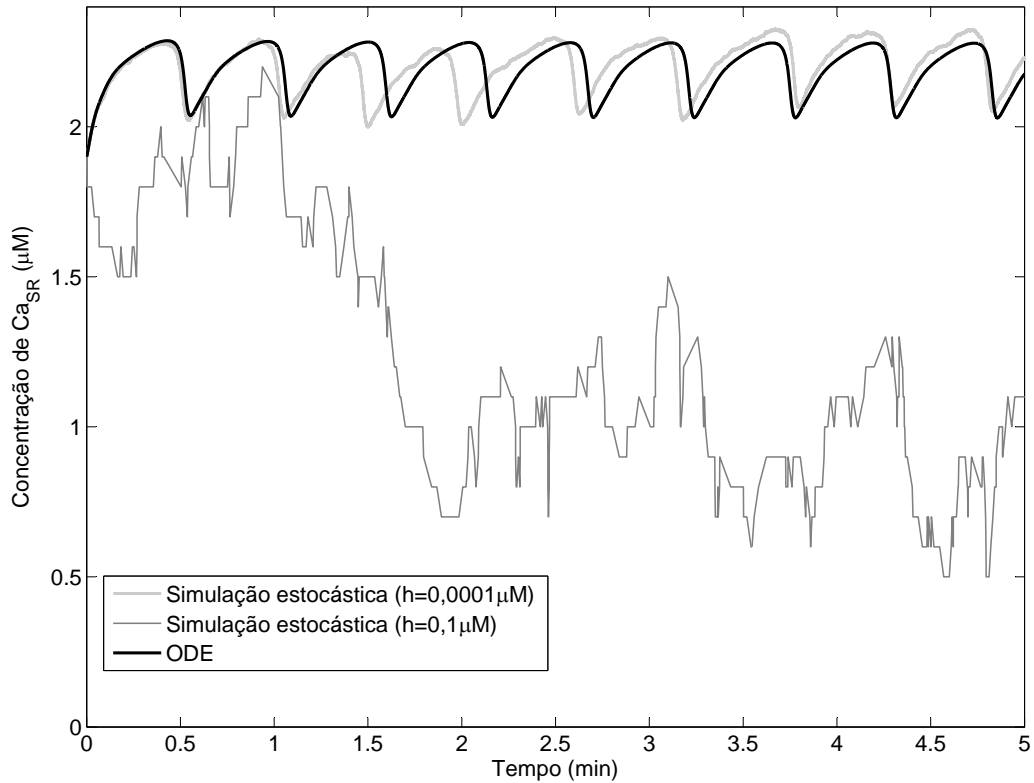


Figura 8.3. Gráfico que representa a concentração de Ca_{SR} ao longo do tempo. As curvas em cinza representam simulações estocásticas do modelo baseado em níveis no PRISM variando o valor de h . Já a curva preta é o resultado de uma simulação determinística.

uso de um marcador $\Delta\mu$ que assume o valor 1 toda vez que há uma reação que modifica a concentração μ do substrato sendo observado e o valor 0 toda vez que há uma reação que não envolve o substrato sendo observado. Logo, para a concentração do substrato sendo observado não convergir, e portanto oscilar, é preciso que a seguinte propriedade seja satisfeita:

$$P_{\geq 1} [GF \Delta\mu = 1] \quad (8.1)$$

ou seja, que sempre o valor μ da concentração do reagente sendo observado mude (não convirja). A Figura 8.4 mostra as mudanças necessárias no modelo do sistema de oscilação intracelular apresentado anteriormente para se detectar a oscilação do substrato Ca_{SR} .

```

module ca

caIn : [0..caInM] init caInI;
caOut : [0..caOutM] init caOutI;
caSR : [0..caSRM] init caSRI;
delta : [0..1] init 0;

[r1f] caIn<caInM -> 1 : (caIn'=caIn+1) & (delta'=0);

[r1b] caIn>0 -> caIn*h2 : (caIn'=caIn-1) & (delta'=0);

[r2f] caIn>0 & caSR<caSRM -> pow(caIn*h2,2)/(pow(caIn*h2,2)+pow(K2,2)) :
      (caIn'=caIn-1) & (caSR'=caSR+1) & (delta'=1);

[r2b] caIn<caInM & caSR>0 & caIn>0 -> (pow(caIn*h2,4)/(1+((a/d)*pow(caIn*h2,4))))*
      (pow(caSR*h2,2)/(pow(Kca,2)+ pow(caSR*h2,2))) :
      (caIn'=caIn+1) & (caSR'=caSR-1) & (delta'=1);

[r3f] caSR>0 & caIn<caInM -> caSR*h2 : (caIn'=caIn+1) & (caSR'=caSR-1) & (delta'=1);

[r4f] caIn>0 -> pow(caIn*h2,4) : (delta'=0);

[r4b] true -> 1 : (delta'=0);

endmodule

```

Figura 8.4. Extensão do modelo baseado em níveis para a oscilação intracelular de cálcio no PRISM a fim de detectar a oscilação do cálcio no SR.

Note que foi necessário adicionar a variável *delta* ao módulo **ca** e ela deve ter seu valor alterado para cada conjunto de comandos identificados por um rótulo. Observando a Figura 8.1 é possível perceber que somente as reações identificadas como *2f*, *2b* e *3f* modificam a concentração de cálcio no SR. Portanto, somente os comandos com esses rótulos atribuem o valor 1 para a variável *delta*. O resultado da verificação da propriedade (8.1), considerando a concentração de cálcio no SR, é **verdadeiro**, tanto para o modelo com $h = 0,1 \mu M$, quanto para o modelo com $h = 0,0001 \mu M$, indicando que, de fato, a oscilação de cálcio no SR é permanente nesses modelos.

8.1.4 Detectando Comportamento Periódico

Usualmente biólogos têm interesse em não só determinar se um sistema oscila, mas também se possui um comportamento periódico, que pode ser definido como um comportamento que mostra algum padrão para a concentração μ de um substrato sendo observado a cada t unidades de tempo e que é repetido de uma maneira sequencial e concatenada sem interrupções [Spieler, 2009]. Entretanto, sistemas estocásticos, por natureza, apresentam ruído, o que resulta em desvios na amplitude e no período do comportamento periódico. Spieler [2009] desenvolve um sistema chamado *BioToPrism* que é capaz de detectar comportamento periódico em modelos CTMC (que lidam com

ruídos) descritos na ferramenta PRISM.

Segundo Spieler [2009] para detectar periodicidade com ruído em um sistema primeiramente é preciso garantir que o sistema oscile indefinidamente entre dois valores definidos como $b_{low} \in \mathbb{R}$ e $b_{high} \in \mathbb{R}$, sendo $b_{low} < b_{high}$. Isso garante que uma trajetória nunca fique em um dos intervalos $I_{low} = (-\infty, b_{low})$, $I_{middle} = [b_{low}, b_{high})$ e $I_{high} = [b_{high}, \infty)$. Mais ainda, é preciso que os três intervalos sejam atravessados infinitamente frequente, não sendo permitida a passagem entre I_{low} e I_{high} sem passar por I_{middle} . Para garantir que essa premissa seja verdade para um modelo é preciso que:

$$|b_{low} - b_{high}| > L \quad (8.2)$$

sendo $L = \max_{s, s' \in S, R(s, s') > 0} |\mu(s) - \mu(s')|$, $\mu(s)$ a concentração da espécie em observação μ no estado s e $\mu(s')$ a concentração da espécie em observação μ no estado s' . Assim a amplitude $A = |b_{low} - b_{high}|$ pode ser considerada como a amplitude mínima das trajetórias do sistema. A Figura 8.5 ilustra os conceitos até então discutidos e introduz alguns predicados:

1. *start*: é verdade somente se um período acabou de iniciar. Caracteriza a primeira transição de I_{low} para I_{middle} desde que o último período se completou;
2. *accept*: é verdade se um sistema iniciou um período, chegou a I_{high} e depois voltou à I_{low} , sempre passando por I_{middle} .

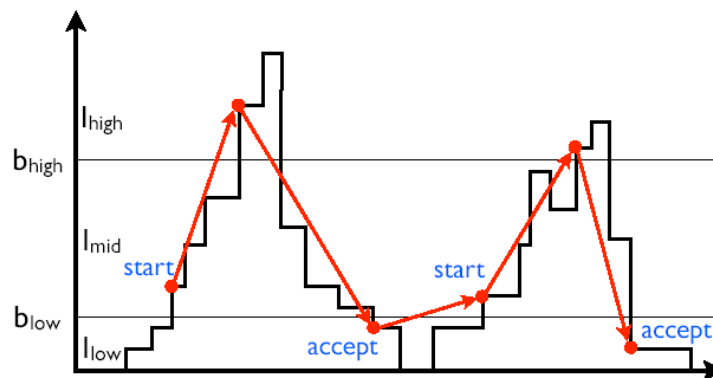


Figura 8.5. Exemplo de caminho estocástico que exhibe periodicidade.

Dadas essas definições basta apenas verificar se a seguinte propriedade CSL é verdadeira para garantir que as trajetórias do sistema biológico exibam comportamento periódico permanente com respeito a concentração μ de um substrato sendo observado:

$$P_{\geq 1} [GF \text{ start}] \quad \& \quad P_{\geq 1} [\text{start} \Rightarrow P_{\geq 1} F \text{ accept}] \quad (8.3)$$

Segundo essa propriedade, para que um período seja detectado, ele deve iniciar (*start*), passando de I_{low} para I_{middle} , e deve terminar (*accept*), passando de I_{middle} para I_{high} , de I_{high} para I_{middle} e, finalmente, de I_{middle} para I_{low} .

Os predicados *start* e *accept* são adicionados automaticamente pela ferramenta *BioToPrism* ao modelo CTMC descrito na ferramenta PRISM utilizando o autômato de estados finitos mostrado na Figura 8.6, onde o estado inicial é q_0 . O sistema pode atingir o estado q_E , que representa um estado de erro. Mais especificamente, esse estado é alcançado toda vez que ocorre uma transição entre I_{low} e I_{high} sem passar por I_{middle} , indicando que $|b_{low} - b_{high}| \leq L$ e contradizendo, portanto, a premissa (8.2).

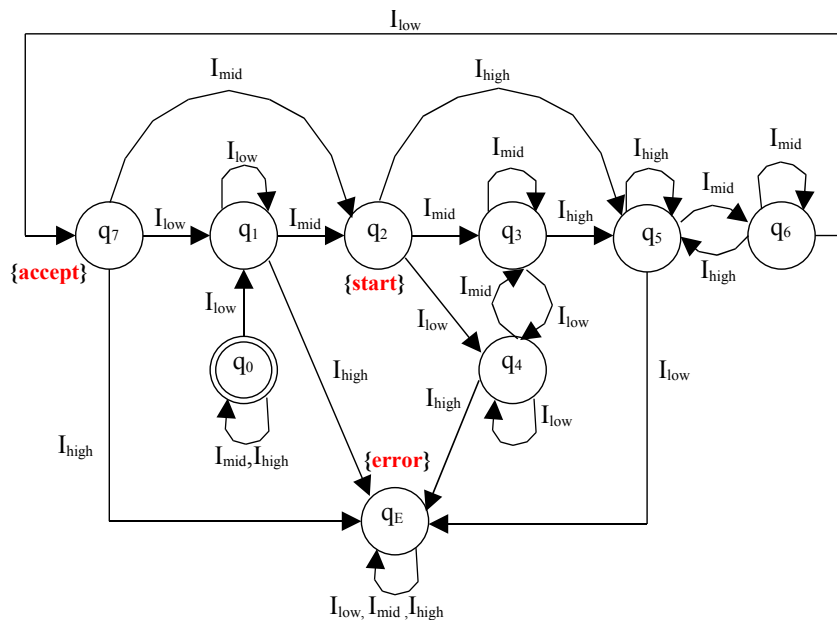


Figura 8.6. Autômato que detecta periodicidade.

Como a detecção da periodicidade proposta por [Spieler, 2009] se baseia na construção do modelo na ferramenta PRISM, ela herda os mesmos problemas dessa ferramenta, como a incapacidade de lidar com grandes modelos. Por exemplo, no modelo baseado em níveis do sistema de oscilação intracelular, quando se emprega um valor de $h = 0,0001 \mu M$ (ver Figura 8.3) o sistema tem um comportamento periódico bem próximo do real, mas não pode ser construído na ferramenta PRISM por causa da explosão de estados. Por outro lado, quando $h = 0,1 \mu M$ (ver Figura 8.3) o sistema pode ser construído, entretanto, o seu comportamento periódico mais se assemelha a uma comportamento oscilatório ou pulsante, sem muita regularidade. Dessa forma, visando

lidar com modelos maiores e mais acurados, o que se propõe é a detecção de periodicidade em sistemas estocásticos empregando a ferramenta BIOLAB, um algoritmo de verificação de modelos aproximado, e utilizando uma abordagem similar à apresentada nesta seção.

8.1.5 Implementação da Detecção de Periodicidade no BIOLAB

Para identificar periodicidade no BIOLAB não é possível empregar a propriedade 8.3 após integrar o autômato da Figura 8.6 ao sistema. Isso porque o BIOLAB é baseado em amostragens de simulações estocásticas finitas (até um tempo t_{end}), enquanto essa propriedade envolve os operadores $G F \phi$ que são somente aplicáveis a caminhos infinitos e que dependem da construção do modelo matemático implícito que representa o sistema. Dessa forma, a proposta, dada a impossibilidade de verificar a probabilidade da periodicidade persistir indefinidamente, é, ao invés disso, saber a probabilidade de haver v ou mais períodos no intervalo de tempo $[0, t_{end}]$.

Para isso, foi adicionado ao BIOLAB o reconhecimento sintático da seguinte nova propriedade ϕ :

$$\phi ::= \{ \mathbf{P}^t x \trianglelefteq v \} \quad b_{low}, b_{high}$$

onde $\trianglelefteq \in \{ \leq, \geq, =, >, < \}$, x é uma variável no modelo de descrição associada ao substrato a ser observado (μ) e $v \in \mathbb{R}_{\geq 0}$. A propriedade ϕ é verdade para um dado caminho finito que descreve a concentração do substrato x no tempo, somente se o número de períodos nesse caminho, considerando a amplitude mínima dada por $|b_{low} - b_{high}|$, dentro de t unidades de tempo, satisfaz $\trianglelefteq v$.

Da mesma maneira como ocorre para as outras propriedade BLTL reconhecidas pelo BIOLAB (ver Seção 3.3.2.2), o algoritmo verifica se ϕ é satisfeito para cada novo caminho estocástico do modelo e utiliza o algoritmo SPRT para decidir se a propriedade PBLTL $\mathcal{P}_{\geq \theta} \phi$ é satisfeita ou não para o modelo ou se é necessário que mais caminhos sejam gerados para que o algoritmo consiga decidir.

Para saber se ϕ é satisfeito para um dado caminho estocástico representado por uma sequência finita $\pi = \mu_0 t_0 \mu_1 t_1 \mu_2 t_2 \dots \mu_n t_n$, onde μ_i é a concentração do substrato x sendo observado no tempo t_i , o autômato da Figura 8.6 é implementado. Ele inicia no estado **q0**, e cada novo valor μ_i leva a uma transição no autômato. Toda vez que o estado **q7** é atingido um novo período é detectado. Após o processamento de todo o caminho, caso o estado **qE** não tenha sido atingido, o número de períodos detectados é comparado com v segundo \trianglelefteq , determinando assim se a propriedade $\{ \mathbf{P}^t x \trianglelefteq v \} \quad b_{low}, b_{high}$ é satisfeita

para o caminho.

Vale ressaltar, como já mencionado, que o sistema de oscilação de cálcio não pode ser descrito no BIONETGEN, dado que não permite funções de taxas cinéticas tão complexas como as desse sistema. Para contornar esse problema e empregar o algoritmo de verificação de modelos BIOLAB, a solução proposta foi a integração dessa ferramenta com o PRISM para gerar simulações estocásticas deles baseadas no método de Gillespie¹. Dessa forma, o modelo é descrito na linguagem PRISM e o algoritmo SPRT utiliza as simulações desse modelo para verificar a propriedade. A Figura 8.7 mostra a mudança no algoritmo de verificação de modelos estatístico BIOLAB de maneira a incorporar essa integração com a ferramenta PRISM.

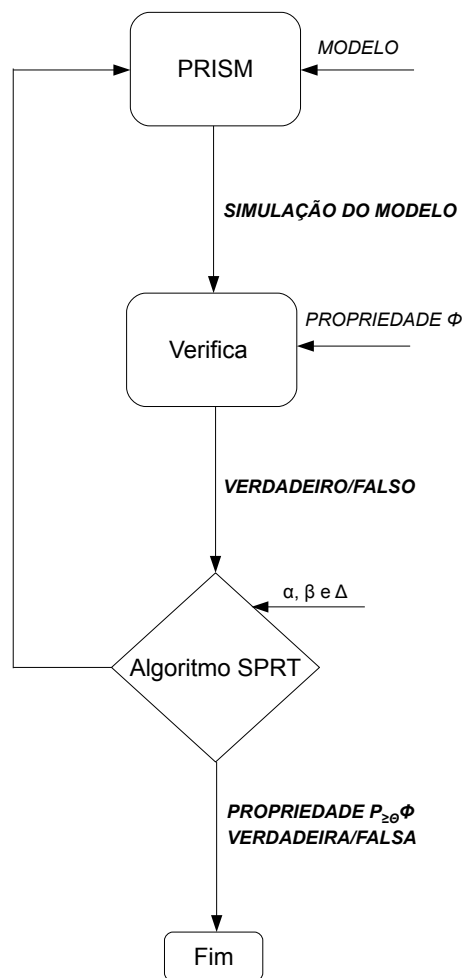


Figura 8.7. Diagrama que mostra arquitetura simplificada da integração do BIOLAB e da ferramenta PRISM

¹O PRISM possibilita também a geração de caminhos estocásticos de modelos descritos na linguagem, além de permitir a verificação exata do modelo

8.1.6 Detectando Períodos no Sistema de Oscilação Intracelular do Cálcio

Esta seção mostra como a metodologia definida anteriormente pode ser aplicada para determinar o comportamento periódico do substrato Ca_{SR} do sistema de oscilação intracelular do cálcio, utilizando o modelo PRISM baseado em níveis da Figura 8.2.

Primeiramente, a amplitude mínima do comportamento periódico é estabelecida, através da indicação dos valores b_{low} e b_{high} . Para o sistema sendo considerado, os respectivos valores utilizados foram $b_{\text{low}} = 2,1 \mu\text{M}$ e $b_{\text{high}} = 2,2 \mu\text{M}$, como mostram as linhas pontilhadas da Figura 8.8, que adicionalmente também mostra a concentração de Ca_{SR} ao longo do tempo, obtida pela simulação estocástica, variando o tamanho do passo h do modelo baseado em níveis.

O tempo t_{end} dos caminhos obtidos, essencial na ferramenta BIOLAB, foi definido como 5 minutos. A propriedade BLTL para detectar pelo menos v períodos no comportamento de Ca_{SR} é

$$\phi_{\text{CaSR}} ::= \{\mathbf{P}^5 \text{CaSR} \geq v\} \quad 2.1, 2.2$$

Além disso, os valores de θ_1 e de θ_2 foram definidos, respectivamente, como 0,95 e 0,98, e α (Erro Tipo 1) e β (Erro Tipo 2), ambos, como 0,01. Isso significa que se a resposta do algoritmo for verdade, a propriedade ϕ_{CaSR} é satisfeita com probabilidade maior ou igual a 98%.

O tamanho do passo h do modelo CTMC com níveis para o sistema de oscilação intracelular de cálcio foi variado de $h_1 = 0,001 \mu\text{M}$ para $h_2 = 0,0001 \mu\text{M}$. É importante notar que o modelo que utiliza h_2 , passo menor, é mais acurado, ou seja, seu comportamento se aproxima mais ao comportamento do sistema biológico real. Quando h_1 é empregado, é possível determinar, com 98% de probabilidade, que a concentração de Ca_{SR} ao longo de 5 minutos tem um comportamento periódico que exhibe, no máximo, 4 períodos. Por outro lado, se h_2 é empregado, a resposta do algoritmo permitiu determinar, com 98% de probabilidade, que a concentração de Ca_{SR} ao longo de 5 minutos pode exibir até 8 períodos. Esse valor é bem próximo do apresentado quando se utiliza a abordagem determinística, mostrada na Figura 8.3 (curva preta), embora a metodologia empregada aqui tenha a vantagem de utilizar um modelo estocástico que pode fornecer diversas visões sobre o comportamento do sistema. Além disso, a abordagem determinística pode não evidenciar o comportamento periódico de certos sistemas biológicos, como no caso do *3-way oscillator* com dopagem [Ballarini et al., 2009]. Mais ainda, vale a pena ressaltar a impossibilidade de construir os modelos no PRISM e utilizar a detecção de periodicidade proposta por Spieler [2009] com os

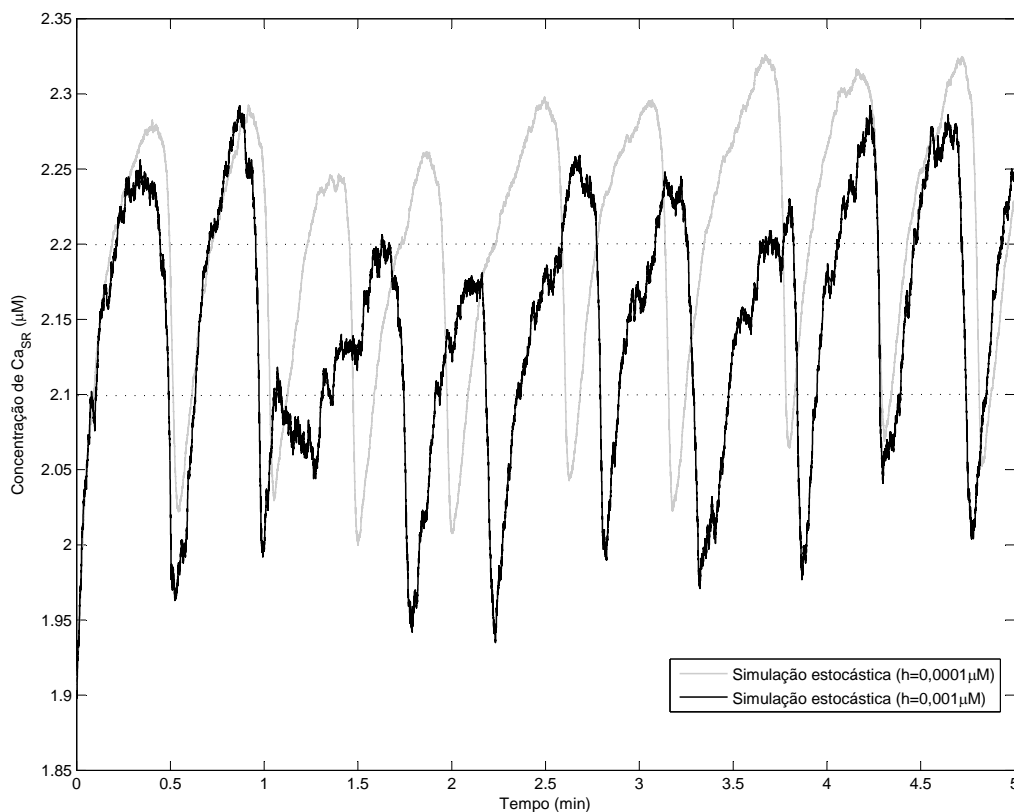


Figura 8.8. Gráfico que representa a concentração de Ca_{SR} ao longo do tempo. A curva preta é o resultado de uma simulação estocástica do modelo baseado em níveis no PRISM usando $h = 0,001 \mu M$. Já a curva cinza é o resultado de uma simulação estocástica do modelo baseado em níveis no PRISM usando $h = 0,0001 \mu M$. As duas retas pontilhadas correspondem aos valores b_{low} e b_{high} que necessitam ser passados ao algoritmo, e delimitam a amplitude mínima $A = |b_{low} - b_{high}|$ do comportamento periódico.

valores h_1 e h_2 empregados.

8.2 Sistemas Multiescalares

Esta seção introduz uma abordagem para lidar com a modelagem de sistemas multiescalares quando se emprega modelos CTMCs baseados em níveis no PRISM. Neste trabalho a multiescalaridade de um sistema é caracterizada pela coexistência de espécies com níveis de concentração de ordens de magnitude muito diferentes. A bomba de sódio e potássio estudada nos capítulos anteriores é um exemplo de um sistema com essa característica, conforme já mencionado, dado que o número de íons de sódio e

potássio dentro e fora da célula são bem maiores (cerca de 10^5 vezes) que o número de bombas encontradas na membrana celular. Essa multiescalaridade é uma característica natural de muitos sistemas biológicos, que podem lidar com órgãos, tecidos, células e moléculas.

Para esses sistemas multiescalares, a utilização de um único tamanho de passo h nos modelos CTMC baseados em níveis pode levar a um grande número de estados para o modelo, caso h seja muito pequeno objetivando representar de maneira acurada a espécie com baixa concentração, ou levar a perda de acurácia na representação da espécie com baixa concentração, caso h seja grande. A proposta sugerida para lidar com tais sistemas multiescalares é a utilização de dois tamanhos de passo diferentes: h_1 , o passo menor, e h_2 , o passo maior. Toda vez que uma reação envolver reagentes do grupo de menor concentração, o tamanho de passo menor (h_1) deve ser usado para obter a taxa final da reação e para atualizar as concentrações dos reagentes e produtos após a reação. Caso contrário, o passo maior h_2 é empregado. Além disso, o número de níveis de concentração de cada substrato é sempre definido inicialmente em termos de h_1 , ou seja, para qualquer substrato X , seja ele pertencente ao grupo de menor concentração ou ao de maior concentração, haverá inicialmente $\frac{[X]}{h_1}$ níveis, onde $[X]$ é a concentração inicial do substrato X dada em molaridade (M ou $\frac{mols}{l}$). Entretanto, caso a reação envolva somente substratos do grupo de maior concentração, a atualização nos níveis de concentração devido à reação se dá em termos de $r = \left\lceil \frac{h_2}{h_1} \right\rceil$, como ilustra a Figura 8.9. O valor r , portanto, representa o número de variações em termos de h_1 que representam uma variação em termos de h_2 .

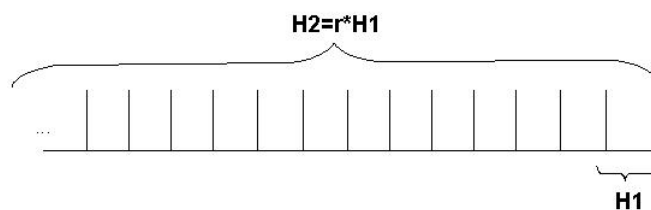


Figura 8.9. Ilustração da divisão da concentração de um substrato em níveis e a sua respectiva atualização quando o tamanho passo da reação da qual participa é h_1 ou h_2 .

Espera-se obter ganhos em escalabilidade em termos de tempo de verificação e de memória com essa abordagem. Por exemplo, para alguns sistema o emprego somente do passo menor h_1 , embora possa gerar uma maior acurácia na representação das espécies de menor concentração, pode impossibilitar a construção do modelo na ferramenta PRISM, por questões de memória, ou mesmo levar a tempos de verificação bem superiores quando se emprega a abordagem sugerida.

Conforme já mencionado, no sistema da bomba de sódio e potássio não há reações que envolvem somente espécies do grupo de maior concentração, e, por isso, os efeitos da utilização da abordagem proposta nesse sistema não são significativos, dado que sempre o passo h_1 é empregado para compor as taxas finais das reações e atualizar as concentrações dos substratos. Assim, para ilustrar os efeitos em escalabilidade obtidos com a metodologia, a composição do sistema de oscilação intracelular de cálcio, apresentado na seção anterior, com o sistema do canal BK que será descrito na subseção seguinte é utilizada. No novo sistema composto, os substratos envolvidos na oscilação do cálcio pertencem ao grupo de espécies de maior concentração, enquanto o canal BK representa a espécie que existe em menor concentração.

8.2.1 Canal BK

O canal BK é um canal iônico que permite o fluxo passivo de íons de potássio de dentro para fora da célula, tornando o potencial da membrana celular mais negativa. Ele é ativado tanto pela concentração intracelular de cálcio ($\text{Ca}_{\text{in}}^{2+}$) quanto pela diferença de potencial na membrana celular e tem sido observado em uma variedade de tipos celulares, sendo crucial em muitos processos fisiológicos como liberação de neurotransmissores, contração do músculo liso e ajustes elétricos das células ciliadas [Rothberg e Magleby, 2000].

O trabalho [Rothberg e Magleby, 2000] propõe o modelo cinético mostrado na Figura 8.10 para representar o mecanismo de funcionamento do canal BK. Nesse modelo, o canal BK pode assumir dois estados: C_i que indica que o canal está fechado, e O_i , que indica que o canal está aberto. O canal somente permite o fluxo de íons de potássio através da célula quando está aberto. Além disso, as mudanças de estado que dependem da concentração interna de cálcio são marcadas com $\xrightarrow{\text{Ca}^{2+}}$, sendo que um estado $C_i(NCa)$ ou $O_i(NCa)$ indica que N íons de cálcio estão ligados à proteína do canal que está no estado C_i ou O_i .

A Tabela 8.2 apresenta as constantes de taxas das transições do modelo apresentado na Figura 8.10. Esses dados também se encontram em Rothberg e Magleby [2000]. A constante de taxa r_{i_j} é referente à transição do estado i (C_i ou O_i) para o estado j (C_j ou O_j) na Figura 8.10.

8.2.2 Modelagem da Integração do Canal BK e do Sistema de Oscilação de Cálcio no PRISM

A Figura 8.11 ilustra o modelo com a integração do sistema de oscilação de cálcio e do canal BK. Tanto o modelo BK quanto o modelo de oscilação de cálcio são depen-

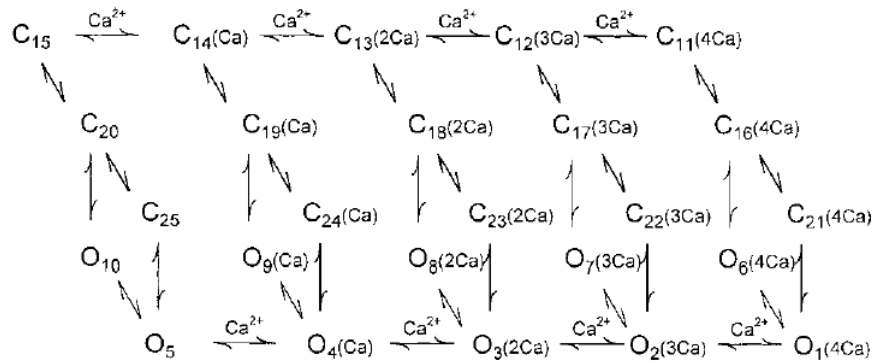


Figura 8.10. Modelo cinético que representa as mudanças de estado de um canal BK.

dentos e modificam a concentração de cálcio no meio citosólico. Neste trabalho, para fins de simplificação, as mudanças na concentração de potássio, realizadas pelo canal BK, serão desconsideradas.

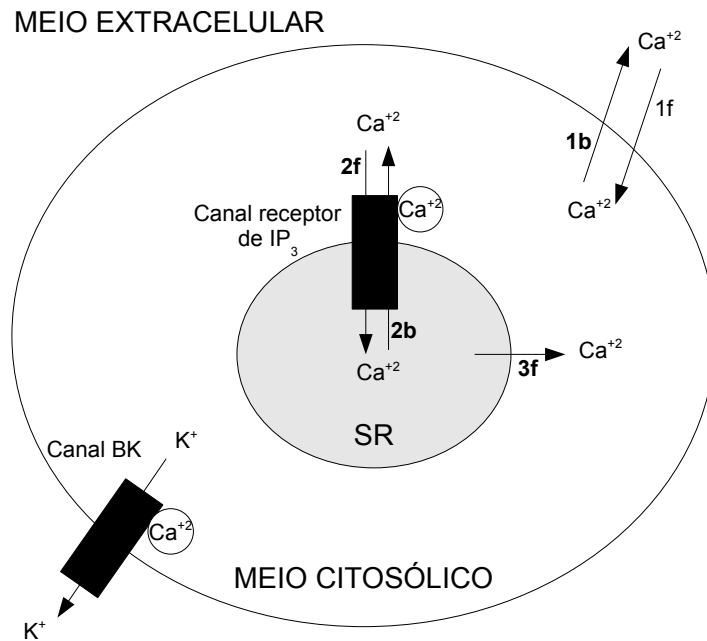


Figura 8.11. Representação esquemática para a integração do sistema de oscilação intracelular de cálcio e o canal BK. Os fluxos de Ca²⁺ são representados por setas.

No modelo integrado, os substratos envolvidos na oscilação do cálcio, como Ca_{out}, Ca_{in}, Ca_{cyt}, o canal de cálcio receptor de IP₃ são considerados como pertencentes ao grupo de maior concentração, enquanto o canal BK representa a espécie que existe em menor concentração.

Tabela 8.2. Constantes usadas nas definições das taxas que regem a cinética do canal BK. A unidade das constantes marcadas com (*) é $\mu M^{-1} min^{-1}$, enquanto a unidade das demais constantes é min^{-1} .

<i>Constante</i>	<i>Valor</i>	<i>Constante</i>	<i>Valor</i>
r_{1_2}	150×60	$r_{2_1}^*$	140×60
r_{1_6}	800×60	r_{6_1}	3300×60
r_{1_21}	480×60	r_{21_1}	23000×60
r_{2_3}	$0,1 \times 60$	$r_{3_2}^*$	$0,1 \times 60$
r_{2_7}	$0,1 \times 60$	r_{7_2}	$0,1 \times 60$
r_{2_22}	1700×60	r_{22_2}	6900×60
r_{3_4}	$0,1 \times 60$	$r_{4_3}^*$	$0,1 \times 60$
r_{3_8}	2600×60	r_{8_3}	3200×60
r_{3_23}	510×60	r_{23_3}	1100×60
r_{4_5}	$0,1 \times 60$	$r_{5_4}^*$	$0,1 \times 60$
r_{4_9}	20000×60	r_{9_4}	3900×60
r_{4_24}	$0,1 \times 60$	r_{24_4}	850×60
r_{5_10}	2500×60	r_{10_5}	3000×60
r_{5_25}	2200×60	r_{25_5}	21000×60
r_{6_16}	2600×60	r_{16_6}	2900×60
r_{7_17}	18000×60	r_{17_7}	45000×60
r_{8_18}	15000×60	r_{18_8}	45000×60
r_{9_19}	1000×60	r_{19_9}	13000×60
r_{10_20}	8700×60	r_{20_10}	1500×60
r_{11_12}	6600×60	$r_{12_11}^*$	40×60
r_{11_16}	25000×60	r_{16_11}	5600×60
r_{12_13}	190×60	$r_{13_12}^*$	130×60
r_{12_17}	$0,1 \times 60$	r_{17_12}	$0,1 \times 60$
r_{13_14}	25000×60	$r_{14_13}^*$	800×60
r_{13_18}	3300×60	r_{18_13}	25000×60
r_{14_15}	5000×60	$r_{15_14}^*$	22×60
r_{14_19}	190×60	r_{19_14}	7500×60
r_{15_20}	$1,9 \times 60$	r_{20_15}	7000×60
r_{16_21}	$0,1 \times 60$	r_{21_16}	$0,1 \times 60$
r_{17_22}	$0,1 \times 60$	r_{22_17}	$0,1 \times 60$
r_{18_23}	$0,1 \times 60$	r_{23_18}	$0,1 \times 60$
r_{19_24}	$0,1 \times 60$	r_{24_19}	14000×60
r_{20_25}	130×60	r_{25_20}	6000×60

Parte do modelo PRISM baseado em níveis que integra o canal BK e o sistema de oscilação de cálcio é mostrado na Figura 8.12. A versão completa do modelo encontra-se no Apêndice C.2. Uma das grandes diferenças em relação ao modelo apresentado na Figura 8.2, que modela somente o sistema de oscilação de cálcio, é a adição do módulo **BKchannel**, para modelar o canal *BK*. Nesse modelo há 25 variáveis locais, sendo

apenas duas delas mostradas ($o1$ e $o2$), que representam os estados do canal, conforme mostrado na Figura 8.10. Inicialmente, todos os canais BK estão no estado O_1 . Além disso, foi preciso adicionar ao módulo **ca** as reações do modelo do canal BK que são dependentes da concentração interna de cálcio. Os novos comandos, com rótulo ri_j , representam as reações do canal BK que levam a mudança do estado i para o estado j , conforme Figura 8.10. Por exemplo, os comandos rotulados com $r2_1$ representam as condições e atualizações necessárias para que o canal BK mude do estado $O_2(3Ca)$ para o estado $O_1(4Ca)$. Quando isso acontece, mais um íon de cálcio que estava dentro da célula se liga à proteína do canal BK, que permanece aberto. A taxa final dessa reação é $r2_1 * (o2 * h1) * (caIn * h1)$, onde $r2_1$ é a constante de taxa obtida após aplicar a Equação (5.7), necessária para construir um modelo CTMC com níveis. A multiplicação pelos valores $o2 * h1$ e $caIn * h1$ se deve a Lei de Ação das Massas. Note que $o2 * h1$ e $caIn * h1$ são os valores das concentrações atuais (ver Equação (5.6)) da proteína do canal em seu estado $O_2(3Ca)$ e de cálcio dentro da célula. Como essa reação envolve um substrato do grupo de menor concentração (o canal BK), o tamanho de passo menor (h_1) é empregado. Diferentemente, para todas as reações do sistema de oscilação intracelular de cálcio, o passo maior h_2 é sempre empregado e as atualizações nas concentrações dos substratos são dadas em termos da constante r , que representa o número de níveis de tamanho h_1 que correspondem à variação de um nível de tamanho h_2 .

8.2.3 Resultados

Alguns experimentos foram realizados a fim de verificar os ganhos computacionais ao se empregar dois tamanhos de passos diferentes no modelo baseado em níveis apresentado anteriormente, composto pelo sistema de oscilação intracelular de cálcio e o pelo canal BK.

Primeiramente, dois tamanhos de passos $h_1 = 0,02 \mu M$ e $h_2 = 0,1 \mu M$ foram empregados, de modo que o tamanho de passo menor ($h_1 = 0,02 \mu M$) é utilizado para obter maior granularidade na representação somente para os substratos de menor concentração (canal BK). O modelo CTMC e a representação MTBDD resultantes na ferramenta PRISM possuem, respectivamente, 900.900 e 4.796.747 estados. Além disso, o tempo para construir o modelo foi de aproximadamente 8 minutos e 6 segundos.

Posteriormente, empregou-se somente um tamanho de passo $h_1 = h_2 = 0,02$. O modelo CTMC e a representação MTBDD resultantes na ferramenta PRISM, neste caso, possuem, respectivamente, 20.241.900 e 108.140.131 estados. Além disso, o tempo para construir o modelo foi de aproximadamente 11 minutos e 9 segundos. Note, portanto, que embora os tempos de construção não sejam significativamente diferentes,

```

ctmc

const double h2;
const double h1;
const int r=ceil(h2/h1)

const int caInI=ceil(0.5/h1);
const int caInM=caInI+caOutI+caSRI;
const double kd=5000;
const double r2_1=140*60/h1;
...

module ca
caIn : [0..caInM] init caInI;
caOut : [0..caOutM] init caOutI;
caSR : [0..caSRM] init caSRI;

//Calcium oscillation reactions
[r4f] caIn>0 -> pow(caIn*h1,4) : true;
...

//BK reactions
[r2_1] caIn>0 -> caIn*h1 : (caIn'=caIn-1);
...

endmodule

module ip3
rAc : [0..rAcM] init rAcI;
rIn : [0..rInM] init rInI;

[r4f] rIn<rInM & rAc>0 -> rAc*h1 : (rIn'=min(rIn+r,rInM)) & (rAc'=max(rAc-r,0));
...
endmodule

const int NP=1;
module BKchannel
o1: [0..NP] init NP;
o2: [0..NP] init 0;
...

[r2_1] o2>0 & o1<NP -> o2*h1 : (o2'=o2-1) & (o1'=o1+1);
...
endmodule

module start
[r4f] true -> kd/h1 : true;
...

[r2_1] true -> r2_1 : true;
...

endmodule

```

Figura 8.12. Modelo PRISM baseado em níveis que representa a integração do sistema de oscilação intracelular de cálcio e o canal BK.

há uma redução significativa no tamanho do modelo, quando a abordagem com dois tamanhos de passo diferentes é empregada (mais de 20 vezes).

Assim, a abordagem proposta, apesar de ser mais abstrata, uma vez que a granu-

laridade na representação do grupo com concentrações maiores é reduzida, consegue produzir modelos menores. Dessa forma, dada as limitações de recursos de memória e por ser mais escalável em termos desse recurso, conforme apresentado, a abordagem com dois tamanhos de passos diferentes pode ser preferida.

Capítulo 9

Conclusões e Trabalhos Futuros

Este trabalho, conforme apresentado, foca na análise e verificação de sistemas de transporte de íons através das membranas celulares utilizando a técnica de verificação probabilística de modelos.

Primeiramente, foram construídos três modelos distintos para a Na,K-ATPase enfatizando as vantagens de cada modelo, bem com as deficiências. O modelo PRISM baseado na química discreta trata cada molécula ou íon de maneira individual. Entretanto, tem uma grande restrição de não poder lidar com volumes reais, dado a intratabilidade do modelo. A abstração alternativa apresentada, o modelo PRISM baseado em níveis, sofre como o problema da discretização, ao tratar as concentrações dos substratos somente em passos de tamanho h . Já o modelo do BIOLAB também sofre com o problema da discretização, porém no tempo, e também não consegue verificar propriedades que não considerem um intervalo de tempo, dado que se baseia em simulações estocásticas finitas. Além disso, apresenta certas restrições para a descrição dos modelos, dado que não permite a definição de leis cinéticas mais gerais, e possui uma linguagem de especificação de propriedades bem mais limitada que a da ferramenta PRISM, apesar de ser mais intuitiva para o contexto biológico.

Apontar o melhor modelo a ser empregado não é uma tarefa trivial, dado que, como já mencionado, cada um apresenta certas vantagens e também restrições. Assim tal tarefa depende dos objetivos e do contexto considerados. Por exemplo, se as análises focam em propriedades em que existe um intervalo de tempo a ser considerado, e não é essencial saber a probabilidade exata da ocorrência do evento, o mais interessante é o uso dos modelos no BIOLAB. Nesse caso, os tempos de verificação são geralmente inferiores aos obtidos com o modelo PRISM baseado na química discreta e é possível trabalhar com o volume real celular, sem a necessidade de reduzi-lo ou mesmo empregar modelos PRISM baseados em níveis. Entretanto, se o problema envolve um número reduzido de representantes dos substratos, o custo fixo do BIOLAB pode não compensar

o seu emprego, conforme apresentado na Seção 3.3.3, sendo mais interessante o uso do PRISM nesse caso. Finalmente, caso a propriedade de interesse não envolva intervalos de tempo, ou seja, deseja-se saber se um certo evento ocorre, independente de quando, ou mesmo se o sistema é caracterizado por oscilações permanentes, os modelos PRISM são preferíveis. Normalmente, nesse caso, algum tipo de abstração se faz necessária para lidar com o grande número de representantes dos substratos: ou a redução do volume celular, ou o emprego da abordagem baseada em níveis. A comparação entre essas duas formas de abstração se apresenta como uma questão complexa que exige o amparo de experimentos biológicos, a fim de verificar qual das duas abstrações fornece resultados para a verificação das propriedades mais próximas dos resultados reais.

Apesar das limitações de cada modelo, cada um deles pode ser usado conjuntamente com as abordagens tradicionais para uma melhor compreensão do funcionamento da Na,K-ATPase. Isso porque o objetivo do emprego da técnica de verificação de modelos não é uma reprodução fiel do comportamento real dos sistemas, mas sim apoiar os biólogos sugerindo aspectos interessantes que devam ser verificados experimentalmente. Mesmo porque as abordagens tradicionais também apresentam restrições: a simulação estocástica, por exemplo, também sofre com o problema da discretização do tempo e com a impossibilidade de verificar eventos que ocorrem indefinidamente; a abordagem determinística, por não ser estocástica, não consegue apresentar as diversas visões possíveis de um modelo, podendo não identificar certas características do sistema.

Este trabalho apresenta também um estudo de várias questões sobre o comportamento da bomba de sódio e potássio, como sua reversibilidade e o esgotamento do potássio fora da célula, utilizando a técnica de verificação de modelos. Foi possível verificar que esses eventos são raros, ou seja, ocorrem com baixa probabilidade em curtos períodos de tempo e, por isso mesmo, podem não ser facilmente identificados pelas abordagens tradicionais. Entretanto, apesar de raros, esses eventos podem ter grande interesse biológico, caso acarretem graves problemas ao indivíduo ou mesmo sua morte. Por exemplo, a redução do potássio no meio extracelular abaixo de certos limiares caracteriza a doença conhecida como hipocalemia, que pode, em casos extremos, causar a morte do indivíduo. Assim, esse estudo pode ser estendido para reproduzir situações reais, caracterizadas não necessariamente pelo esgotamento do substrato, mas pela sua redução à limiares críticos.

Adicionalmente, o conceito de tendência de reação foi introduzido e aplicado no estudo da dinâmica da Na,K-ATPase, identificando as causas que levam a bomba a funcionar lentamente no sentido normal com o passar do tempo. Identificados os gargalos, novos modelos foram propostos de maneira a eliminá-lo, objetivando favorecer o funcionamento da bomba. A identificação dos pontos críticos (ou gargalos) no funcio-

namento de um sistema se apresenta como um passo potencialmente útil na criação de novas drogas. Por exemplo, se uma toxina bloqueia o funcionamento de um canal, a identificação dos gargalos no funcionamento do sistema poderia ser um primeiro passo a fim de eliminar o comportamento indesejado. Trabalhos futuros incluem ainda a aplicação do conceito de tendência em outros sistemas biológicos cíclicos, como outras bombas, além da utilização dessa metodologia no estudo de sistemas mais complexos, objetivando identificar reações com tendências muito negativas (variando o limiar ξ) e simplificar o conjunto de reações desses sistemas desconsiderando tais reações.

Finalmente o trabalho apresenta também alguns estudos iniciais sobre a detecção de comportamentos periódicos usando o algoritmo aproximado de verificação de modelos BIOLAB e sobre a modelagem de sistemas multiescalares. O sistema de oscilação intracelular de cálcio foi empregado em ambos estudos, considerando uma modelagem CTMC baseada em níveis. O emprego do BIOLAB se apresenta como uma abordagem mais escalável na verificação de comportamentos periódicos e que, por isso, consegue lidar com sistemas que não poderiam ser construídos na ferramenta PRISM. Trabalhos futuros visam incorporar a estimação do valor da amplitude mínima de oscilação (b_{low} e b_{high}) pelo menos para certos tipos de modelos periódicos e a quantificação do tamanho dos períodos. A metodologia de modelagem para lidar com sistemas multiescalares propõe uma maior granularidade na representação apenas dos substratos de menor concentração, enquanto que os substratos de maior concentração mantêm uma representação menos acurada. Trabalhos futuros são importantes para validar a metodologia introduzida através da sua aplicação em outros sistemas biológicos e também do estudo dos efeitos na qualidade das soluções com a redução da granularidade dos substratos de maior concentração, quando comparado com um modelo que emprega a mesma granularidade para os dois grupos de substratos. Validações com experimentos biológicos também seriam interessantes nesse sentido.

Apêndice A

Modelos da Na,K-ATPase

A.1 Modelo Baseado na Química Discreta no PRISM

```
ctmc

// constants
const double exp;
const double AV=6.022*pow(10.0,23);
const double V=pow(10.0,-exp);

const int NI=ceil(0.022*AV*V); // maximal number of Na ions inside the cell
const int NO=ceil(0.14*AV*V); // maximal number of Na ions outside the cell
const int KI=ceil(0.127*AV*V); // maximal number of K ions inside the cell
const int KO=ceil(0.01*AV*V); // maximal number of K ions outside the cell
const int ATPI=ceil(0.005*AV*V); // number of ATP molecules inside the cell
const int ADP=ceil(0.00006*AV*V); // initial number of ADP molecules inside the cell
const int Pi=ceil(0.00495*AV*V); // initial number of Phosphate ions inside the cell
const int naFlow=3; // Na ions that go through membrane each time
const int kFlow=2; // K ions that go through membrane each time
const int N=ATPI+min(ADP,Pi)+NP; // upper bound limit of ATP molecules
const int NP=1;

module na
naIn : [0..(NI+NO)] init NI; //Number of Na inside cell
naOut : [0..NO+NI] init NO; //Number of Na outside cell

[r1] naIn>=naFlow -> pow(naIn,3) : (naIn'=naIn-naFlow);
[rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);

[r3] naOut<=NO+NI-naFlow -> 1 : (naOut'=naOut+naFlow);
[rr3] naOut>=naFlow -> pow(naOut,3) : (naOut'=naOut-naFlow);

endmodule
```

```

module k
kOut : [0..K0+KI] init K0; // number of K molecules outside the cell
kIn : [0..KI+K0] init KI; // number of K molecules inside the cell

[r6] kIn<=KI+K0-kFlow -> 1 : (kIn'=kIn+kFlow);
[rr6] kIn>=kFlow -> pow(kIn,2) : (kIn'=kIn-kFlow);

[r4] kOut>=kFlow -> pow(kOut,2) : (kOut'=kOut-kFlow);
[rr4] kOut<=(K0+KI-kFlow) -> 1 : (kOut'=kOut+kFlow);

endmodule

module p
p : [0..(Pi+ATPI+NP)] init Pi;

[r4] p<(Pi+ATPI+NP)-> 1 : (p'=p+1);
[rr4] p>0 -> p : (p'=p-1);

endmodule

module atp
atp : [0..N] init ATPI;

[r5] atp>0 -> atp : (atp'=atp-1);
[rr5] atp<N -> 1 : (atp'=atp+1);

endmodule

module adp
adp : [0..(ADP+ATPI+NP)] init ADP;

[r2] adp<(ADP+ATPI+NP) -> 1 : (adp'=adp+1);
[rr2] adp>0 -> adp : (adp'=adp-1);

endmodule

module pump
E1ATP : [0..1] init 1;
E1ATPNa : [0..1] init 0;
E1PNa : [0..1] init 0;
E2P : [0..1] init 0;
E2K : [0..1] init 0;
E1ATPK : [0..1] init 0;

//reaction1: 3 Na ions bind to pump
[r1] E1ATP=1 & E1ATPNa=0 -> 1 : (E1ATP'=0) & (E1ATPNa'=1);
[rr1] E1ATP=0 & E1ATPNa=1 -> 1 : (E1ATP'=1) & (E1ATPNa'=0);

//reaction2: It happens the fosforilization of the ATP molecule
[r2] E1ATPNa=1 & E1PNa=0 -> 1 : (E1ATPNa'=0) & (E1PNa'=1);
[rr2] E1ATPNa=0 & E1PNa=1 -> 1 : (E1ATPNa'=1) & (E1PNa'=0);

//reaction3: Pump changes its conformation and releases 3 Na ions outside the cell
[r3] E1PNa=1 & E2P=0 -> 1 : (E1PNa'=0) & (E2P'=1);

```

```

[rr3] E1PNa=0 & E2P=1 -> 1 : (E1PNa'=1) & (E2P'=0);

//reaction4: 2 K ions bind to pump
[r4] E2P=1 & E2K=0 -> 1 : (E2P'=0) & (E2K'=1);
[rr4] E2P=0 & E2K=1 -> 1 : (E2P'=1) & (E2K'=0);

//reaction5: The channel changes its conformation
[r5] E2K=1 & E1ATPK=0 -> 1 : (E2K'=0) & (E1ATPK'=1);
[rr5] E2K=0 & E1ATPK=1 -> 1 : (E2K'=1) & (E1ATPK'=0);

//reaction6: Pump release 2 K ions inside the cell
[r6] E1ATPK=1 & E1ATP=0 -> 1 : (E1ATPK'=0) & (E1ATP'=1);
[rr6] E1ATPK=0 & E1ATP=1 -> 1 : (E1ATPK'=1) & (E1ATP'=0);

endmodule

// base rates
const double r1rate = 2.5*pow(10.0,11)/(pow((V*AV),3));
const double r2rate = 10000;
const double r3rate = 360;
const double r4rate = 1.5*pow(10.0,7)/(pow((V*AV),2));
const double r5rate = 2*pow(10.0,6)/(pow((V*AV),1));
const double r6rate = 1.15*pow(10.0,4);

const double rr1rate = pow(10.0,5);
const double rr2rate = 1*pow(10.0,5)/(pow((V*AV),1));
const double rr3rate = 8.5*pow(10.0,3)/(pow((V*AV),3));
const double rr4rate = 2*pow(10.0,5)/(pow((V*AV),1));
const double rr5rate = 30;
const double rr6rate = 6*pow(10.0,8)/(pow((V*AV),2));

// module representing the base rates of reactions
module base_rates

[r1] true -> r1rate : true;
[rr1] true -> rr1rate : true;

[r2] true -> r2rate : true;
[rr2] true -> rr2rate : true;

[r3] true -> r3rate : true;
[rr3] true -> rr3rate : true;

[r4] true -> r4rate : true;
[rr4] true -> rr4rate : true;

[r5] true -> r5rate : true;
[rr5] true -> rr5rate : true;

[r6] true -> r6rate : true;
[rr6] true -> rr6rate : true;

endmodule

```

A.2 Modelo Baseado na Química Discreta no PRISM e com Número de Bombas Modelado usando a Abordagem Individual

```

ctmc

// constants
const double expoente;
const double AV=6.022*pow(10.0,23);
const double V=pow(10.0,-expoente);

const int NI=ceil(0.022*AV*V); // maximal number of Na ions inside the cell
const int NO=ceil(0.14*AV*V); // maximal number of Na ions outside the cell
const int KI=ceil(0.127*AV*V); // maximal number of K ions inside the cell
const int KO=ceil(0.01*AV*V); // maximal number of K ions outside the cell
const int ATPI=ceil(0.005*AV*V); // number of ATP molecules inside the cell
const int ADP=ceil(0.00006*AV*V); // initial number of ADP molecules inside the cell
const int Pi=ceil(0.00495*AV*V); // initial number of Phosphate ions inside the cell
const int naFlow=3; // Na ions that go through membrane each time
const int kFlow=2; // K ions that go through membrane each time
const int NP=2;
const int N=ATPI+min(Pi,ADP)+NP; // upper bound to ATP, ADP and ATP molecules inside the cell

module na
naIn : [0..(NI+NO)] init NI; //Number of Na inside cell
naOut : [0..NO+NI] init NO; //Number of Na outside cell

[r1] naIn>=naFlow -> pow(naIn,3) : (naIn'=naIn-naFlow);
[rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);

[r3] naOut<=NO+NI-naFlow -> 1 : (naOut'=naOut+naFlow);
[rr3] naOut>=naFlow -> pow(naOut,3) : (naOut'=naOut-naFlow);

endmodule

module k
kOut : [0..KO+KI] init KO; // number of K molecules outside the cell
kIn : [0..KI+KO] init KI; // number of K molecules inside the cell

[r6] kIn<=KI+KO-kFlow -> 1 : (kIn'=kIn+kFlow);
[rr6] kIn>=kFlow -> pow(kIn,2) : (kIn'=kIn-kFlow);

[r4] kOut>=kFlow -> pow(kOut,2) : (kOut'=kOut-kFlow);
[rr4] kOut<=(KO+KI-kFlow) -> 1 : (kOut'=kOut+kFlow);

endmodule

module p
p : [0..(Pi+ATPI+NP)] init Pi;

[r4] p<(Pi+ATPI+NP) -> 1 : (p'=p+1);
[rr4] p>0 -> p : (p'=p-1);

```

```

endmodule

module atp
atp : [0..N] init ATPI;

[r5] atp>0 -> atp : (atp'=atp-1);
[rr5] atp<N -> 1 : (atp'=atp+1);

endmodule

module adp
adp : [0..(ADP+ATPI+NP)] init ADP;

[r2] adp<(ADP+ATPI+NP) -> 1 : (adp'=adp+1);
[rr2] adp>0 -> adp : (adp'=adp-1);

endmodule

module pump
E1ATP : [0..1] init 1;
E1ATPNa : [0..1] init 0;
E1PNa : [0..1] init 0;
E2P : [0..1] init 0;
E2K : [0..1] init 0;
E1ATPK : [0..1] init 0;

//reaction1: 3 Na ions bind to pump
[r1] E1ATP=1 & E1ATPNa=0 -> 1 : (E1ATP'=0) & (E1ATPNa'=1);
[rr1] E1ATP=0 & E1ATPNa=1 -> 1 : (E1ATP'=1) & (E1ATPNa'=0);

//reaction2: It happens the fosforilization of the ATP molecule
[r2] E1ATPNa=1 & E1PNa=0 -> 1 : (E1ATPNa'=0) & (E1PNa'=1);
[rr2] E1ATPNa=0 & E1PNa=1 -> 1 : (E1ATPNa'=1) & (E1PNa'=0);

//reaction3: Pump changes its conformation and releases 3 Na ions outside the cell
[r3] E1PNa=1 & E2P=0 -> 1 : (E1PNa'=0) & (E2P'=1);
[rr3] E1PNa=0 & E2P=1 -> 1 : (E1PNa'=1) & (E2P'=0);

//reaction4: 2 K ions bind to pump
[r4] E2P=1 & E2K=0 -> 1 : (E2P'=0) & (E2K'=1);
[rr4] E2P=0 & E2K=1 -> 1 : (E2P'=1) & (E2K'=0);

//reaction5: The channel changes its conformation
[r5] E2K=1 & E1ATPK=0 -> 1 : (E2K'=0) & (E1ATPK'=1);
[rr5] E2K=0 & E1ATPK=1 -> 1 : (E2K'=1) & (E1ATPK'=0);

//reaction6: Pump release 2 K ions inside the cell
[r6] E1ATPK=1 & E1ATP=0 -> 1 : (E1ATPK'=0) & (E1ATP'=1);
[rr6] E1ATPK=0 & E1ATP=1 -> 1 : (E1ATPK'=1) & (E1ATP'=0);

endmodule

module pump2=pump [

```

```
E1ATP=E1ATP2,
E1ATPNa=E1ATPNa2,
E1PNa=E1PNa2,
E2P=E2P2,
E2K=E2K2,
E1ATPK=E1ATPK2
]
endmodule

system // system definition (molecules of the same type do not interact)
(pump ||| pump2) || na || k || p || adp || atp || base_rates
endsystem

// base rates
const double r1rate = 2.5*pow(10.0,11)/(pow((V*AV),3));
const double r2rate = 10000;
const double r3rate = 360;
const double r4rate = 1.5*pow(10.0,7)/(pow((V*AV),2));
const double r5rate = 2*pow(10.0,6)/(pow((V*AV),1));
const double r6rate = 1.15*pow(10.0,4);

const double rr1rate = pow(10.0,5);
const double rr2rate = 1*pow(10.0,5)/(pow((V*AV),1));
const double rr3rate = 8.5*pow(10.0,3)/(pow((V*AV),3));
const double rr4rate = 2*pow(10.0,5)/(pow((V*AV),1));
const double rr5rate = 30;
const double rr6rate = 6*pow(10.0,8)/(pow((V*AV),2));

// module representing the base rates of reactions
module base_rates

[r1] true -> r1rate : true;
[rr1] true ->rr1rate : true;

[r2] true -> r2rate : true;
[rr2] true -> rr2rate : true;

[r3] true -> r3rate : true;
[rr3] true -> rr3rate : true;

[r4] true -> r4rate : true;
[rr4] true -> rr4rate : true;

[r5] true -> r5rate : true;
[rr5] true -> rr5rate : true;

[r6] true -> r6rate : true;
[rr6] true -> rr6rate : true;

endmodule
```

A.3 Modelo Baseado na Química Discreta no PRISM e com Número de Bombas Modelado usando a Abordagem Populacional

```

ctmc

// constants
const double expoente;
const double AV=6.022*pow(10.0,23);
const double V=pow(10.0,-expoente);

const int NI=ceil(0.022*AV*V); // number of Na ions inside the cell
const int NO=ceil(0.14*AV*V); // number of Na ions outside the cell
const int KI=ceil(0.127*AV*V); // number of K ions inside the cell
const int KO=ceil(0.010*AV*V); // number of K ions outside the cell
const int ATPI=ceil(0.005*AV*V); // number of ATP molecules inside the cell
const int ADP=ceil(0.00006*AV*V); // number of ADP molecules inside the cell
const int Pi=ceil(0.00495*AV*V); // number of Phosphate ions inside the cell
const int naFlow=3; // Na molecules that go through membrane each time
const int kFlow=2; // K molecules that go through membrane each time
const int NP; // number of pumps
const int N=ATPI+min(ADP,Pi)+NP; // upper bound limit of ATP molecules

module na
naIn : [0 .. (NI+NO)] init NI; //Number of Na inside cell
naOut : [0..NO+NI] init NO; //Number of Na outside cell

[r1] naIn>=naFlow -> pow(naIn,3) : (naIn'=naIn-naFlow);
[rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);

[r3] naOut<=NO+NI-naFlow -> 1 : (naOut'=naOut+naFlow);
[rr3] naOut>=naFlow -> pow(naOut,3) : (naOut'=naOut-naFlow);

endmodule

module k
kOut : [0..(KO+KI)] init KO; // number of K molecules outside the cell
kIn : [0..(KI+KO)] init KI; // number of K molecules inside the cell

[r4] kOut>=kFlow -> pow(kOut,2) : (kOut'=kOut-kFlow);
[rr4] kOut<=(KO+KI-kFlow) -> 1 : (kOut'=kOut+kFlow);

[r6] kIn<=(KI+KO-kFlow) -> 1 : (kIn'=kIn+kFlow);
[rr6] kIn>=kFlow -> pow(kIn,2) : (kIn'=kIn-kFlow);

endmodule

module p
p : [0..(Pi+ATPI+NP)] init Pi;

[r4] p<(Pi+ATPI+NP) -> 1 : (p'=p+1);

```

```

[rr4] p>0 -> p : (p'=p-1);

endmodule

module atp
atp : [0..N] init ATPI;

[r5] atp>0 -> atp : (atp'=atp-1);
[rr5] atp<N -> 1 : (atp'=atp+1);

endmodule

module adp
adp : [0..(ATPI+ADP+NP)] init ADP;

[r2] adp<(ATPI+ADP+NP) -> 1 : (adp'=adp+1);
[rr2] adp>0 -> adp : (adp'=adp-1);

endmodule

module pump
E1ATP : [0..NP] init NP;
E1ATPNa : [0..NP] init 0;
E1PNa : [0..NP] init 0;
E2P : [0..NP] init 0;
E2K : [0..NP] init 0;
E1ATPK : [0..NP] init 0;

//reaction1: 3 Na ions bind to pump
[r1] E1ATP>0 & E1ATPNa<NP -> E1ATP : (E1ATP'=E1ATP-1) & (E1ATPNa'=E1ATPNa+1);
[rr1] E1ATP<NP & E1ATPNa>0 -> E1ATPNa : (E1ATP'=E1ATP+1) & (E1ATPNa'=E1ATPNa-1);

//reaction2: It happens the fosforilization of the ATP molecule
[r2] E1ATPNa>0 & E1PNa<NP -> E1ATPNa : (E1ATPNa'=E1ATPNa-1) & (E1PNa'=E1PNa+1);
[rr2] E1ATPNa<NP & E1PNa>0 -> E1PNa : (E1ATPNa'=E1ATPNa+1) & (E1PNa'=E1PNa-1);

//reaction3: Pump changes its conformation and releases 3 Na ions outside the cell
[r3] E1PNa>0 & E2P<NP -> E1PNa : (E1PNa'=E1PNa-1) & (E2P'=E2P+1);
[rr3] E1PNa<NP & E2P>0 -> E2P : (E1PNa'=E1PNa+1) & (E2P'=E2P-1);

//reaction4: 2 K ions bind to pump
[r4] E2P>0 & E2K<NP -> E2P : (E2P'=E2P-1) & (E2K'=E2K+1);
[rr4] E2P<NP & E2K>0 -> E2K : (E2P'=E2P+1) & (E2K'=E2K-1);

//reaction5: The channel changes its conformation
[r5] E2K>0 & E1ATPK<NP -> E2K : (E2K'=E2K-1) & (E1ATPK'=E1ATPK+1);
[rr5] E2K<NP & E1ATPK>0 -> E1ATPK : (E2K'=E2K+1) & (E1ATPK'=E1ATPK-1);

//reaction6: Pump release 2 K ions inside the cell
[r6] E1ATPK>0 & E1ATP<NP -> E1ATPK : (E1ATPK'=E1ATPK-1) & (E1ATP'=E1ATP+1);
[rr6] E1ATPK<NP & E1ATP>0 -> E1ATP : (E1ATPK'=E1ATPK+1) & (E1ATP'=E1ATP-1);

endmodule

```

```

// base rates
const double r1rate = 2.5*pow(10.0,11)/(pow((V*AV),3));
const double r2rate = 10000;
const double r3rate = 360;
const double r4rate = 1.5*pow(10.0,7)/(pow((V*AV),2));
const double r5rate = 2*pow(10.0,6)/(pow((V*AV),1));
const double r6rate = 1.15*pow(10.0,4);

const double rr1rate = 100000;
const double rr2rate = 1*pow(10.0,5)/(pow((V*AV),1));
const double rr3rate = 8.5*pow(10.0,3)/(pow((V*AV),3));
const double rr4rate = 2*pow(10.0,5)/(pow((V*AV),1));
const double rr5rate = 30;
const double rr6rate = 6*pow(10.0,8)/(pow((V*AV),2));

// module representing the base rates of reactions
module base_rates
[r1] true -> r1rate : true;
[rr1] true ->rr1rate : true;

[r2] true -> r2rate : true;
[rr2] true -> rr2rate : true;

[r3] true -> r3rate : true;
[rr3] true -> rr3rate : true;

[r4] true -> r4rate : true;
[rr4] true -> rr4rate : true;

[r5] true -> r5rate : true;
[rr5] true -> rr5rate : true;

[r6] true -> r6rate : true;
[rr6] true -> rr6rate : true;

endmodule

```

A.4 Modelo Baseado em Níveis no PRISM

```

ctmc

// constants
const double h;
const int NI=ceil(0.022/h); // number of Na ions inside the cell
const int N0=ceil(0.140/h); // number of Na ions outside the cell
const int KI=ceil(0.127/h); // number of K ions inside the cell
const int K0=ceil(0.010/h); // number of K ions outside the cell
const int ATPI=ceil(0.005/h); // number of ATP molecules inside the cell
const int ADP=ceil(0.00006/h); // number of ADP molecules inside the cell
const int Pi=ceil(0.00495/h); // number of Phosphate ions inside the cell
const int naFlow=3; // Na molecules that go through membrane each time

```

```

const int kFlow=2;      // K molecules that go through membrane each time
const int NP=1;        // number of pumps
const int N=ATPI+min(ADP,Pi)+NP; // upper bound limit of ATP molecules

module na
naIn : [0 .. (NI+NO)] init NI; //Number of Na inside cell
naOut : [0..NO+NI] init NO; //Number of Na outside cell

[r1] naIn>=naFlow -> pow(naIn*h,naFlow) : (naIn'=naIn-naFlow) ;
[rr1] naIn<=(NI+NO-naFlow) -> 1 : (naIn'=naIn+naFlow);

[r3] naOut<=NO+NI-naFlow -> 1 : (naOut'=naOut+naFlow);
[rr3] naOut>=naFlow -> pow(naOut*h,naFlow) : (naOut'=naOut-naFlow);

endmodule

module k
kOut : [0..(KO+KI)] init KO; // number of K molecules outside the cell
kIn : [0..(KI+KO)] init KI; // number of K molecules inside the cell

[r4] kOut>=kFlow -> pow(kOut*h,kFlow) : (kOut'=kOut-kFlow);
[rr4] kOut<=(KO+KI-kFlow) -> 1 : (kOut'=kOut+kFlow);

[r6] kIn<=(KI+KO-kFlow) -> 1 : (kIn'=kIn+kFlow);
[rr6] kIn>=kFlow -> pow(kIn*h,kFlow) : (kIn'=kIn-kFlow);

endmodule

module p
p : [0..(Pi+ATPI+NP)] init Pi;

[r4] p<(Pi+ATPI+NP) -> 1 : (p'=p+1);
[rr4] p>0 -> p*h : (p'=p-1);

endmodule

module atp
atp : [0..N] init ATPI;

[r5] atp>0 -> atp*h : (atp'=atp-1);
[rr5] atp<N -> 1 : (atp'=atp+1);

endmodule

module adp
adp : [0..(ATPI+ADP+NP)] init ADP;

[r2] adp<(ATPI+ADP+NP) -> 1 : (adp'=adp+1);
[rr2] adp>0 -> adp*h : (adp'=adp-1);

endmodule

module pump

```

```

E1ATP : [0..NP] init NP;
E1ATPNa : [0..NP] init 0;
E1PNa : [0..NP] init 0;
E2P : [0..NP] init 0;
E2K : [0..NP] init 0;
E1ATPK : [0..NP] init 0;

//reaction1: 3 Na ions bind to pump
[r1] E1ATP>0 & E1ATPNa<NP -> E1ATP*h : (E1ATP'=E1ATP-1) & (E1ATPNa'=E1ATPNa+1);
[rr1] E1ATP<NP & E1ATPNa>0 -> E1ATPNa*h : (E1ATP'=E1ATP+1) & (E1ATPNa'=E1ATPNa-1);

//reaction2: It happens the fosforilization of the ATP molecule
[r2] E1ATPNa>0 & E1PNa<NP -> E1ATPNa*h : (E1ATPNa'=E1ATPNa-1) & (E1PNa'=E1PNa+1);
[rr2] E1ATPNa<NP & E1PNa>0 -> E1PNa*h : (E1ATPNa'=E1ATPNa+1) & (E1PNa'=E1PNa-1);

//reaction3: Pump changes its conformation and releases 3 Na ions outside the cell
[r3] E1PNa>0 & E2P<NP -> E1PNa*h : (E1PNa'=E1PNa-1) & (E2P'=E2P+1);
[rr3] E1PNa<NP & E2P>0 -> E2P*h : (E1PNa'=E1PNa+1) & (E2P'=E2P-1);

//reaction4: 2 K ions bind to pump
[r4] E2P>0 & E2K<NP -> E2P*h : (E2P'=E2P-1) & (E2K'=E2K+1);
[rr4] E2P<NP & E2K>0 -> E2K*h : (E2P'=E2P+1) & (E2K'=E2K-1);

//reaction5: The channel changes its conformation
[r5] E2K>0 & E1ATPK<NP -> E2K*h : (E2K'=E2K-1) & (E1ATPK'=E1ATPK+1);
[rr5] E2K<NP & E1ATPK>0 -> E1ATPK*h : (E2K'=E2K+1) & (E1ATPK'=E1ATPK-1);

//reaction6: Pump release 2 K ions inside the cell
[r6] E1ATPK>0 & E1ATP<NP -> E1ATPK*h : (E1ATPK'=E1ATPK-1) & (E1ATP'=E1ATP+1);
[rr6] E1ATPK<NP & E1ATP>0 -> E1ATP*h : (E1ATPK'=E1ATPK+1) & (E1ATP'=E1ATP-1);
endmodule

// base rates
const double r1rate = 2.5*pow(10.0,11)/h;
const double r2rate = 10000/h;
const double r3rate = 360/h;
const double r4rate = 1.5*pow(10.0,7)/h;
const double r5rate = 2*pow(10.0,6)/h;
const double r6rate = 1.15*pow(10.0,4)/h;

const double rr1rate = 100000/h;
const double rr2rate = 1*pow(10.0,5)/h;
const double rr3rate = 8.5*pow(10.0,3)/h;
const double rr4rate = 2*pow(10.0,5)/h;
const double rr5rate = 30/h;
const double rr6rate = 6*pow(10.0,8)/h;

// module representing the base rates of reactions
module base_rates

[r1] true -> r1rate : true;
[rr1] true ->rr1rate : true;

[r2] true -> r2rate : true;

```

```

[rr2] true -> rr2rate : true;

[r3] true -> r3rate : true;
[rr3] true -> rr3rate : true;

[r4] true -> r4rate : true;
[rr4] true -> rr4rate : true;

[r5] true -> r5rate : true;
[rr5] true -> rr5rate : true;

[r6] true -> r6rate : true;
[rr6] true -> rr6rate : true;

endmodule

```

A.5 Modelo Discreto no BIOLAB

```

begin parameters
Na      6.02214179e23
V       1e-12
DIV     Na*V
sADP    int(0.00006*DIV+3*(1/3))
sATP    int(0.005*DIV+3*(1/3))
sA      0
sP      int(0.00495*DIV+3*(1/3))
sNaI    int(0.022*DIV+3*(1/3))
sNaO    int(0.14*DIV+3*(1/3))
sKI     int(0.127*DIV+3*(1/3))
sKO     int(0.01*DIV+3*(1/3))
spump   1000

# reactions
sT1     2.5e11 / DIV^3
sT2     10000
sT3     360
sT4     1.5e7 / DIV^2
sT5     2e6 / DIV
sT6     11500
sR1     100000
sR2     100000 / DIV
sR3     8500 / DIV^3
sR4     200000 / DIV
sR5     30
sR6     6e8 / DIV^2
end parameters

begin molecule types
# Pump -----
# The pump has two forms(Based on Mirlaine's presentation):
# A <-> When binding Na
# V <-> When binging K

```

```

# There are 3 sites for Na and 2 for K
# Pump end-----
pump(f~A~V,a,n1,n2,n3,k1,k2)

# Na and K -----
# Na and K have one site and a state:
# I <-> Inside the cell
# O <-> Outside the cell
# Na and K end-----
Na(s~I~O~C,n)
K(s~I~O~C,k)

# A and P -----
# ADP is A(u,p1!1,p2!2,p3).P(p!1).P(p!2) and ATP is A(u,p1!1,p2!2,p3!3).P(p!1).P(p!2).P(p!3)
# A and P end-----
P(p)
A(u,p1,p2,p3)

end molecule types

begin seed species
pump(f~A,a!1,n1,n2,n3,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4) spump
pump(f~V,a,n1,n2,n3,k1,k2) 0
Na(s~I,n)                sNaI
Na(s~O,n)                sNaO
Na(s~C,n)                0
K(s~I,k)                 sKI
K(s~O,k)                 sKO
K(s~C,k)                 0
P(p)                     sP
A(u,p1,p2,p3)            sA
A(u,p1!1,p2!2,p3!3).P(p!1).P(p!2).P(p!3) sATP
A(u,p1!1,p2!2,p3).P(p!1).P(p!2)          sADP
end seed species

begin observables
Molecules oADP A(u,p1!1,p2!2,p3).P(p!1).P(p!2)
Molecules oATP A(u,p1!1,p2!2,p3!3).P(p!1).P(p!2).P(p!3)
Molecules oP P(p)
Molecules oKIn K(s~I,k)
Molecules oNaIn Na(s~I,n)
Molecules oKOut K(s~O,k)
Molecules oNaOut Na(s~O,n)
Molecules oE1ATP pump(f~A,a!1,n1,n2,n3,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4)
Molecules oE1ATPNa pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).
P(p!3).P(p!4).Na(s~C,n!5).Na(s~C,n!6).Na(s~C,n!7)
end observables

begin reaction rules
# Rules-----
# E1.ATP + Na <-> Na.E1.ATP
# Na.E1.ATP <-> Na.E2.P + ADP
# Na.E2.P <-> E2.P + NaOut

```

```

# E2.P + KOut <-> KOut.E2 + P
# KOut.E2 + ATP <-> KOut.E1.ATP
# KOut.E1.ATP <-> Kin E1.ATP
# E1 means pump(f~A) and E2 means pump(f~V)
# -----
# And now the code:

# E1.ATP + 3Na <-> Na3.E1.ATP
pump(f~A,a!1,n1,n2,n3,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4) + Na(s~I,n) + Na(s~I,n) +
Na(s~I,n) <-> pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4).
Na(s~C,n!5). Na(s~C,n!6).Na(s~C,n!7) sT1,sR1

# Na3.E1.ATP <-> Na3.E1.P + ADP
pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).A(u!1,p1!2,p2!3,p3!4).P(p!2).P(p!3).P(p!4).Na(s~C,n!5).
Na(s~C,n!6).Na(s~C,n!7) <-> pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).P(p!1).Na(s~C,n!5).Na(s~C,n!6).
Na(s~C,n!7) + A(u,p1!2,p2!3,p3).P(p!2).P(p!3) sT2,sR2

# Na3.E1.P <-> E2.P + 3NaOut
pump(f~A,a!1,n1!5,n2!6,n3!7,k1,k2).P(p!1).Na(s~C,n!5).Na(s~C,n!6).Na(s~C,n!7) <->
pump(f~V,a!1,n1,n2,n3,k1,k2).P(p!1) + Na(s~O,n) + Na(s~O,n) + Na(s~O,n) sT3,sR3

# E2.P + 2KOut <-> KOut2.E2 + P
pump(f~V,a!1,n1,n2,n3,k1,k2).P(p!1) + K(s~O,k) + K(s~O,k) <->
pump(f~V,a,n1,n2,n3,k1!1,k2!2).K(s~C,k!1).K(s~C,k!2) + P(p) sT4,sR4

# KOut2.E2 + ATP <-> KOut2.E1.ATP
pump(f~V,a,n1,n2,n3,k1!1,k2!2).K(s~C,k!1).K(s~C,k!2) + A(u,p1!3,p2!4,p3!5).P(p!3).
P(p!4).P(p!5) <-> pump(f~A,a!6,n1,n2,n3,k1!1,k2!2).K(s~C,k!1).K(s~C,k!2).
A(u!6,p1!3,p2!4,p3!5).P(p!3).P(p!4).P(p!5) sT5,sR5

# KOut2.E1.ATP <-> 2Kin + E1.ATP
pump(f~A,a!6,n1,n2,n3,k1!1,k2!2).K(s~C,k!1).K(s~C,k!2).A(u!6,p1!3,p2!4,p3!5).P(p!3).
P(p!4).P(p!5) <-> pump(f~A,a!6,n1,n2,n3,k1,k2).A(u!6,p1!3,p2!4,p3!5).P(p!3).P(p!4).
P(p!5) + K(s~I,k) + K(s~I,k) sT6,sR6

end reaction rules

generate_network({overwrite=>1});
simulate_ssa({suffix=>ssa,t_end=>10,n_steps=>100});

```

Apêndice B

Tendências de Taxas de Transição dos Modelos da Na,K-ATPase

B.1 Tendências do Modelo PRISM Baseado na Química Discreta

```
const double threshold;
//trend formulas
formula rate_r1 = (((naIn>=naFlow)&(E1ATP>0)) ? r1rate*pow(naIn,naFlow)*E1ATP : 0);
formula rate_r2 = ((E1ATPNa>0 & E1PNa<NP) ? r2rate*E1ATPNa : 0);
formula rate_r3 = (((naOut<=NO+NI-naFlow)&(E1PNa>0 & E2P<NP)) ? r3rate*E1PNa : 0);
formula rate_r4 = ( ((kOut>=kFlow)&(E2P>0 & E2K<NP)) ? r4rate*pow(kOut,kFlow)*E2P : 0 );
formula rate_r5 = (((atp>0)&(E2K>0 & E1ATPK<NP)) ? r5rate*atp*E2K : 0);
formula rate_r6 = (((E1ATPK>0 & E1ATP<NP)&(kIn<=(KI+KO-kFlow))) ? r6rate*E1ATPK : 0);
formula rate_rr1 = (((naIn<=(NI+NO-naFlow))&(E1ATP<NP&E1ATPNa>0)) ? rr1rate*E1ATPNa : 0);
formula rate_rr2 = ((adp>0)&(E1ATPNa<NP & E1PNa>0) ? rr2rate*adp*E1PNa : 0);
formula rate_rr3 = (((naOut>=naFlow)&(E1PNa<NP & E2P>0)) ? rr3rate*pow(naOut,naFlow)*E2P : 0);
formula rate_rr4 = (((kOut<=(KO+KI-kFlow))&(p>0)&(E2P<NP & E2K>0)) ? rr4rate*p*E2K : 0);
formula rate_rr5 = (((E2K<NP & E1ATPK>0)) ? rr5rate*E1ATPK : 0);
formula rate_rr6 = (((E1ATPK<NP & E1ATP>0)&(kIn>=kFlow)) ? rr6rate*E1ATP*pow(kIn,kFlow) : 0);

formula exit_rate = rate_r1 + rate_r2 + rate_r3 + rate_r4 + rate_r5 + rate_r6 + rate_rr1 +
rate_rr2 + rate_rr3 + rate_rr4 + rate_rr5 + rate_rr6;

//forwards rates

//trend r1
formula rate_r1_d = (exit_rate=0 ? 0:rate_r1/exit_rate);
label "trend_r1_up" = (rate_r1_d>=threshold);

//trend r2
formula rate_r2_d = (exit_rate=0 ? 0:rate_r2/exit_rate);
label "trend_r2_up" = (rate_r2_d>=threshold);

//trend r3
formula rate_r3_d = (exit_rate=0 ? 0:rate_r3/exit_rate);
```

```

label "trend_r3_up" = (rate_r3_d>=threshold);

//trend r4
formula rate_r4_d = (exit_rate=0 ? 0:rate_r4/exit_rate);
label "trend_r4_up" = (rate_r4_d>=threshold);

//trend r2
formula rate_r5_d = (exit_rate=0 ? 0:rate_r5/exit_rate);
label "trend_r5_up" = (rate_r5_d>=threshold);

//trend r6
formula rate_r6_d = (exit_rate=0 ? 0:rate_r6/exit_rate);
label "trend_r6_up" = (rate_r6_d>=threshold);

//backward rates

//trend rr1
formula rate_rr1_d = (exit_rate=0 ? 0:rate_rr1/exit_rate);
label "trend_rr1_up" = (rate_rr1_d>=threshold);

//trend rr2
formula rate_rr2_d = (exit_rate=0 ? 0:rate_rr2/exit_rate);
label "trend_rr2_up" = (rate_rr2_d>=threshold);

//trend rr3
formula rate_rr3_d = (exit_rate=0 ? 0:rate_rr3/exit_rate);
label "trend_rr3_up" = (rate_rr3_d>=threshold);

//trend rr4
formula rate_rr4_d = (exit_rate=0 ? 0:rate_rr4/exit_rate);
label "trend_rr4_up" = (rate_rr4_d>=threshold);

//trend rr2
formula rate_rr5_d = (exit_rate=0 ? 0:rate_rr5/exit_rate);
label "trend_rr5_up" = (rate_rr5_d>=threshold);

//trend rr6
formula rate_rr6_d = (exit_rate=0 ? 0:rate_rr6/exit_rate);
label "trend_rr6_up" = (rate_rr6_d>=threshold);

```

B.2 Tendências do Modelo PRISM Baseado em Níveis

```

formula rate_r1 = ((E1ATP>0 & E1ATPNa<NP) ? r1rate*pow(naIn*h,naFlow)*E1ATP*h : 0);
formula rate_r2 = ((E1ATPNa>0 & E1PNa<NP) ? r2rate*E1ATPNa*h : 0);
formula rate_r3 = ((E1PNa>0 & E2P<NP) ? r3rate*E1PNa*h : 0);
formula rate_r4 = ((E2P>0 & E2K<NP) ? r4rate*pow(kOut*h,kFlow)*E2P*h : 0);
formula rate_r5 = ((E2K>0 & E1ATPK<NP) ? r5rate*atp*h*E2K*h : 0);
formula rate_r6 = ((E1ATPK>0 & E1ATP<NP) ? r6rate*E1ATPK*h : 0);
formula rate_rr1 = ((E1ATP<NP&E1ATPNa>0) ? rr1rate*E1ATPNa*h : 0);
formula rate_rr2 = ((E1ATPNa<NP & E1PNa>0) ? rr2rate*adp*h*E1PNa*h : 0);

```

```

formula rate_rr3 = ((E1PNa<NP & E2P>0) ? rr3rate*pow(naOut*h,naFlow)*E2P*h : 0);
formula rate_rr4 = ((E2P<NP & E2K>0) ? rr4rate*p*h*E2K*h : 0);
formula rate_rr5 = ((E2K<NP & E1ATPK>0) ? rr5rate*E1ATPK*h : 0);
formula rate_rr6 = ((E1ATPK<NP & E1ATP>0) ? rr6rate*E1ATP*h*pow(kIn*h,kFlow) : 0);

formula exit_rate = rate_r1 + rate_r2 + rate_r3 + rate_r4 + rate_r5 + rate_r6 +
rate_rr1 + rate_rr2 + rate_rr3 + rate_rr4 + rate_rr5 + rate_rr6;

//forwards rates

//trend r1
formula rate_r1_d = (exit_rate=0 ? 0:rate_r1/exit_rate);
label "trend_r1_up" = (rate_r1_d>=threshold);

//trend r2
formula rate_r2_d = (exit_rate=0 ? 0:rate_r2/exit_rate);
label "trend_r2_up" = (rate_r2_d>=threshold);

//trend r3
formula rate_r3_d = (exit_rate=0 ? 0:rate_r3/exit_rate);
label "trend_r3_up" = (rate_r3_d>=threshold);

//trend r4
formula rate_r4_d = (exit_rate=0 ? 0:rate_r4/exit_rate);
label "trend_r4_up" = (rate_r4_d>=threshold);

//trend r5
formula rate_r5_d = (exit_rate=0 ? 0:rate_r5/exit_rate);
label "trend_r5_up" = (rate_r5_d>=threshold);

//trend r6
formula rate_r6_d = (exit_rate=0 ? 0:rate_r6/exit_rate);
label "trend_r6_up" = (rate_r6_d>=threshold);

//backward rates

//trend rr1
formula rate_rr1_d = (exit_rate=0 ? 0:rate_rr1/exit_rate);
label "trend_rr1_up" = (rate_rr1_d>=threshold);

//trend rr2
formula rate_rr2_d = (exit_rate=0 ? 0:rate_rr2/exit_rate);
label "trend_rr2_up" = (rate_rr2_d>=threshold);

//trend rr3
formula rate_rr3_d = (exit_rate=0 ? 0:rate_rr3/exit_rate);
label "trend_rr3_up" = (rate_rr3_d>=threshold);

//trend rr4
formula rate_rr4_d = (exit_rate=0 ? 0:rate_rr4/exit_rate);
label "trend_rr4_up" = (rate_rr4_d>=threshold);

//trend rr2

```

```

formula rate_rr5_d = (exit_rate=0 ? 0:rate_rr5/exit_rate);
label "trend_rr5_up" = (rate_rr5_d>=threshold);

//trend rr6
formula rate_rr6_d = (exit_rate=0 ? 0:rate_rr6/exit_rate);
label "trend_rr6_up" = (rate_rr6_d>=threshold);

```

B.3 Tendências no modelo BIOLAB

Esta seção introduz a definição das tendências de reações na ferramenta BIOLAB. No modelo BIOLAB da Na,K-ATPase a verificação das tendências das reações é totalmente incorporada na especificação da propriedade. Por exemplo, a propriedade (B.1) permite verificar se a reação r_1 sempre exibe uma tendência negativa durante a evolução do sistema:

$$G \text{ NOT } (\{ r_1 / (r_1 + rr_6) \geq 0.5 \} \text{ AND } (\{ oE1ATP > 0 \} \text{ AND } \{ oE1ATPNa < oNP \})) \quad (\text{B.1})$$

onde $r_1 = 1.14e - 24 * oNaIn * oNaIn * oNaIn$ e $rr_6 = 1.65e - 15 * oKIn * oKIn$. Os termos $oNaIn$, $oKIn$, $oE1ATP$, $oE1ATPNa$ e oNP devem ser definidos no bloco *observables* do arquivo de descrição dos modelos e se referem, respectivamente, a concentração de sódio dentro da célula, a concentração de potássio dentro da célula, o número de bombas na configuração E_1 .ATP, o número de bombas na configuração Na_3 . E_1 .ATP e o número total de bombas. O valor $1,14e - 24$ se refere à constante de taxa da reação r_1 na química discreta, enquanto o valor $1,65e - 15$ se refere a constante de taxa da reação rr_6 , a única outra reação que poderia ocorrer, além de r_1 , quando a configuração da bomba é E_1 .ATP.

Se o resultado da verificação da propriedade (B.1) é **verdadeiro**, considerando um valor de $\theta = 0,99$, o parâmetro da região de indiferença Δ e o tempo de simulação t_{end} , com probabilidade maior ou igual à $0,99 + \Delta$ a reação r_1 apresenta uma tendência sempre negativa ao longo da evolução do sistema nas primeiras t_{end} unidades de tempo.

A definição das tendências das reações no BIOLAB é bem mais verbosa quando comparada ao PRISM. Isso porque não existe uma integração completa entre os parâmetros definidos no bloco *parameters* do arquivo de descrição do modelo (BIONETGEN) e o arquivo de especificação da propriedade a ser verificada. Somente os rótulos para as concentrações dos substratos a serem observados ou constantes inteiras, definidos no bloco *observables*, podem ser usados na especificação das propriedades. Dessa forma, é preciso especificar novamente todas as constantes de taxas quando necessárias, mesmo

que elas já estejam definidas no módulo *parameters*.

Além disso, foi necessário adicionar à ferramenta BIOLAB o reconhecimento de números reais com formato em notação científica e também do parênteses para separar a ordem das operações, possibilitando as definições das tendências das reações.

Apêndice C

Modelos do Sistema de Oscilação Intracelular de Cálcio e do Canal BK

Esta seção apresenta as versões completas dos modelos PRISM baseados em níveis do sistema de oscilação intracelular de cálcio e do sistema composto pelo canal iônico BK e a oscilação intracelular de cálcio.

C.1 Modelo PRISM Baseado em Níveis da Oscilação Intracelular de Cálcio

```
ctmc

const double h;

const int caInI=ceil(0.5/h);
const int caInM=caInI+caOutI+caSRI;

const int caOutI=ceil(0.1/h);
const int caOutM=caInI+caOutI+caSRI;

const int caSRI=ceil(1.9/h);
const int caSRM=caInI+caOutI+caSRI;

const int rAcI=ceil(0.001/h);
const int rAcM=rAcI+rInI;

const int rInI=ceil(0.009/h);
const int rInM=rAcI+rInI;

const double v0=1;
const double v1=1;
const double beta=0.5;
const double k=10;
const double kf=1;
```

```

const double VM2=6.5;
const double K2=0.1;
const double VM3=50;
const double Kca=0.2;
const double kd=5000;
const double kr=5;
const double a=40000;
const double d=100;

module ca
caIn : [0..caInM] init caInI;
caOut : [0..caOutM] init caOutI;
caSR : [0..caSRM] init caSRI;

[r1f] caIn<caInM -> 1 : (caIn'=caIn+1);

[r1b] caIn>0 -> caIn*h : (caIn'=caIn-1);

[r2f] caIn>0 & caSR<caSRM -> pow(caIn*h,2)/(pow(caIn*h,2)+pow(K2,2)) :
      (caIn'=caIn-1) & (caSR'=caSR+1);

[r2b] caIn<caInM & caSR>0 & caIn>0 -> (pow(caIn*h,4)/(1+((a/d)*pow(caIn*h,4))))*
      (pow(caSR*h,2)/(pow(Kca,2)+pow(caSR*h,2))) : (caIn'=caIn+1) & (caSR'=caSR-1);

[r3] caSR>0 & caIn<caInM -> caSR*h : (caIn'=caIn+1) & (caSR'=caSR-1);

[r4f] caIn>0 -> pow(caIn*h,4) : true;

endmodule

module ip3
rAc : [0..rAcM] init rAcI;
rIn : [0..rInM] init rInI;

[r2f] rAc>0 -> 1 : true;

[r2b] rAc>0 -> rAc*h : true;

[r4f] rIn<rInM & rAc>0 -> rAc*h : (rIn'=rIn+1) & (rAc'=rAc-1);

[r4b] rAc<rAcM & rIn>0 -> rIn*h : (rAc'=rAc+1) & (rIn'=rIn-1);

endmodule

module start
[r1f] true -> ((v0)+(v1*beta))/h : true;

[r1b] true -> k/h : true;

[r2f] true -> VM2/h : true;

[r2b] true -> beta*(a/d)*VM3/h : true;

```

```
[r3] true -> kf/h : true;

[r4f] true -> kd/h : true;

[r4b] true -> kr/h : true;

endmodule
```

C.2 Modelo PRISM Baseado em Níveis da Integração da Oscilação Intracelular de Cálcio e do Canal BK

```
ctmc

const double h2;
const double h1;
const int r=ceil(h2/h1);

const int caInI=ceil(0.5/h1);
const int caInM=caInI+caOutI+caSRI;

const int caOutI=ceil(0.1/h1);
const int caOutM=caInI+caOutI+caSRI;

const int caSRI=ceil(1.9/h1);
const int caSRM=caInI+caOutI+caSRI;

const int rAcI=ceil(0.1/h1);
const int rAcM=rAcI+rInI;

const int rInI=ceil(0.9/h1);
const int rInM=rAcI+rInI;

const double v0=1;
const double v1=1;
const double beta=0.5;
const double k=10;
const double kf=1;
const double VM2=6.5;
const double K2=0.1;
const double VM3=50;
const double Kca=0.2;
const double kd=5000;
const double kr=5;
const double a=40000;
const double d=100;

module ca
caIn : [0..caInM] init caInI;
```

```

caOut : [0..caOutM] init caOutI;
caSR   : [0..caSRM] init caSRI;

[r1f] caIn<caInM -> 1 : (caIn'=min(caIn+r,caInM));

[r1b] caIn>0 -> caIn*h1 : (caIn'=max(caIn-r,0));

[r2f] caIn>0 & caSR<caSRM -> pow(caIn*h1,2)/(pow(caIn*h1,2)+pow(K2,2)) :
      (caIn'=max(caIn-r,0)) & (caSR'=min(caSR+r,caSRM));

[r2b] caIn<caInM & caSR>0 & caIn>0 -> (pow(caIn*h1,4)/(1+((a/d)*pow(caIn*h1,4))))*
      (pow(caSR*h1,2)/(pow(Kca,2)+pow(caSR*h1,2))) : (caIn'=min(caIn+r,caInM)) &
      (caSR'=max(caSR-r,0));

[r3] caSR>0 & caIn<caInM -> caSR*h1 : (caIn'=min(caIn+r,caInM)) & (caSR'=max(caSR-r,0));

[r4f] caIn>0 -> pow(caIn*h1,4) : true;

//BK
[r1_2] caIn<caInM -> 1 : (caIn'=caIn+1);
[r2_1] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r2_3] caIn<caInM -> 1 : (caIn'=caIn+1);
[r3_2] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r3_4] caIn<caInM -> 1 : (caIn'=caIn+1);
[r4_3] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r4_5] caIn<caInM -> 1 : (caIn'=caIn+1);
[r5_4] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r11_12] caIn<caInM -> 1 : (caIn'=caIn+1);
[r12_11] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r12_13] caIn<caInM -> 1 : (caIn'=caIn+1);
[r13_12] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r13_14] caIn<caInM -> 1 : (caIn'=caIn+1);
[r14_13] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

[r14_15] caIn<caInM -> 1 : (caIn'=caIn+1);
[r15_14] caIn>0 -> caIn*h1 : (caIn'=caIn-1);

endmodule

module ip3

rAc : [0..rAcM] init rAcI;
rIn : [0..rInM] init rInI;

[r2f] rAc>0 -> 1 : true;

[r2b] rAc>0 -> rAc*h1 : true;

```

```
[r4f] rIn<rInM & rAc>0 -> rAc*h1 : (rIn'=min(rIn+r,rInM)) & (rAc'=max(rAc-r,0));
```

```
[r4b] rAc<rAcM & rIn>0 -> rIn*h1 : (rAc'=min(rAc+r,rAcM)) & (rIn'=max(rIn-r,0));
```

```
endmodule
```

```
const int NP=1;
```

```
module channel
```

```
o1: [0..NP] init NP;
```

```
o2: [0..NP] init 0;
```

```
o3: [0..NP] init 0;
```

```
o4: [0..NP] init 0;
```

```
o5: [0..NP] init 0;
```

```
o6: [0..NP] init 0;
```

```
o7: [0..NP] init 0;
```

```
o8: [0..NP] init 0;
```

```
o9: [0..NP] init 0;
```

```
o10: [0..NP] init 0;
```

```
o11: [0..NP] init 0;
```

```
o12: [0..NP] init 0;
```

```
o13: [0..NP] init 0;
```

```
o14: [0..NP] init 0;
```

```
o15: [0..NP] init 0;
```

```
o16: [0..NP] init 0;
```

```
o17: [0..NP] init 0;
```

```
o18: [0..NP] init 0;
```

```
o19: [0..NP] init 0;
```

```
o20: [0..NP] init 0;
```

```
o21: [0..NP] init 0;
```

```
o22: [0..NP] init 0;
```

```
o23: [0..NP] init 0;
```

```
o24: [0..NP] init 0;
```

```
o25: [0..NP] init 0;
```

```
r1_2] o1>0 & o2<NP -> o1*h1 : (o1'=o1-1) & (o2'=o2+1);
```

```
[r2_1] o2>0 & o1<NP -> o2*h1 : (o2'=o2-1) & (o1'=o1+1);
```

```
[r1_6] o1>0 & o6<NP -> o1*h1 : (o1'=o1-1) & (o6'=o6+1);
```

```
[r6_1] o6>0 & o1<NP -> o6*h1 : (o6'=o6-1) & (o1'=o1+1);
```

```
[r1_21] o1>0 & o21<NP -> o1*h1 : (o1'=o1-1) & (o21'=o21+1);
```

```
[r21_1] o21>0 & o1<NP -> o21*h1 : (o21'=o21-1) & (o1'=o1+1);
```

```
[r2_3] o2>0 & o3<NP -> o2*h1 : (o2'=o2-1) & (o3'=o3+1);
```

```
[r3_2] o3>0 & o2<NP -> o3*h1 : (o3'=o3-1) & (o2'=o2+1);
```

```
[r2_7] o2>0 & o7<NP -> o2*h1 : (o2'=o2-1) & (o7'=o7+1);
```

```
[r7_2] o7>0 & o2<NP -> o7*h1 : (o7'=o7-1) & (o2'=o2+1);
```

```
[r2_22] o2>0 & o22<NP -> o2*h1 : (o2'=o2-1) & (o22'=o22+1);
```

```
[r22_2] o22>0 & o2<NP -> o22*h1 : (o22'=o22-1) & (o2'=o2+1);
```

```
[r3_4] o3>0 & o4<NP -> o3*h1 : (o3'=o3-1) & (o4'=o4+1);
```

- [r4_3] $o4 > 0 \ \& \ o3 < NP \rightarrow o4 * h1 : (o4' = o4 - 1) \ \& \ (o3' = o3 + 1);$
- [r3_8] $o3 > 0 \ \& \ o8 < NP \rightarrow o3 * h1 : (o3' = o3 - 1) \ \& \ (o8' = o8 + 1);$
[r8_3] $o8 > 0 \ \& \ o3 < NP \rightarrow o8 * h1 : (o8' = o8 - 1) \ \& \ (o3' = o3 + 1);$
- [r3_23] $o3 > 0 \ \& \ o23 < NP \rightarrow o3 * h1 : (o3' = o3 - 1) \ \& \ (o23' = o23 + 1);$
[r23_3] $o23 > 0 \ \& \ o3 < NP \rightarrow o23 * h1 : (o23' = o23 - 1) \ \& \ (o3' = o3 + 1);$
- [r4_5] $o4 > 0 \ \& \ o5 < NP \rightarrow o4 * h1 : (o4' = o4 - 1) \ \& \ (o5' = o5 + 1);$
[r5_4] $o5 > 0 \ \& \ o4 < NP \rightarrow o5 * h1 : (o5' = o5 - 1) \ \& \ (o4' = o4 + 1);$
- [r4_9] $o4 > 0 \ \& \ o9 < NP \rightarrow o4 * h1 : (o4' = o4 - 1) \ \& \ (o9' = o9 + 1);$
[r9_4] $o9 > 0 \ \& \ o4 < NP \rightarrow o9 * h1 : (o9' = o9 - 1) \ \& \ (o4' = o4 + 1);$
- [r4_24] $o4 > 0 \ \& \ o24 < NP \rightarrow o4 * h1 : (o4' = o4 - 1) \ \& \ (o24' = o24 + 1);$
[r24_4] $o24 > 0 \ \& \ o4 < NP \rightarrow o24 * h1 : (o24' = o24 - 1) \ \& \ (o4' = o4 + 1);$
- [r5_10] $o5 > 0 \ \& \ o10 < NP \rightarrow o5 * h1 : (o5' = o5 - 1) \ \& \ (o10' = o10 + 1);$
[r10_5] $o10 > 0 \ \& \ o5 < NP \rightarrow o10 * h1 : (o10' = o10 - 1) \ \& \ (o5' = o5 + 1);$
- [r5_25] $o5 > 0 \ \& \ o25 < NP \rightarrow o5 * h1 : (o5' = o5 - 1) \ \& \ (o25' = o25 + 1);$
[r25_5] $o25 > 0 \ \& \ o5 < NP \rightarrow o25 * h1 : (o25' = o25 - 1) \ \& \ (o5' = o5 + 1);$
- [r6_16] $o6 > 0 \ \& \ o16 < NP \rightarrow o6 * h1 : (o6' = o6 - 1) \ \& \ (o16' = o16 + 1);$
[r16_6] $o16 > 0 \ \& \ o6 < NP \rightarrow o16 * h1 : (o16' = o16 - 1) \ \& \ (o6' = o6 + 1);$
- [r7_17] $o7 > 0 \ \& \ o17 < NP \rightarrow o7 * h1 : (o7' = o7 - 1) \ \& \ (o17' = o17 + 1);$
[r17_7] $o17 > 0 \ \& \ o7 < NP \rightarrow o17 * h1 : (o17' = o17 - 1) \ \& \ (o7' = o7 + 1);$
- [r8_18] $o8 > 0 \ \& \ o18 < NP \rightarrow o8 * h1 : (o8' = o8 - 1) \ \& \ (o18' = o18 + 1);$
[r18_8] $o18 > 0 \ \& \ o8 < NP \rightarrow o18 * h1 : (o18' = o18 - 1) \ \& \ (o8' = o8 + 1);$
- [r9_19] $o9 > 0 \ \& \ o19 < NP \rightarrow o9 * h1 : (o9' = o9 - 1) \ \& \ (o19' = o19 + 1);$
[r19_9] $o19 > 0 \ \& \ o9 < NP \rightarrow o19 * h1 : (o19' = o19 - 1) \ \& \ (o9' = o9 + 1);$
- [r10_20] $o10 > 0 \ \& \ o20 < NP \rightarrow o10 * h1 : (o10' = o10 - 1) \ \& \ (o20' = o20 + 1);$
[r20_10] $o20 > 0 \ \& \ o10 < NP \rightarrow o20 * h1 : (o20' = o20 - 1) \ \& \ (o10' = o10 + 1);$
- [r11_12] $o11 > 0 \ \& \ o12 < NP \rightarrow o11 * h1 : (o11' = o11 - 1) \ \& \ (o12' = o12 + 1);$
[r12_11] $o12 > 0 \ \& \ o11 < NP \rightarrow o12 * h1 : (o12' = o12 - 1) \ \& \ (o11' = o11 + 1);$
- [r11_16] $o11 > 0 \ \& \ o16 < NP \rightarrow o11 * h1 : (o11' = o11 - 1) \ \& \ (o16' = o16 + 1);$
[r16_11] $o16 > 0 \ \& \ o11 < NP \rightarrow o16 * h1 : (o16' = o16 - 1) \ \& \ (o11' = o11 + 1);$
- [r12_13] $o12 > 0 \ \& \ o13 < NP \rightarrow o12 * h1 : (o12' = o12 - 1) \ \& \ (o13' = o13 + 1);$
[r13_12] $o13 > 0 \ \& \ o12 < NP \rightarrow o13 * h1 : (o13' = o13 - 1) \ \& \ (o12' = o12 + 1);$
- [r12_17] $o12 > 0 \ \& \ o17 < NP \rightarrow o12 * h1 : (o12' = o12 - 1) \ \& \ (o17' = o17 + 1);$
[r17_12] $o17 > 0 \ \& \ o12 < NP \rightarrow o17 * h1 : (o17' = o17 - 1) \ \& \ (o12' = o12 + 1);$
- [r13_14] $o13 > 0 \ \& \ o14 < NP \rightarrow o13 * h1 : (o13' = o13 - 1) \ \& \ (o14' = o14 + 1);$
[r14_13] $o14 > 0 \ \& \ o13 < NP \rightarrow o14 * h1 : (o14' = o14 - 1) \ \& \ (o13' = o13 + 1);$
- [r13_18] $o13 > 0 \ \& \ o18 < NP \rightarrow o13 * h1 : (o13' = o13 - 1) \ \& \ (o18' = o18 + 1);$

```
[r18_13] o18>0 & o13<NP -> o18*h1 : (o18'=o18-1) & (o13'=o13+1);

[r14_15] o14>0 & o15<NP -> o14*h1 : (o14'=o14-1) & (o15'=o15+1);
[r15_14] o15>0 & o14<NP -> o15*h1 : (o15'=o15-1) & (o14'=o14+1);

[r14_19] o14>0 & o19<NP -> o14*h1 : (o14'=o14-1) & (o19'=o19+1);
[r19_14] o19>0 & o14<NP -> o19*h1 : (o19'=o19-1) & (o14'=o14+1);

[r15_20] o15>0 & o20<NP -> o15*h1 : (o15'=o15-1) & (o20'=o20+1);
[r20_15] o20>0 & o15<NP -> o20*h1 : (o20'=o20-1) & (o15'=o15+1);

[r16_21] o16>0 & o21<NP -> o16*h1 : (o16'=o16-1) & (o21'=o21+1);
[r21_16] o21>0 & o16<NP -> o21*h1 : (o21'=o21-1) & (o16'=o16+1);

[r17_22] o17>0 & o22<NP -> o17*h1 : (o17'=o17-1) & (o22'=o22+1);
[r22_17] o22>0 & o17<NP -> o22*h1 : (o22'=o22-1) & (o17'=o17+1);

[r18_23] o18>0 & o23<NP -> o18*h1 : (o18'=o18-1) & (o23'=o23+1);
[r23_18] o23>0 & o18<NP -> o23*h1 : (o23'=o23-1) & (o18'=o18+1);

[r19_24] o19>0 & o24<NP -> o19*h1 : (o19'=o19-1) & (o24'=o24+1);
[r24_19] o24>0 & o19<NP -> o24*h1 : (o24'=o24-1) & (o19'=o19+1);

[r20_25] o20>0 & o25<NP -> o20*h1 : (o20'=o20-1) & (o25'=o25+1);
[r25_20] o25>0 & o20<NP -> o25*h1 : (o25'=o25-1) & (o20'=o20+1);
```

endmodule

module start

```
[r1f] true -> ((v0)+(v1*beta))/h2 : true;

[r1b] true -> k/h2: true;

[r2f] true -> VM2/h2 : true;

[r2b] true -> beta*(a/d)*VM3/h2: true;

[r3] true -> kf/h2 : true;

[r4f] true -> kd/h2 : true;

[r4b] true -> kr/h2 : true;

[r1_2] true -> r1_2 : true;
[r2_1] true -> r2_1 : true;

[r1_6] true -> r1_6 : true;
[r6_1] true -> r6_1 : true;

[r1_21] true -> r1_21 : true;
[r21_1] true -> r21_1 : true;

[r2_3] true -> r2_3 : true;
```

```
[r3_2] true -> r3_2 : true;

[r2_7] true -> r2_7 : true;
[r7_2] true -> r7_2 : true;

[r2_22] true -> r2_22 : true;
[r22_2] true -> r22_2 : true;

[r3_4] true -> r3_4 : true;
[r4_3] true -> r4_3 : true;

[r3_8] true -> r3_8 : true;
[r8_3] true -> r8_3 : true;

[r3_23] true -> r3_23 : true;
[r23_3] true -> r23_3 : true;

[r4_5] true -> r4_5 : true;
[r5_4] true -> r5_4 : true;

[r4_9] true -> r4_9 : true;
[r9_4] true -> r9_4 : true;

[r4_24] true -> r4_24 : true;
[r24_4] true -> r24_4 : true;

[r5_10] true -> r5_10 : true;
[r10_5] true -> r10_5 : true;

[r5_25] true -> r5_25 : true;
[r25_5] true -> r25_5 : true;

[r6_16] true -> r6_16 : true;
[r16_6] true -> r16_6 : true;

[r7_17] true -> r7_17 : true;
[r17_7] true -> r17_7 : true;

[r8_18] true -> r8_18 : true;
[r18_8] true -> r18_8 : true;

[r9_19] true -> r9_19 : true;
[r19_9] true -> r19_9 : true;

[r10_20] true -> r10_20 : true;
[r20_10] true -> r20_10 : true;

[r11_12] true -> r11_12 : true;
[r12_11] true -> r12_11 : true;

[r11_16] true -> r11_16 : true;
[r16_11] true -> r16_11 : true;

[r12_13] true -> r12_13 : true;
```

```
[r13_12] true -> r13_12 : true;

[r12_17] true -> r12_17 : true;
[r17_12] true -> r17_12 : true;

[r13_14] true -> r13_14 : true;
[r14_13] true -> r14_13 : true;

[r13_18] true -> r13_18 : true;
[r18_13] true -> r18_13 : true;

[r14_15] true -> r14_15 : true;
[r15_14] true -> r15_14 : true;

[r14_19] true -> r14_19 : true;
[r19_14] true -> r19_14 : true;

[r15_20] true -> r15_20 : true;
[r20_15] true -> r20_15 : true;

[r16_21] true -> r16_21 : true;
[r21_16] true -> r21_16 : true;

[r17_22] true -> r17_22 : true;
[r22_17] true -> r22_17 : true;

[r18_23] true -> r18_23 : true;
[r23_18] true -> r23_18 : true;

[r19_24] true -> r19_24 : true;
[r24_19] true -> r24_19 : true;

[r20_25] true -> r20_25 : true;
[r25_20] true -> r25_20 : true;

endmodule

const double r1_2=150*60/h1;
const double r2_1=140*60/h1;

const double r1_6=800*60/h1;
const double r6_1=3300*60/h1;

const double r1_21=480*60/h1;
const double r21_1=23000.0*60/h1;

const double r2_3=0.1*60/h1;
const double r3_2=0.1*60/h1;

const double r2_7=0.1*60/h1;
const double r7_2=0.1*60/h1;

const double r2_22=1700.0*60/h1;
const double r22_2=6900.0*60/h1;
```

```
const double r3_4=0.1*60/h1;
const double r4_3=0.1*60/h1;

const double r3_8=2600*60/h1;
const double r8_3=3200*60/h1;

const double r3_23=510*60/h1;
const double r23_3=1100*60/h1;

const double r4_5=0.1*60/h1;
const double r5_4=0.1*60/h1;

const double r4_9=20000*60/h1;
const double r9_4=3900*60/h1;

const double r4_24=0.1*60/h1;
const double r24_4=850*60/h1;

const double r5_10=2500*60/h1;
const double r10_5=3000*60/h1;

const double r5_25=2200*60/h1;
const double r25_5=21000*60/h1;

const double r6_16=2600*60/h1;
const double r16_6=2900*60/h1;

const double r7_17=18000*60/h1;
const double r17_7=45000*60/h1;

const double r8_18=15000*60/h1;
const double r18_8=45000*60/h1;

const double r9_19=1000*60/h1;
const double r19_9=13000*60/h1;

const double r10_20=8700*60/h1;
const double r20_10=1500*60/h1;

const double r11_12=6600*60/h1;
const double r12_11=40*60/h1;

const double r11_16=25000*60/h1;
const double r16_11=5600*60/h1;

const double r12_13=190*60/h1;
const double r13_12=130*60/h1;

const double r12_17=0.1*60/h1;
const double r17_12=0.1*60/h1;

const double r13_14=25000*60/h1;
const double r14_13=800*60/h1;
```

```
const double r13_18=3300*60/h1;
const double r18_13=25000*60/h1;

const double r14_15=5000*60/h1;
const double r15_14=22*60/h1;

const double r14_19=190*60/h1;
const double r19_14=7500*60/h1;

const double r15_20=1.9*60/h1;
const double r20_15=7000*60/h1;

const double r16_21=0.1*60/h1;
const double r21_16=0.1*60/h1;

const double r17_22=0.1*60/h1;
const double r22_17=0.1*60/h1;

const double r18_23=0.1*60/h1;
const double r23_18=0.1*60/h1;

const double r19_24=0.1*60/h1;
const double r24_19=14000.0*60/h1;

const double r20_25=130*60/h1;
const double r25_20=6000*60/h1;
```

Apêndice D

Solução Determinística do Sistema de Oscilação Intracelular de Cálcio no MATLAB

D.1 Conjunto de ODEs

A função $oscillation(t,x)$ que segue descreve como os substratos envolvidos na oscilação intracelular de cálcio variam no tempo segundo [Borghans et al., 1997]. As três equações no vetor **osc** descrevem, respectivamente, a variação do potássio no meio citosólico ($\frac{dC_{acyt}}{dt}$), a variação do potássio no meio SR ($\frac{dC_{asr}}{dt}$) e a variação dos canais receptores de IP_3 ativos ($\frac{dC_{Ac}}{dt}$) no tempo.

```
function osc = oscilation(t,x);
v0=1;
v1=1;
beta=0.5;
k=10;
kf=1;
kr=5;
kd=5000;
K2=0.1;
Ky=0.2;
VM2=6.5;
VM3=50;
a=40000;
d=100;
Vin=v0+(v1*beta);
r=a/d;

osc=[Vin-(VM2*(x(1)^2)/(K2^2 + x(1)^2))+ ( beta*( x(3)*r*x(1)^4/(1+(r*x(1)^4)))*
      VM3*(x(2)^2/(Ky^2+x(2)^2)) )+kf*x(2)-k*x(1);
-kf*x(2)+(VM2*(x(1)^2)/(K2^2 + x(1)^2))-(beta*( x(3)*r*x(1)^4/(1+(r*x(1)^4)))*VM3
      *(x(2)^2/(Ky^2+x(2)^2)) );
```

```
-kd*x(1)^4*x(3)+kr*(1-x(3))];
```

D.2 Resolvendo o Conjunto de ODEs

Para resolver o conjunto de ODEs descrito na seção anterior, primeiramente é preciso definir os valores iniciais do potássio no meio citosólico, do potássio no meio SR e dos canais receptores de IP_3 ativos, segundo dados da Tabela 8.1:

```
n0=[0.5;1.9;0.1]
```

Finalmente, a função do MATLAB `ode15s` é empregada para resolver as ODEs, considerando $t_{end} = 5$ minutos:

```
[t,x] = ode15s('oscillation',[0 5],n0);
```

Referências Bibliográficas

- Aidley, D. J. e Stanfield, P. R. (1996). *Ion Channels*. Cambridge University Press.
- Albers, R. W. (1967). Biochemical aspects of active transport. *Annu. Rev. Biochem.*, 36:727–756.
- Andrei, O. e Calder, M. (2010). A model and analysis of the AKAP scaffold. *Electronic Notes in Theoretical Computers Science*.
- Ballarini, P.; Mardare, R. e Mura, I. (2009). Analysing biochemical oscillation through probabilistic model checking. *Electronic Notes in Theoretical Computers Science*.
- Borghans, J. M.; Dupont, G. e Goldbeter, A. (1997). Complex intracellular calcium oscillations a theoretical exploration of possible mechanisms. *Biophysical Chemistry*, 66(1):25 – 41.
- Bryant, R. E. (1986a). Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, 35(8):677–691.
- Bryant, R. E. (1986b). Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–310.
- Calder, M.; Vyshemirsky, V.; Gilbert, D. e Orton, R. (2006). Analysis of signalling pathways using continuous time markov chains. *Trans. Computational Systems Biology*, 4:44–67.
- Campos, F. O. (2008). Modelagem computacional da eletrofisiologia cardíaca: o desenvolvimento de um novo modelo para células de camundongos e a avaliação de novos esquemas numéricos. Master’s thesis, Universidade Federal de Juiz de Fora.
- Campos, S. V. A. e Clarke, E. (2001). The verus language: representing time efficiently with bdds. *Theoretical Computer Science*, 253:95–118.
- Chapman, J. B.; Johnson, E. A. e Kootsey, J. M. (1983). Electrical and biochemical properties of an enzyme model of the sodium pump. *Membrane Biology*, 74:139–153.

- Cimatti, A.; Clarke, E.; Giunchiglia, E.; Giunchiglia, F.; Pistore, M.; Roveri, M.; Sebastiani, R. e Tacchella, A. (2002). Nusmv 2: An opensource tool for symbolic model checking. In Brinksma, E. e Larsen, K., editores, *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pp. 241–268. Springer Berlin / Heidelberg.
- Cindy e Geisiane (2010). Membrana plasmática e transporte passivo. Disponível em: <<http://webbed.com.br/geisy/>>. Acesso em: 18 ago. 2011, 23:21:00.
- Ciobanu, G.; Ciubotariu, V. e Tanasa, B. (2002). A π -calculus model of the Na pump. *Genome Informatics*, 13:469–471.
- Ciocchetta, F.; Degasperi, A.; Hillston, J. e Calder, M. (2008). Some investigations concerning the ctmc and the ode model derived from bio-pepa. *Electronic Notes in Theoretical Computers Science*.
- Ciocchetta, F. e Guerriero, M. L. (2009). Modelling biological compartments in bio-pepa. *Electronic Notes in Theoretical Computer Science*, pp. 77–95.
- Clarke, E. M. e Emerson, E. A. (1981). Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop, London, UK*. Springer-Verlag., pp. 52–71.
- Clarke, E. M.; Faeder, J. R.; Langmead, C. J.; Harris, L. A.; Jha, S. K. e Axel, L. (2008). Statistical model checking in biolab: Applications to the automated analysis of t-cell receptor signaling pathway. In *CMSB '08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, pp. 231–250, Berlin, Heidelberg. Springer-Verlag.
- Clarke, E. M.; Grumberg, O. e Peled, D. A. (1999). *Model Checking*. The Mit Press.
- Colquhoun, D. e Hawkes, A. G. (1977). Relaxation and fluctuations of membrane currents that flow through drug-operated channels. *Proc. R. Soc. London*.
- Cruz, J. (2011). [Conversa Pessoal; 27-Maio-2011].
- de Souza, E. A.; Silva e Muntz, R. R. (1992). *Métodos Computacionais de Solução de Cadeias de Markov: Aplicações a Sistemas de Computação e Comunicação*. Instituto de Informática da UFRGS.
- Dean, R. B. (1941). Theories of electrolyte equilibrium in muscle. *Biol. Symp.*, 3:331–348.

- Dworakowska, B. e Dolowy, K. (2000). Ion channels-related diseases. *Acta Biochimica Polonica*, 47.
- Faeder, J. R.; Blinov, M. L. e Hlavacek, W. S. (2009). *Rule-based modeling of biochemical networks with BioNetGen*, volume 500 of *Methods in Molecular Biology: System Biology*, pp. 1–55. Humana Press.
- Finkbeiner, B. e Sipma, H. (2001). Checking finite traces using alternating automata. In *Proceedings of Runtime Verification 1 (RV01)*, pp. 44–60.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.
- Godefroid, P.; Peled, D. e Staskauskas, M. (1996). Using partial-order methods in the formal validation of industrial concurrent programs. *IEEE Trans. Softw. Eng.*, 22(7):496–507.
- Grumberg, O.; Livne, S. e Markovitch, S. (2003). Learning to order BDD variables in verification. *Journal of Artificial Intelligence Research*, 18:83–116.
- Guldberg, C. (1984). Concerning the laws of chemical affinity. *C. M. Forhandlingar: Videnskabs-Selskabet i Christiana*, 111.
- Hartonas-Garmhausen, V.; Campos, S. e Clarke, E. (1999). Probverus: Probabilistic symbolic model checking. In *FORMAL METHODS FOR REAL-TIME AND PROB. SYS., LNCS 1601*, pp. 96–110. Springer.
- Heath, J. e et al (2008). Probabilistic model checking of complex biological pathways. *Theoretical Computer Science*, 319:239–257.
- Hernández, J. A. e Chifflet, S. (2000). Eletrogenic properties of the sodium pump in a dynamic model of membrane transport. *Membrane Biology*, 176:41–52.
- Hillston, A.; Kwiatkowska, M.; Norman, G. e Parker, D. (2006). Prism: A tool for automatic verification of probabilistic systems. *Proc. 12th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, 3920:441–444.
- Jha, S. K.; Clarke, E. M.; Langmead, C. J.; Legay, A.; Platzer, A. e Zuliani, P. (2009). A bayesian approach to model checking biological systems. In *CMSB '09: Proceedings of the 7th International Conference on Computational Methods in Systems Biology*, pp. 218–234, Berlin, Heidelberg. Springer-Verlag.

- Karp, G. (2008). *Cell and Molecular Biology*. Wiley.
- Kwiatkowska, M.; Norman, G. e Parker, D. (2004). Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)*, 6(2):128–142.
- Kwiatkowska, M.; Norman, G. e Parker, D. (2007). Stochastic model checking. In Bernardo, M. e Hillston, J., editores, *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, volume 4486 of *LNCS (Tutorial Volume)*, pp. 220–270. Springer.
- Kwiatkowska, M.; Norman, G. e Parker, D. (2008). Using probabilistic model checking in systems biology. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):14–21.
- Kwiatkowska, M.; Norman, G. e Parker, D. (2010). *Symbolic Systems Biology*, chapter Probabilistic Model Checking for Systems Biology. Jones and Bartlett.
- Lauger, P. (1991). *Electrogenic Ion Pumps*, volume 5. Sinauer Associates Inc., Sunderland, Massachusetts, U.S.A.
- Lehninger, A.; Nelson, D. L. e Cox, M. M. (2004). *Lehninger Principles of Biochemistry*. W. H. Freeman, 4 edição.
- Magleby, K. L. e Weiss, D. S. (1990). Identifying kinetic gating mechanisms for ion channels by using two-dimensional distributions of simulated dwell times. In *Proceedings: Biological Sciences*, pp. 220–228.
- MATLAB (2011). web site matlab. <http://www.mathworks.com/products/matlab/>.
- McMillan, K. L. (1992). *Symbolic Model Checking*. PhD thesis, Carnegie Mellon.
- Emerson Flamarion da Cruz (2006). Modelos markovianos para canais iônicos em membranas celulares. Master's thesis, Universidade de São Paulo/Instituto de Física.
- Monteiro, P. T.; Ropers, D.; Mateescu, R.; Freitas, A. T. e de Jong, H. (2008). Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233.
- Moss, B. L.; Silberberg, S. D.; Nimigeon, C. M. e Magleby, K. L. (1999). ca^{2+} -dependent gating mechanisms for dSlo, a large-conductance ca^{2+} -activated k^+ (BK) channel. *Biophysical Journal*.

- Neher, E. e Stevens, C. F. (1977). Conductance fluctuations and ionic pores in membranes. *Annu. Rev. Biophys. Bioeng.*
- Owen, R.; McKeever, S.; Davies, J. e Garfinkel, A. (2006). Toward provably correct models of ventricular cell function. *Computers in Cardiology.*
- Pancrazio, J. J. (1994). Ion channel events simulated with the program simstate. *Computer Methods and Programs in BioMedicine.*
- Pinto, M. C.; Foss, L.; Mombach, J. C. M. e Ribeiro, L. (2007). Modelling, property verification and behavioural equivalence of lactose operon regulation. *Comput. Biol. Med.*, 37:134–148.
- Pronk, T.; de Vink, E.; Bosnacki, D. e Breit, T. (2007). Stochastic modeling of codon bias with prism. *Proc. 3rd Int. Workshop Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2007).*
- Queille, J. P. e Sifakis, J. (1982). A temporal logic to deal with fairness in transition systems. *Foundations of Science. SFCS'08. 23rd Annual Symposium on*, pp. 217–225.
- Rodrigues, A. M.; Almeida, A.-C. G. e Infantosi, A. F. C. (2008). Effect of palytoxin on the sodium-potassium pump: model and simulation. *Physical Biology*, 5:1–11.
- Rothberg, B. S. e Magleby, K. L. (2000). Voltage and Ca^{2+} activation of single large-conductance Ca^{2+} -activated K^{+} channels described by a two-tiered allosteric gating mechanism. *General Physiology*, pp. 75–99.
- Sansom, M. e et al (1989). Markov, fractal, diffusion and related models of ion channel gating: a comparison with experimental data from two ion channels. *Biophysical Journal.*
- Sauro, H. M.; Harel, D.; Kwiatkowska, M.; Shaffer, C. A.; Uhrmacher, A. M.; Hucka, M.; Mendes, P.; Strömback, L. e Tyson, J. J. (2006). Challenges for modeling and simulation methods in systems biology. In *Proceedings of the 38th conference on Winter simulation, WSC '06*, pp. 1720–1730. Winter Simulation Conference.
- Sepúlveda, F. V. e Bezanilla, F. (2005). *Pumps, Transporters and Ion Channels*. Kluwer Academic/Plenum Publishers.
- Song, M. A. J. (2004). *The UML-CAFE: an Environment to Specify and Verify Transactional Systems*. PhD thesis, Universidade Federal de Minas Gerais.

- sophion (2010). Ion channels are targets for selective drugs. Disponível em: <<http://www.sophion.dk/about-sophion/technology/ion-channels.aspx>>. Acesso em: 18 ago. 2011, 23:39:00.
- Spieler, D. (2009). Model checking of oscillatory and noisy periodic behavior in markovian population models. Master's thesis, Saarland University/Faculty of Natural Sciences and Technology I.
- Ullah, M.; Schmidt, H.; Cho, K. H. e Wolkenhauer, O. (2006). Deterministic modelling and stochastic simulation of biochemical pathways using MATLAB. *Systems Biology*, 153(2):53–60.
- Waage, P. (1984). Experiments for determining the affinity law. *Forhandlinger: Videnskabs-Selskabet i Christiana*, 92.
- Wald, A. (1945). Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186.
- Weiner, I. e Wingo, C. (1997). Hypokalemia—consequences, causes, and correction. *Journal of the American Society of Nephrology*, 8(7):1179–1188.
- Wolkenhauer, O.; Ullah, M.; Kolch, W. e Cho, K.-H. (2004). Modeling and simulation of intracellular dynamics: Choosing an appropriate framework. *IEEE Transactions on NanoBioscience*, 3:200–207.