



1506  
**UNIVERSITÀ  
DEGLI STUDI  
DI URBINO  
CARLO BO**

UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Dipartimento di Scienze Biomolecolari  
Corso di Laurea in Scienze Biologiche

---

Tesi di Laurea

## **MODELLAZIONE FORMALE APPLICATA AD ANALISI *IN SILICO* DI RETI METABOLICHE**

Relatore:  
Chiar.mo Prof. Mauro Magnani

Candidato:  
Luca Salvatore Lorello

Correlatori:  
Chiar.mo Prof. Marco Bernardo  
Dott.ssa Noemi Bigini  
Dott.ssa Francesca Pierigè

---

Anno Accademico 2017-2018



A quei pochi che leggeranno effettivamente.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Contesto della tesi . . . . .	1
1.2	Obiettivi e contributi della tesi . . . . .	2
1.3	Organizzazione della tesi . . . . .	3
<b>2</b>	<b>Modellazione formale</b>	<b>4</b>
2.1	Caratteristiche dei metabolismi biochimici . . . . .	4
2.1.1	Cinetica enzimatica di Michaelis-Menten . . . . .	6
2.1.2	Diffusione passiva . . . . .	7
2.2	Analisi <i>in silico</i> . . . . .	7
2.2.1	Verifica formale . . . . .	10
2.3	Catene di Markov . . . . .	12
2.3.1	Linguaggio di modellazione di PRISM . . . . .	14
2.4	Algebre di processi . . . . .	17
2.4.1	Bio-PEPA . . . . .	19
2.5	Logiche temporali . . . . .	22
2.5.1	Continuous Stochastic Logic . . . . .	24
2.6	Simulazioni su catene di Markov a tempo continuo . . . . .	24
2.7	Model checking di catene di Markov a tempo continuo . . . . .	25
2.7.1	Model checking simbolico . . . . .	27
<b>3</b>	<b>Caso di studio: sviluppo di eritrociti ingegnerizzati per il trattamento della deficienza GAMT</b>	<b>31</b>
3.1	Deficienza GAMT . . . . .	31
3.1.1	Metabolismo della creatina . . . . .	31
3.1.2	Deficienza dell'enzima guanidinoacetato metiltransferasi . . . . .	33
3.1.3	Sintomi . . . . .	34
3.1.4	Diagnosi . . . . .	34
3.1.5	Terapie . . . . .	35
3.2	Red Cell Loader . . . . .	36
3.3	Cromatografia liquida ad alte prestazioni . . . . .	38
3.4	Spettrometria di massa a ionizzazione per elettrospray . . . . .	38
3.5	Stato dell'arte del trattamento della deficienza GAMT . . . . .	39

3.5.1	Esperimento sull'uptake del guanidinoacetato . . . . .	39
3.5.2	Esperimenti sul dosaggio della creatina prodotta . . . . .	41
<b>4</b>	<b>Costruzione del modello per il trattamento della deficienza GAMT</b>	<b>43</b>
4.1	Via metabolica da modellare . . . . .	43
4.1.1	Cinetiche di reazione . . . . .	44
4.2	Modello Bio-PEPA . . . . .	46
4.2.1	Ingresso di guanidinoacetato negli eritrociti . . . . .	47
4.2.2	Sintesi di creatina in eritrociti incapsulati con SAM eso- gena . . . . .	50
4.2.3	Predizione della sintesi di creatina in eritrociti incapsu- lati con SAM sintasi di <i>E. coli</i> . . . . .	52
4.3	Modelli PRISM . . . . .	56
<b>5</b>	<b>Analisi simulativa del trattamento della deficienza GAMT</b>	<b>60</b>
5.1	Simulazione dell'esperimento sull'uptake del guanidinoacetato .	60
5.2	Simulazione dell'esperimento sulla sintesi di creatina . . . . .	61
5.3	Simulazione di eritrociti ingegnerizzati con GAMT e SAMS . .	65
<b>6</b>	<b>Model checking del trattamento della deficienza GAMT</b>	<b>72</b>
6.1	Limiti e requisiti del model checking . . . . .	72
6.2	Proprietà generate da Bio-PEPA . . . . .	73
6.3	Proprietà definite . . . . .	78
<b>7</b>	<b>Conclusioni</b>	<b>80</b>
7.1	Lavoro svolto . . . . .	80
7.2	Necessità di un approccio formale integrato . . . . .	80
7.3	Sviluppi futuri . . . . .	81
	<b>Ringraziamenti</b>	<b>91</b>

# Elenco delle figure

2.1	Modello a stati e transizioni del dogma centrale della biologia molecolare . . . . .	9
2.2	Rappresentazione grafica di una catena di Markov a tempo continuo . . . . .	15
2.3	Albero di decisione relativo alla funzione logica $(x \vee y) \wedge z$ . . .	28
2.4	Binary decision diagram relativo alla funzione logica $(x \vee y) \wedge z$	29
3.1	Red Cell Loader . . . . .	37
3.2	Rate di ingresso del guanidinoacetato negli eritrociti a diverse concentrazioni . . . . .	40
3.3	Diagramma dei doppi reciproci dell'ingresso di guanidinoacetato negli eritrociti . . . . .	41
4.1	Andamento della velocità di reazione in funzione della concentrazione di substrato in un enzima di Michaelis-Menten . . . .	45
5.1	Validazione dell'esperimento di uptake di guanidinoacetato . .	61
5.2	Validazione dell'esperimento di sintesi di creatina: andamento del guanidinoacetato intracellulare . . . . .	63
5.3	Validazione dell'esperimento di sintesi di creatina: andamento della creatina . . . . .	64
5.4	Simulazione di controllo con SAMS nativa (modello 4.8) . . . .	66
5.5	Simulazione dell'incapsulamento di 0.05 mg di SAMS (modello 4.9) . . . . .	66
5.6	Simulazione dell'incapsulamento di 0.1 mg di SAMS (modello 4.10) . . . . .	67
5.7	Simulazione dell'incapsulamento di 0.5 mg di SAMS (modello 4.11) . . . . .	67
5.8	Simulazione dell'incapsulamento di 1 mg di SAMS (modello 4.12)	68
5.9	Simulazione dell'incapsulamento di 2.5 mg di SAMS (modello 4.13) . . . . .	68
5.10	Simulazione dell'incapsulamento di 5 mg di SAMS (modello 4.14)	69
5.11	Sintesi di creatina negli eritrociti ingegnerizzati con quantità variabili di SAMS di <i>E. coli</i> . . . . .	70

5.12 Consumo di guanidinoacetato negli eritrociti ingegnerizzati con quantità variabili di SAMS di <i>E. coli</i> . . . . .	71
--	----



# Elenco delle tabelle

2.1	Parametri del modello sulla regolazione genica basata su dime- rizzazione di <i>E. coli</i> . . . . .	21
2.2	Tabella di verità della funzione logica $(x \vee y) \wedge z$ . . . . .	28
3.1	Concentrazioni di guanidinoacetato e creatina in eritrociti non ingegnerizzati. . . . .	42
3.2	Concentrazioni di guanidinoacetato e creatina in eritrociti in- gegnerizzati con GAMT e 25 $\mu M$ di SAM. . . . .	42
3.3	Concentrazioni di guanidinoacetato e creatina in eritrociti in- gegnerizzati con GAMT e 125 $\mu M$ di SAM. . . . .	42
4.1	Esperimento di uptake con 10 $\mu M$ di guanidinoacetato extracel- lulare . . . . .	48
4.2	Esperimento di uptake con 25 $\mu M$ di guanidinoacetato extracel- lulare . . . . .	48
4.3	Esperimento di uptake con 50 $\mu M$ di guanidinoacetato extracel- lulare . . . . .	49
4.4	Esperimento di uptake con 100 $\mu M$ di guanidinoacetato extra- cellulare . . . . .	49
4.5	Esperimento di sintesi della creatina con 10 $\mu M$ di S-adenosil metionina incapsulata . . . . .	50
4.6	Esperimento di sintesi della creatina con 50 $\mu M$ di S-adenosil metionina incapsulata . . . . .	51
4.7	Esperimento di controllo della sintesi della creatina con S-adenosil metionina endogena . . . . .	51
4.8	Esperimento di controllo della sintesi della creatina con SAM sintasi endogena . . . . .	52
4.9	Esperimento di sintesi della creatina con 0.05 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	53
4.10	Esperimento di sintesi della creatina con 0.1 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	53
4.11	Esperimento di sintesi della creatina con 0.5 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	54

4.12	Esperimento di sintesi della creatina con 1 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	54
4.13	Esperimento di sintesi della creatina con 2.5 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	55
4.14	Esperimento di sintesi della creatina con 5 mg di SAM sintasi di <i>E. coli</i> incapsulati . . . . .	55

# Capitolo 1

## Introduzione

### 1.1 Contesto della tesi

Nonostante i progressi nel campo biologico della ricerca *in vitro*, i costi elevati e la potenziale assenza di informazioni preliminari su un dato fenomeno rendono al giorno d'oggi l'utilizzo delle tecniche *in silico* sempre più interessanti, non solo come strumenti per l'osservazione e la replicazione del fenomeno, ma soprattutto in termini predittivi.

In questa tesi viene descritto un approccio *formale*, basato su algebre di processi, catene di Markov e logiche temporali, applicato alla costruzione di modelli di reti metaboliche, adatti sia all'aspetto replicativo che a quello predittivo tipici di un "esperimento" *in silico*, facendo riferimento in particolare alle tecniche di analisi basate su *simulazione* e *verifica formale*.

I metodi formali risultano particolarmente efficaci per l'analisi di sistemi complessi, eliminando ogni sorgente di errore attraverso appositi linguaggi che permettono altresì di automatizzare lo studio dei sistemi stessi. Partendo da una rappresentazione corretta del metabolismo oggetto d'esame, che catturi aspetti quantitativi, interattivi e cinetici, si possono ottenere, tra le altre, informazioni di natura quantitativa, temporale e prestazionale che non soffrono dell'incertezza tipica degli esperimenti *in vitro*.

Sebbene l'approccio simulativo abbia goduto di notevole successo grazie alla sua semplicità di applicazione, non risulta tanto potente quanto quello basato su verifica, mancando dell'esaustività che contraddistingue quest'ultimo. Viceversa, l'approccio basato su verifica risulta computazionalmente oneroso e richiede strumenti matematici più complessi. Queste differenze rendono auspicabile un utilizzo sinergico di entrambi gli approcci, per ottenere il maggior numero di informazioni possibile, a partire da una singola descrizione formale del metabolismo.

In questa tesi viene affrontata la modellazione di patologie di origine genetica. Esse sono caratterizzate da disordini metabolici permanenti che si riflettono su alterazioni più o meno estese delle vie coinvolte. Tra le terapie

proposte per contrastare tali alterazioni, l'utilizzo di bioreattori circolanti nel flusso ematico si presenta come un approccio promettente, essendo il sangue in grado di raggiungere l'intero organismo. Bioreattori ideali per circolare nel sangue sono gli eritrociti che, se opportunamente ingegnerizzati, sono in grado di incapsulare enzimi in forma nativa che sopperiscono alla mancanza di funzionalità di quelli patologici.

La patologia utilizzata come esempio per questa tesi è la deficienza dell'enzima guanidinoacetato metiltransferasi (GAMT), necessario per la catalisi dell'ultima tappa della via biosintetica della creatina. Oltre ai deficit, a livello cognitivo e muscolare, causati dalla carenza di creatina, l'accumulo del guanidinoacetato, precursore della creatina che nei mammiferi non è impiegato in nessun'altra reazione, causa danni al sistema nervoso. La semplice supplementazione di creatina con la dieta non è dunque sufficiente ad alleviare i sintomi della patologia. Per una terapia efficace risulta quindi necessario limitare la produzione del guanidinoacetato o, più auspicabilmente, eliminarlo attivamente.

## 1.2 Obiettivi e contributi della tesi

Questa tesi si prefigge l'obiettivo di descrivere ed esemplificare le tecniche di modellazione e verifica formale più adatte alla cattura di aspetti interattivi e quantitativi di un sistema biochimico.

L'approccio descritto è utilizzato per *validare* esperimenti *in vitro* già effettuati e per *predire* l'andamento di esperimenti futuri, relativamente a un nuovo approccio farmacologico, basato su eritrociti ingegnerizzati per il trattamento della deficienza GAMT, usato come caso di studio.

Più in dettaglio, viene proposto un modello espresso nel linguaggio di modellazione algebrico Bio-PEPA, che descrive le interazioni tra il guanidinoacetato dell'ambiente plasmatico di pazienti affetti da deficienza GAMT ed eritrociti modificati da un apparato, chiamato Red Cell Loader, per funzionare da bioreattori in grado di sequestrare guanidinoacetato e liberare creatina. I parametri del modello sono stati successivamente istanziati a valori tali da riprodurre esperimenti *in vitro* già effettuati, allo scopo di validare la fedeltà del modello, e a valori corrispondenti ad esperimenti programmati ma non ancora effettuati, per la predizione dell'andamento.

Le istanze del modello sono state automaticamente convertite in un linguaggio di modellazione di più basso livello, adeguato all'analisi tramite simulazioni ad eventi discreti e model checking utilizzando il software PRISM. Sebbene il modello proposto presenti assunzioni non necessariamente soddisfatte *in vivo*, le simulazioni prodotte a partire da tale modello fungeranno da validazione degli esperimenti che verranno realizzati *in vitro* mentre le proprietà verificate tramite model checking forniranno una quantificazione preliminare dell'efficacia della terapia.

## 1.3 Organizzazione della tesi

Nel capitolo 2 sono descritte in maniera generale le caratteristiche dei metabolismi e le tecniche di modellazione e analisi più adatte a catturare tali caratteristiche. Il capitolo 3 presenta come caso di studio la deficienza GAMT, un difetto genetico nella via di sintesi di creatina, e il principio di funzionamento del trattamento proposto a base di eritrociti ingegnerizzati tramite Red Cell Loader.

Nei capitoli successivi viene mostrato il processo di sperimentazione *in silico* nelle due fasi di creazione del modello (cap. 4) e analisi, tramite simulazioni a eventi discreti e model checking (capp. 5 e 6 rispettivamente), sia per validare risultati già ottenuti che per predire comportamenti non ancora osservati *in vitro*.

Il capitolo 7 conclude il lavoro riassumendo vantaggi e svantaggi delle due tecniche di analisi e descrive potenziali sviluppi futuri, sempre in relazione al caso di studio preso in esame.

Si consiglia a un lettore con formazione biologica di cominciare la lettura del capitolo 2 a partire dalla sezione 2.2. Un lettore con formazione informatica può limitarsi a leggere le sezioni 2.1 e 2.4.1.

## Capitolo 2

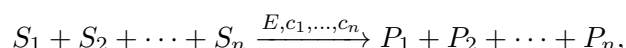
# Modellazione formale

In questo capitolo vengono illustrate le caratteristiche “interattive” comuni a tutti i metabolismi biochimici (sezione 2.1) e gli strumenti *in silico* utili per la descrizione (sezione 2.2), con particolare enfasi verso la modellazione formale tramite catene di Markov (sezione 2.3), algebre di processi (sezione 2.4) e logiche temporali (sezione 2.5), e l’analisi, tramite simulazioni (sezione 2.6) e model checking (sezione 2.7) di tali caratteristiche.

### 2.1 Caratteristiche dei metabolismi biochimici

Una rete metabolica è per sua natura costituita da un insieme di reazioni che *interagiscono* tra loro in termini di quantità e cinetiche.

In generale una singola reazione può essere espressa come:



dove l’insieme  $\mathbb{S} = \{S_1, S_2, \dots, S_n\}$  costituisce i *substrati* (o reagenti) della reazione e l’insieme  $\mathbb{P} = \{P_1, P_2, \dots, P_n\}$  ne costituisce i *prodotti*; i substrati sono specie chimiche che diminuiscono di quantità, mentre i prodotti aumentano, essendo i primi trasformati nei secondi. L’insieme, opzionale,  $\mathbb{C} = \{E, c_1, \dots, c_n\}$  rappresenta i *catalizzatori* (tipicamente costituiti da almeno un enzima e zero o più cofattori) della reazione, sostanze che *accelerano* la cinetica di reazione, ma che non vengono modificate in quantità; in termini biologici, le reazioni non catalizzate avvengono così lentamente che si può assumere che non possano avvenire affatto, in assenza di catalizzatori.

Affinché avvenga una reazione, secondo la teoria degli urti, è necessario che si verifichi una collisione tra le molecole di substrato, con sufficiente energia e corretto orientamento spaziale. Una reazione chimica è quindi un processo aleatorio, la cui velocità (rate) dipende da fattori quali temperatura (energia cinetica delle molecole) e concentrazione di substrati.

La catalisi altera la cinetica, aumentando la probabilità delle collisioni, grazie alla divisione della reazione in sottoreazioni termodinamicamente più favorite. Per meccanismi di tipo allosterico (ovvero dove modifiche di forma dell'enzima si riflettono in alterazioni della cinetica), gli enzimi possono essere modulati in presenza di *promotori* o *inibitori*, che aumentano o diminuiscono, rispettivamente, il rate di reazione.

Una o più reazioni costituiscono un *metabolismo* se interagiscono tra loro. Di seguito sono esemplificate alcune interazioni:

- se due reazioni sono in successione, i prodotti della prima costituiscono i reagenti della seconda: globalmente la reazione più lenta può costituire un collo di bottiglia;
- se due reazioni utilizzano uno stesso substrato, vi competono;
- se una reazione produce o sottrae un modulatore per l'enzima di un'altra reazione, la cinetica di quest'ultima viene alterata;
- se una reazione produce o distrugge l'enzima (o suoi attivatori e inattivatori) di un'altra reazione, la cinetica di quest'ultima viene alterata;
- se sono presenti sistemi di feedback all'interno di una stessa reazione (e.g. se il prodotto è anche un inibitore dell'enzima), la cinetica viene alterata nel tempo.

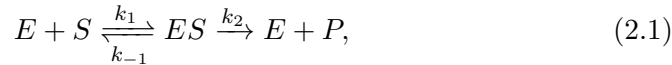
In termini modellistici, un metabolismo costituisce un *sistema reattivo* dove le singole componenti operano su un pool condiviso di risorse su cui competono con modalità stocastiche e dove le risorse possono costituire esse stesse nuove componenti (e.g. una stessa sostanza può comportarsi da enzima in una reazione e da substrato in un'altra). L'elevata dimensione del pool si riflette in un elevato grado di parallelismo, che rende il numero di interazioni estremamente elevato e, conseguentemente, l'analisi del sistema estremamente complicata e tipicamente non trattabile senza l'ausilio di *astrazioni* quali:

- ragionare in termini di aliquote di sostanze, anziché in termini di molecole (e.g. esprimendo le quantità in moli, si ha un numero di interazioni di  $6.022 \times 10^{23}$  volte minori rispetto al sistema reale, introducendo errori di campionamento tipicamente trascurabili);
- eliminare componenti non reattive (ad esempio molecole che non cambiano di quantità, come i cofattori, o che non alterano la cinetica di altre reazioni);
- eliminare componenti poco reattive (i cui effetti sono trascurabili nel contesto preso in esame);
- eliminare interazioni trascurabili (come inibizioni di entità molto limitata);

- aggregare più reazioni, eliminando tutte le sostanze intermedie e calcolando una cinetica complessiva.

### 2.1.1 Cinetica enzimatica di Michaelis-Menten

La maggior parte degli enzimi è accomunata da una cinetica bifasica descritta genericamente dalle seguenti reazioni:



dove  $E$  indica l'enzima ed  $S$  il substrato da trasformare nel prodotto  $P$  e dove la prima reazione causa la formazione reversibile del complesso enzima-substrato ( $ES$ , con costanti di equilibrio  $k_1$  per la reazione diretta e  $k_{-1}$  per quella inversa) mentre la seconda causa la formazione di prodotto (con costante di equilibrio  $k_2$ ). Come è noto, la soluzione dell'equazione differenziale che descrive la cinetica del sistema, dal punto di vista della concentrazione del complesso  $ES$ , sarà:

$$\frac{d[ES]}{dt} = k_1[E][S] - (k_{-1} + k_2)[ES],$$

essendo la velocità complessiva pari alla differenza tra la velocità di sintesi e quella di degradazione.

Indicando con  $e_0$  la quantità complessiva di enzima nel sistema, sarà vero in ogni momento:  $[E] = e_0 - [ES]$  e sarà possibile ricavare  $[ES]$  allo steady state, eguagliando le velocità di sintesi e degradazione:

$$\begin{aligned} k_1[E][S] &= (k_{-1} + k_2)[ES] \\ [ES] &= \frac{k_1 e_0 [S]}{k_{-1} + k_2 + k_1 [S]}. \end{aligned}$$

La velocità di formazione del prodotto sarà:

$$\frac{d[P]}{dt} = k_2[ES] = \frac{k_1 k_2 e_0 [S]}{k_{-1} + k_2 + k_1 [S]}.$$

Ponendo  $V_{max} = k_2 e_0$  e  $K_M = \frac{k_{-1} + k_2}{k_1}$ , è possibile riscrivere l'equazione come:

$$\frac{d[P]}{dt} = \frac{V_{max}[S]}{K_M + [S]}. \quad (2.2)$$

L'equazione 2.2 prende il nome di *equazione di Michaelis-Menten* e l'andamento della velocità di sintesi del prodotto, in funzione della concentrazione di substrato, è descritto dalla figura 4.1.

Per molti enzimi risulta spesso necessario arricchire la cinetica 2.1 per descrivere più fedelmente il comportamento: un enzima comune può presentare



substrati multipli, che formano complessi con svariati meccanismi, e interazioni con sostanze in grado di modulare la reazione (inibitori e promotori se, rispettivamente, diminuiscono o aumentano la velocità di reazione). Tuttavia, mantenendo costanti le concentrazioni di tutte le sostanze ad eccezione di quella di interesse, si osserva un andamento identico a quello descritto dall'equazione 2.2, dove le informazioni relative alle altre sostanze sono incluse nelle “costanti”  $V_{max}$  e  $K_M$ , che diventano quindi funzioni di tutte le altre concentrazioni e sono dette *costanti apparenti*.

### 2.1.2 Diffusione passiva

Il fenomeno di diffusione di una sostanza è interamente governato dal suo *potenziale chimico* e, allo steady state, vale la *prima legge di Fick*:

$$\mathbf{J} = -D\nabla\varphi,$$

che afferma che il flusso della sostanza varia proporzionalmente all'opposto del gradiente di concentrazione ( $\nabla\varphi$ ) di quella sostanza, con  $D$  *coefficiente di diffusività*.

Nel caso di un ambiente interno, separato dall'esterno da una membrana (o analogamente due compartimenti separati da una membrana), il gradiente può essere approssimato dalla differenza tra le concentrazioni tra l'interno e l'esterno:

$$J = -D([X_{dentro}] - [X_{fuori}]),$$

ma, essendo il flusso la derivata parziale della velocità rispetto alla superficie ( $\mathbf{J} = \frac{\partial \mathbf{v}}{\partial \mathbf{A}}$ ) e ponendo costante (in forma e dimensioni) la superficie attraversata, è possibile riscrivere l'equazione come:

$$v_{ingresso} = k \cdot ([X_{fuori}] - [X_{dentro}]), \quad (2.3)$$

dove  $k$  è una costante, ricavabile sperimentalmente, che aggrega diffusività e superficie della membrana.

## 2.2 Analisi *in silico*

Le informazioni che possono essere inferite dalla semplice osservazione di un fenomeno sono tipicamente insufficienti per permettere una comprensione adeguata del fenomeno stesso. Per aumentarne la comprensibilità, il metodo scientifico si avvale di strumenti detti *modelli*, che rappresentano nel modo più preciso possibile il fenomeno, permettendo la formulazione di ipotesi e la loro verifica sul modello (tramite processi di induzione, deduzione e abduzione), anziché direttamente sul fenomeno (tramite esperimenti).

Un modello “cognitivo” soffre tuttavia di ambiguità, inconsistenze e incompletezze che possono introdurre errori, spesso significativi, sui risultati ottenuti

dal modello, rispetto ai risultati sperimentali; un *modello formale* ovvia all'inconveniente tramite rappresentazioni matematiche ben formate, ovvero non ambigue, consistenti e complete. Poiché la complessità dei fenomeni (e conseguentemente dei modelli che li rappresentano) è spesso elevata, un modello formale deve offrire la possibilità di una risoluzione algoritmica (garantita dai formalismi matematici sottostanti) eseguita da un calcolatore, che diminuisca drasticamente il tempo di analisi ed elimini completamente il rischio di errori (di calcolo, avendo scongiurato quelli di fedeltà applicando le tecniche formali) che potrebbero passare inosservati a un esecutore umano.

Tipicamente un singolo formalismo non è in grado di catturare tutti gli aspetti (comportamentale, prestazionale, temporale e strutturale) di un fenomeno, di conseguenza vengono frequentemente integrati più modelli che descrivono le varie sfaccettature del fenomeno.

Per un fenomeno *reattivo* (ossia in grado di interagire con l'ambiente esterno), un modello potrà descrivere proprietà costitutive (che vengono definite *stati*, ovvero ciò che il fenomeno “è” in un determinato istante) o proprietà interattive (dette *azioni*, ovvero ciò che il sistema “fa” in un determinato istante). Grazie a questo dualismo, è possibile ricondurre ogni modello reattivo a un *grafo* (spesso esteso con informazioni aggiuntive), ovvero una coppia  $\mathcal{G} = (\mathbb{V}, \mathcal{E})$ , dove  $\mathbb{V}$  è un insieme di vertici (che corrispondono agli stati del modello) ed  $\mathcal{E} \subseteq \mathbb{V} \times \mathbb{V}$  rappresenta gli archi (che corrispondono alle azioni del modello), definiti come relazione tra vertici. Ogni grafo possiede una rappresentazione grafica e, in ultima analisi, è possibile ricondurre qualsiasi tipologia di operazione sul grafo (e quindi sul modello che l'ha generato) a una ricerca di determinati percorsi, a partire da uno o più *stati iniziali*.

In figura 2.1 è esemplificato un modello per il dogma centrale della biologia molecolare, dove sia i vertici che gli archi sono arricchiti con etichette (la struttura così arricchita prende il nome di diagramma a stati e transizioni) e dove è stato evidenziato lo stato iniziale in grigio. Il modello cattura la “posizione” dell'informazione genetica negli stati e in che modo possa essere trasferita da una componente all'altra (ma non, ad esempio, informazioni sulle cinetiche o sulla regolazione) e può essere utilizzato per derivare informazioni come: “non è possibile trasferire l'informazione dalla proteina al DNA”, “è possibile, ma non certo, che dal DNA si ottenga, prima o poi, una proteina” e via dicendo. Relativamente alla seconda proposizione si può notare che esistono i cicli infiniti  $DNA \xrightarrow{\text{trascrizione}} RNA \xrightarrow{\text{retrotrascrizione}} DNA$  e  $DNA \xrightarrow{\text{replicazione}} DNA$  che non raggiungono lo stato *Proteina*, motivo per cui viene affermata la possibilità, anziché la certezza.

Un modello che codifica informazioni solo sugli stati è detto *state-based*, se invece codifica solo azioni si dice *action-based*. Le due famiglie di modelli sono del tutto equivalenti, in quanto è possibile applicare le semplici conversioni:

*stato*  $\rightarrow$  *azione* per ogni stato  $s$ , l'azione che lo precede viene definita come “diventa vero  $s$ ”;

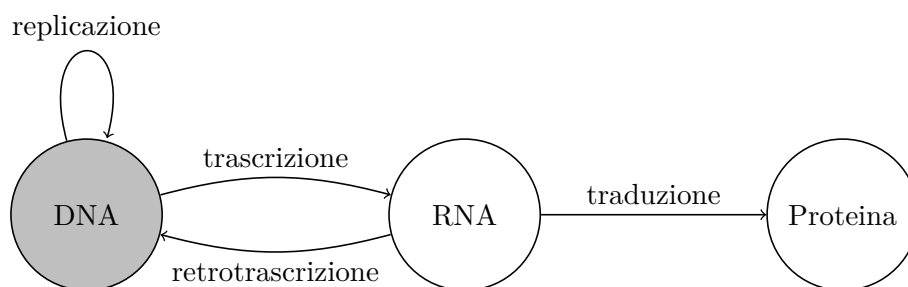


Figura 2.1: Modello a stati e transizioni del dogma centrale della biologia molecolare

*azione*  $\rightarrow$  *stato* per ogni azione  $a$ , lo stato che la precede viene definito come “è possibile eseguire  $a$ ”.

Un'altra classificazione dei formalismi può essere costituita dal meccanismo di risoluzione delle scelte, ovvero in quale stato evolvere tra due o più possibili successori (nei modelli state-based) o quale eseguire tra due o più azioni abilitate (in quelli action-based). I principali meccanismi sono:

- non determinismo non viene fissato alcun criterio di scelta, quindi tutti i possibili percorsi sul grafo verranno analizzati separatamente;
- priorità alle scelte è associato un valore numerico e, per ogni scelta, vengono eliminate quelle con la priorità più bassa, nel caso in cui ci siano due priorità uguali, si procede in maniera non deterministica;
- race policy le scelte alternative competono secondo un criterio (temporale o probabilistico) e viene eseguita solo l'azione più “veloce”.

Da un modello formale è possibile eseguire:

- simulazioni *tracce*, intese come sequenze di stati o azioni che determinano un percorso sul modello, sono generate casualmente a partire dagli stati iniziali (o stabilite dal modellista in modo da raggiungere stati di suo interesse) e si osserva il comportamento del modello man mano che le tracce vengono percorse;
- verifiche vengono definite delle proprietà e viene calcolato se il modello le *soddisfa*, tramite una ricerca esaustiva su tutti gli stati e le azioni.

I due approcci presentano pro e contro che rendono spesso auspicabile l'utilizzo integrato di entrambi. Agendo le simulazioni su tracce, sono efficienti in termini di memoria, al costo di una intrinseca non esaustività e di un tempo di analisi che cresce linearmente con la lunghezza delle tracce. Le verifiche

invece sono corrette e complete rispetto alle proprietà e sono in grado di generare *controesempi* nel caso di proprietà non verificate, al costo di un'estrema avidità in memoria (salvo utilizzare tecniche simboliche) [76].

Poiché la verifica formale risulta memory-intensive, sono spesso necessarie tecniche di modellazione che riducano la dimensione del modello, inficiando il meno possibile sulla fedeltà di quest'ultimo; le più comuni sono:

- astrazione      un grande spazio degli stati è ridotto definendo classi di equivalenza che eliminano informazioni sulle differenze di scarso interesse tra stati;
- composizione    un modello complesso è rappresentato (ed eventualmente semplificato) come un insieme di più modelli semplici che interagiscono.

### 2.2.1 Verifica formale

Le tecniche di verifica formale si dividono in:

- model checking    il modello è descritto con un formalismo riconducibile a un grafo di dimensione finita e le proprietà sono descritte in termini di raggiungibilità di determinati vertici (o sequenze di questi);
- theorem proving    sia il modello che le proprietà sono descritte da formule logiche e la verifica avviene per dimostrazione di teoremi basati su quelle formule.

Gli approcci al model checking tradizionale sono due [22]:

- model checking temporale      il modello è descritto da un sistema di transizioni etichettate e le proprietà sono descritte da formule di logica temporale (che descrivono i percorsi in termini delle relazioni di ordinamento “prima” e “dopo” tra stati);
- model checking basato su automi    sia modello che proprietà sono descritti da automi e la verifica può avvenire tramite: inclusione di linguaggi [37], raffinamento di ordinamenti ed equivalence checking [23].

I due approcci sono equivalenti ed è possibile convertire un modello adatto a un approccio in un modello adatto all'altro [85].

Il *model checking simbolico* [52] ovvia al problema dell'*esplosione dello spazio degli stati*, ovvero il fenomeno per cui la dimensione del modello aumenta

esponenzialmente all'aumentare del numero di componenti del sistema modellato, utilizzando strutture dette *binary decision diagram* [14], che “compattano” il grafo che si otterrebbe con le tecniche tradizionali. Altri approcci per la riduzione del problema dell'esplosione, tramite eliminazione di stati inutili (ovvero astrazione), sono:

- sfruttamento delle informazioni sugli ordinamenti parziali [61];
- riduzione della localizzazione [44];
- minimizzazione semantica [27].

Una tecnica correlata al model checking (e spesso integrata negli stessi tool), in quanto utilizza le stesse strutture per i modelli, è costituita dall'*equivalence checking*. Essa consiste nel verificare che tra due modelli diversi esista una *relazione di equivalenza* comportamentale. Se due modelli esibiscono lo stesso comportamento, allora soddisferanno le stesse proprietà (in un determinato sottoinsieme della logica temporale utilizzata) e ciò consente di coadiuvare il model checking, ad esempio:

- se un modello complesso è equivalente a uno talmente semplice da avere le proprietà banalmente verificate, è possibile bypassare completamente il model checking;
- se un modello grande è equivalente a uno piccolo, è possibile effettuare il model checking su quello piccolo, risparmiando tempo e memoria;
- modificando gli algoritmi di equivalence checking, è possibile *minimizzare* in maniera automatica un modello, ottenendone uno più piccolo equivalente.

Per *caratterizzazione logica* di un'equivalenza comportamentale si intende l'insieme di tutte e sole le formule la cui soddisfacibilità è mantenuta tra i due modelli equivalenti e può essere usata come definizione alternativa dell'equivalenza stessa [9]. Inoltre se il modello è un'algebra di processi (sezione 2.4), è possibile definire degli *assiomi di equivalenza*, che permettono di correlare l'equivalenza comportamentale (semantica) all'equivalenza sintattica (ad esempio, se tra gli assiomi di equivalenza è presente la proprietà distributiva, varrà l'equivalenza  $a.P + a.Q \sim a.(P + Q)$ , senza dover effettuare controlli di natura semantica).

Le equivalenze più comuni sono:

equivalenza a tracce	due modelli sono equivalenti se, partendo dagli stati iniziali, tutte le tracce generate coincidono;
bisimulazione	due modelli sono equivalenti se, partendo dagli stati iniziali, ognuno dei due modelli può simulare passo dopo passo qualsiasi azione eseguita dall'altro;

equivalenza basata su test un test è un modello (di struttura arbitraria) basato sulle stesse azioni dei modelli da verificare e arricchito dall'*azione di successo*, un *esperimento* è un modello costituito dalla *composizione* di un test e del modello da verificare. Si dice che l'esperimento è superato se prima o poi sarà possibile eseguire l'azione di successo, altrimenti l'esperimento è detto fallito. Due modelli sono equivalenti se l'esito degli esperimenti sarà lo stesso, per tutti i test costruibili con le loro azioni.

Il *theorem proving* si basa su sistemi deduttivi da cui ottenere la dimostrazione di un teorema nella forma “modello  $\implies$  proprietà”. Poiché sia il modello che le proprietà sono formule logiche, è possibile operare anche su modelli di dimensione infinita (utilizzando, ad esempio, il principio di induzione). A differenza del model checking, che utilizza modelli molto più complessi, il theorem proving può essere effettuato anche per via manuale e l'ampia varietà di strumenti a disposizione copre un ampio spettro di interattività (da quelli semimanuali a quelli completamente automatizzati).

## 2.3 Catene di Markov

Un *processo stocastico* è un modello che descrive l'andamento di un fenomeno aleatorio in funzione di una variabile indipendente detta *tempo*. Se lo spazio degli stati (ovvero l'insieme di tutte le possibili condizioni in cui può trovarsi il processo) è discreto, il processo prende il nome di *catena*. Se la distribuzione di probabilità del fenomeno è caratterizzata dalla *proprietà di assenza di memoria*, la catena è detta di Markov.

Più formalmente, sia  $X$  la distribuzione di probabilità relativa al tempo di attesa in uno stato, allora la *proprietà di Markov* indica che la probabilità di transitare allo stato successivo dopo un tempo  $t + s$ , condizionata al tempo  $s$  già trascorso, coincide con la probabilità di transitare dopo il tempo  $t$  residuo:

$$\mathcal{P}(X > t + s \mid X > s) = \mathcal{P}(X > t), \quad \forall t, s > 0. \quad (2.4)$$

Le due sole distribuzioni di probabilità che soddisfano la proprietà di Markov sono le distribuzioni geometrica ( $\mathcal{G}_p(n) = p(1 - p)^n$ ) nel caso di tempo discreto ed esponenziale ( $\mathcal{E}_\lambda(t) = \lambda e^{-\lambda t}$ ) nel caso di tempo continuo. Per definizione di probabilità condizionata, si ha che l'equazione 2.4 può essere riscritta come:

$$\mathcal{P}(X > t + s) = \mathcal{P}(X > s)\mathcal{P}(X > t);$$

per la distribuzione esponenziale vale:

$$\mathcal{P}(X > t) = \int_t^\infty \lambda e^{-\lambda x} dx = e^{-\frac{t}{\lambda}},$$

ma allora, per sostituzione, si ha:

$$\begin{aligned}\mathcal{P}(X > t + s) &= e^{-\frac{t+s}{\lambda}} \\ &= e^{-\frac{t}{\lambda}} e^{-\frac{s}{\lambda}} \\ &= \mathcal{P}(X > t) \mathcal{P}(X > s),\end{aligned}$$

dimostrando il caso per la distribuzione esponenziale [82]; con un ragionamento analogo si dimostra il caso della distribuzione geometrica [83].

Una catena di Markov è detta a tempo discreto se il tempo viene osservato in *passi* (di conseguenza tutte le catene di Markov a tempo discreto saranno governate da distribuzioni geometriche), se invece il tempo viene osservato con continuità, viene detta *a tempo continuo* (e sarà conseguentemente governata da distribuzioni esponenziali).

Di seguito ci si concentrerà sulle catene di Markov a tempo continuo (CTMC), più adatte alla modellazione di reti metaboliche: dalla teoria degli urti, accennata nella sezione 2.1, conseguono infatti l'assenza di memoria di una reazione chimica (in quanto la possibilità o meno che avvenga un urto efficace tra due molecole di substrato non dipende da quanti urti, efficaci o meno, hanno già subito le molecole) e la natura continua del processo (lo spostamento delle molecole, e conseguentemente gli urti, non avvengono “a scatti”).

Formalmente una CTMC (secondo il formalismo adattato al model checking) è una tupla  $\mathcal{M} = (\mathbb{S}, \mathbf{R}, \mathcal{P}_{init}, \mathbb{AP}, \mathcal{L})$ , dove:

- $\mathbb{S}$  è lo spazio degli stati;
- $\mathbf{R}$  è la matrice delle transizioni  $\mathbf{R} : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}_{\geq 0}$ , una struttura che indica, per ogni coppia di stati  $(s_1, s_2)$ , con che rate (esponenziale) avviene la transizione  $s_1 \rightarrow s_2$  (poiché il rate è l'inverso del tempo medio di attesa, una transizione che non può avvenire avrà rate pari a 0);
- $\mathcal{P}_{init}$  è la distribuzione di probabilità iniziale, che indica da quali stati (e con che probabilità) dovranno originarsi i cammini sulla catena;
- $\mathbb{AP}$  è un insieme di proposizioni atomiche che rende più leggibile lo spazio degli stati;
- $\mathcal{L}$  è la funzione di etichettamento degli stati  $\mathcal{L} : \mathbb{S} \mapsto 2^{\mathbb{AP}}$ , che associa a ogni stato zero o più etichette (ad esempio, associando allo stato  $s_1$  le etichette “X = 10  $\mu$ mol”, “Y non ha ancora reagito”, “Z esaurito” e “soluzione di colore azzurro”), rendendo più facilmente interpretabile la catena.

Una catena di Markov si presta bene sia a tecniche simulative, basandosi sull'osservazione della matrice  $\mathbf{R}$ , che a tecniche di verifica, basandosi su formule di logica costruite su  $\mathbb{AP}$ .

Ulteriore potere espressivo è dato dall'estensione della catena tramite strutture dette *reward*, che rappresentano dei punteggi dal significato arbitrario (e.g. costi, tempi, quantità). Esistono due tipi di reward:

reward istantaneo punteggio applicato una volta sola, al passaggio da uno stato al successivo;

reward cumulativo punteggio moltiplicato al tempo di soggiorno nello stato a cui è associato.

La tupla  $\mathcal{M}$  viene estesa con la matrice  $\iota : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$  che mappa ogni coppia di stati in un reward istantaneo e/o la funzione  $\rho : \mathbb{S} \mapsto \mathbb{R}$  che associa un reward stazionario ad ogni stato.

La tupla estesa  $\mathcal{M} = (\mathbb{S}, \mathbf{R}, \mathcal{P}_{init}, \iota, \rho, \mathbb{AP}, \mathcal{L})$  prende il nome di Markov reward model (MRM) [3].

È possibile rappresentare una catena di Markov tramite un diagramma a stati e transizioni, dove gli stati sono etichettati da  $\mathcal{L}$  e le transizioni da  $\mathbf{R}$ . Come è intuibile, questa rappresentazione permette di effettuare sia operazioni di simulazione che di model checking, in termini di cammini sulla catena. In figura 2.2 è rappresentata una catena con funzione di etichettamento:

$$\mathcal{L} : \{s_0 \mapsto A, s_1 \mapsto B, s_2 \mapsto C\}$$

e matrice delle transizioni:

$$\mathbf{R} = \begin{matrix} & \begin{matrix} 0 & 0.5 & 0 \end{matrix} \\ \begin{matrix} 100 \\ 3.14 \end{matrix} & \begin{matrix} 2 & 1. \\ 0 & 0 \end{matrix} \end{matrix}$$

### 2.3.1 Linguaggio di modellazione di PRISM

PRISM [45] è un model checker per modelli probabilistici basati su catene di Markov, che utilizza un formalismo state-based per semplificare la creazione dei modelli. I formalismi supportati sono:

catene di Markov a tempo discreto (DTMC) la risoluzione delle scelte è probabilistica;

processi decisionali di Markov (MDP) la risoluzione può avvenire sia non deterministicamente che probabilisticamente;

catene di Markov a tempo continuo (CTMC) la risoluzione avviene tramite race policy su transizioni temporizzate esponenzialmente;



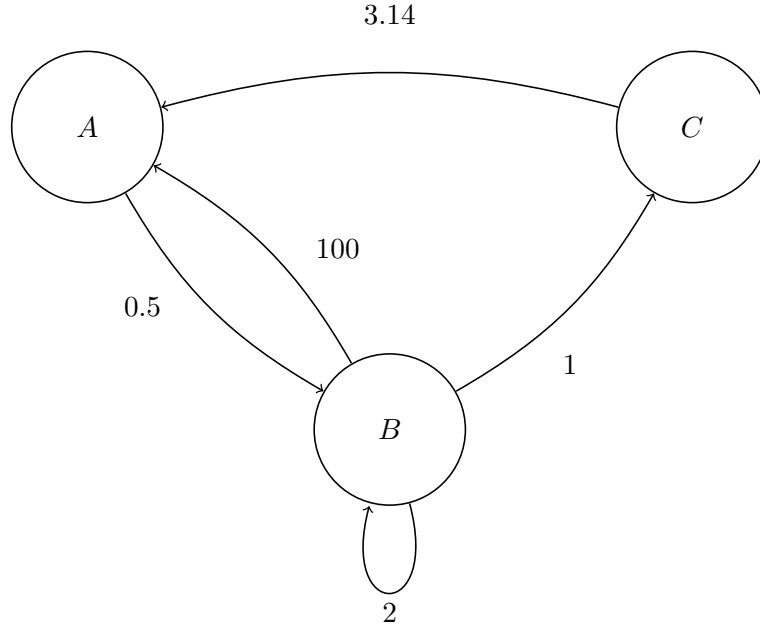


Figura 2.2: Rappresentazione grafica di una catena di Markov a tempo continuo

automi probabilistici temporizzati (PTA)

la risoluzione può avvenire sia non deterministicamente che con race policy.

Un modello PRISM è costituito da un insieme di *moduli* – ognuno dotato di un proprio *stato interno* e di un comportamento – che possono essere considerati delle piccole catene indipendenti.

All'interno di un modulo, lo stato è rappresentato da *variabili* che possono assumere valori appartenenti a un intervallo limitato e viene aggiornato ad ogni transizione. Le transizioni possono essere *etichettate* da un'azione e *abilitate* da un'espressione di guardia. Il comportamento di un modulo è descritto da una serie di *comandi* nella forma:

$$[azione](guardia) \rightarrow prob_1 : agg_1 + prob_2 : agg_2 + \dots + prob_n : agg_n,$$

dove *azione* è l'etichetta da associare alla transizione, *guardia* è un predicato che attiva la transizione solo negli stati in cui viene soddisfatto,  $prob_n$  è una probabilità (per DTMC e MDP) o un rate (per CTMC e PTA) e  $agg_n$  è un'istruzione di aggiornamento. La scelta tra più aggiornamenti (separati dall'operatore +) avviene tramite probabilità o race policy. Le istruzioni di aggiornamento sono nella forma:

$$(x' = expr_x) \& (y' = expr_y) \& \dots \& (z' = expr_z),$$

dove ogni variabile (a sinistra) è aggiornata valutando l'espressione a destra dell'uguale. Se delle variabili vengono omesse, si assume che il loro valore non cambi.

La catena di Markov a tempo continuo di figura 2.2 può essere rappresentata da un singolo modulo contenente una sola variabile e con il comportamento descritto dai seguenti comandi:

$$\begin{aligned} &[](\text{stato} = A) \rightarrow 0.5 : \text{stato}' = B \\ &[](\text{stato} = B) \rightarrow 100 : \text{stato}' = A + 2 : \text{stato}' = B + 1 : \text{stato}' = C \\ &[](\text{stato} = C) \rightarrow 3.14 : \text{stato}' = A \end{aligned}$$

Il modello del sistema è ottenuto per composizione dei singoli moduli, tramite *sincronizzazione* su tutte le azioni aventi lo stesso nome. PRISM presenta quattro motori computazionali (applicati a tutti i modelli a eccezione degli automi probabilistici temporizzati) per il *model checking esatto*, che utilizzano rappresentazioni diverse del modello per ottimizzare le prestazioni nella maggior parte dei casi d'uso:

- explicit    il modello è rappresentato con una matrice delle transizioni e il model checking avviene per via tradizionale, con prestazioni temporali elevate, ma occupazione in memoria altrettanto elevata: ciò lo rende inadeguato per modelli di grosse dimensioni;
- sparse     il modello è rappresentato con una matrice sparsa, risparmiando enormi quantità di memoria se il modello contiene molti stati non raggiungibili (in caso contrario l'occupazione in memoria è peggiore della matrice esplicita), anche in questo caso il model checking è di tipo tradizionale;
- MTBDD    la catena viene rappresentata da un multi terminal binary decision diagram (sezione 2.7.1), risparmiando molta memoria se il modello presenta un elevato livello di regolarità (ad esempio molte transizioni che convergono in uno stesso stato con lo stesso rate) al costo di prestazioni temporali ridotte; per modelli altamente irregolari l'occupazione in memoria potrebbe essere di molto maggiore rispetto alla matrice esplicita;
- hybrid    la catena viene rappresentata sia da un MTBDD che da una matrice sparsa e il model checking è di tipo ibrido [47], nella maggior parte dei casi offre il miglior compromesso tra occupazione in memoria e tempo di calcolo.

PRISM è inoltre in grado di effettuare *model checking statistico* [46], che permette di stimare l'esito di una proprietà a partire da *simulazioni* sul modello, nel caso in cui le tecniche esatte non siano attuabili.

PRISM è stato impiegato con successo in casi di studio in ambito biologico come i seguenti:

- controllo del ciclo cellulare eucariota [64];
- ciclo circadiano astratto [65];
- via FGF [38];
- DNA computing [24];
- fusione dei virus dell'influenza [26];
- cinetica dei ribosomi [11];
- segnalazione dei linfociti T [57];
- pacemaker cardiaci [17].

## 2.4 Algebre di processi

Le algebre di processi sono un formalismo action-based che estremizza il concetto di composizionalità definendo un modello come un'espressione costituita da operazioni (che determinano interazioni) tra entità dette *processi*: così come l'algebra dei numeri reali permette la costruzione di espressioni quali  $3 + 2 \cdot 1.4$ , un'algebra di processi permette di costruire modelli del tipo  $a.(b.\underline{0} + c.\underline{0})$ .

Un processo è un'entità che può compiere *azioni* (evolvendo in altri processi) o rimanere bloccata (e in tal caso si indica come *processo nullo*, rappresentato da  $\underline{0}$ ) e le operazioni definite dall'algebra permettono di descrivere tutti i possibili comportamenti (intesi come tracce o punti di scelta) tramite un'espressione che utilizzi  $\underline{0}$  o costanti (ad esempio  $P$ ) che identifichino gli altri processi. Le operazioni definite da un'algebra di processi generica sono:

prefisso d'azione	$a.P$ : viene eseguita l'azione $a$ e si evolve nel processo $P$ ;
scelta	$P_1 + P_2$ : si evolve in $P_1$ oppure $P_2$ , secondo un meccanismo di risoluzione (come quelli esemplificati nella sezione 2.2) dipendente dall'algebra di processi. Nel caso di meccanismi diversi da quello non deterministico, è necessario arricchire le azioni con informazioni aggiuntive;
composizione parallela	$P_1 \underset{\mathbb{A}}{\bowtie} P_2$ : $P_1$ e $P_2$ evolvono parallelamente, indipendentemente tra loro per ogni azione non contenuta nell'insieme $\mathbb{A}$ , ma aspettando ognuno la controparte (eventualmente per un tempo infinito) per eseguire <i>contemporaneamente</i> le azioni contenute in $\mathbb{A}$ ;

hiding	$P/\mathbb{A}$ : tutte le azioni contenute in $\mathbb{A}$ diventano <i>invisibili</i> (indicate con $\tau$ ). Questo operatore è particolarmente utile per l'equivalence checking basato su <i>equivalenze deboli</i> (varianti delle equivalenze comportamentali che consentono di ignorare le azioni invisibili, nel caso della bisimulazione debole, ad esempio, ogni processo dovrà simulare passo dopo passo solo le azioni visibili della controparte, potendo ignorare tutte quelle invisibili);
restrizione	$P \setminus \mathbb{A}$ : nessuna delle azioni contenute nell'insieme $\mathbb{A}$ può essere eseguita;
ricorsione	il processo evolve in sé stesso (tipicamente dopo aver eseguito alcune azioni), la notazione fa utilizzo delle costanti di processo e di equazioni di definizione ( $P = \text{comportamento}$ , dove $P$ compare all'interno del comportamento stesso).

Un'equazione di process algebra che può generare il diagramma a stati e transizioni della figura 2.1 è la seguente:

$$P = \text{replicazione}.P + \text{trascrizione}.\text{(retrotrascrizione}.P + \text{traduzione}.0),$$

tuttavia risulta più agevole definire un sistema di equazioni con un numero maggiore di costanti (e con identificatori più descrittivi) nel seguente modo:

$$\begin{aligned} DNA &= \text{replicazione}.DNA + \text{trascrizione}.RNA \\ RNA &= \text{retrotrascrizione}.DNA + \text{traduzione}.PROTEINA. \end{aligned}$$

Dal sistema così definito risulta più semplice ricavare il diagramma a stati e transizioni, essendo sufficiente (in questo caso in cui manca l'operatore di sincronizzazione) creare uno stato per ogni processo ed etichettare le transizioni uscenti con le azioni abilitate in ognuno di essi.

Il contenuto informativo delle azioni, che determina il meccanismo di risoluzione, varia per ogni algebra di processi. I modi più comuni per etichettare le transizioni sono:

<i>azione</i>	associare soltanto un'etichetta (eventualmente $\tau$ ) alla transizione, per algebre di processi <i>non deterministiche</i> ;
$\langle \text{azione}, \text{priorità} \rangle$	per algebre di processi <i>prioritarie</i> ;
$\langle \text{azione}, \text{tempo} \rangle$	per algebre di processi <i>temporizzate deterministicamente</i> ;
$\langle \text{azione}, \text{rate} \rangle$	per algebre di processi <i>stocastiche</i> (dove il <i>rate</i> indica il parametro da associare a una distribuzione di probabilità, tipicamente dotata di proprietà di Markov, usata per risolvere la race policy).

Tra le algebre di processi stocastiche, spiccano quelle temporizzate esponenzialmente (tutte le azioni hanno distribuzione  $\mathcal{E}(t) = \lambda e^{-\lambda t}$ ), dove la race policy è esprimibile (nel caso in cui i due rate non coincidano) tramite:

$$\begin{aligned} \langle a, \lambda \rangle.P_1 + \langle b, \mu \rangle.P_2 &\xrightarrow{a, \lambda} P_1 \text{ con probabilità } \frac{\lambda}{\lambda + \mu} \\ \langle a, \lambda \rangle.P_1 + \langle b, \mu \rangle.P_2 &\xrightarrow{b, \mu} P_2 \text{ con probabilità } \frac{\mu}{\lambda + \mu}. \end{aligned}$$

Le algebre temporizzate esponenzialmente sono equivalenti a un diagramma a stati e transizioni esteso da informazioni sui rate di transizione, ovvero da catene di Markov a tempo continuo etichettate da azioni (ACTMC).

### 2.4.1 Bio-PEPA

Bio-PEPA [20] è un'algebra di processi basata su Performance Evaluation Process Algebra (PEPA) [62] (un'algebra temporizzata esponenzialmente), alla quale aggiunge informazioni di tipo *quantitativo* sui processi, finalizzate a semplificare la modellazione di sistemi biologici.

Da un unico modello Bio-PEPA sono ottenibili modelli in altri formalismi che descrivono più aspetti del sistema modellato:

- un modello PRISM, che rappresenta una CTMC e una serie di formule CSL preformate, permette di effettuare model checking e simulazioni;
- un modello basato su equazioni differenziali ordinarie permette di effettuare analisi sulla cinetica;
- un modello StockKit [72] permette di compilare un simulatore efficiente in linguaggio C.

I risultati delle analisi sui vari modelli vengono integrati automaticamente in un unico rapporto, contenente il modello Bio-PEPA e vari grafici ottenuti dalle analisi.

Alcuni casi studio in cui è stato applicato Bio-PEPA sono:

- via NF- $\kappa$  B [18];
- ciclo circadiano di *Neurospora crassa* [1];
- via gp130 JAK STAT [34];
- via cAMP/PKA/MAPK [19];
- modelli epidemiologici per l'influenza aviaria [21];
- ciclo circadiano di *Ostreococcus tauri* [2].

Un modello Bio-PEPA è composto da tre sezioni:

- quantità iniziali e costanti, definite in una matrice memorizzata in un file .csv: ogni riga della matrice costituisce un'istanza dello stesso modello con valori diversi, agevolando l'esecuzione di più *esperimenti* sullo stesso modello;
- cinetiche e contatori, raggruppati nella sezione iniziale del file .biopepa;
- processi veri e propri, raggruppati nella sezione finale del file .biopepa.

Ogni azione è temporizzata esponenzialmente e il rate viene calcolato tramite espressioni (dette *cinetiche*, anche se non vengono necessariamente utilizzate a tale scopo), che utilizzano la quantità corrente ed eventuali costanti. Ciò permette, ad esempio, di modellare l'andamento di una reazione, variandone il rate in funzione della concentrazione di substrati. Un *contatore* è un costruito Bio-PEPA che permette di rappresentare grandezze derivate dal modello (ad esempio la carica energetica, come rapporto tra i nucleotidi trifosfati e i nucleotidi totali). Sia cinetiche che contatori utilizzano la sintassi:

$$\begin{aligned} \text{azione} &= [\text{espressione}]; \\ \text{CONTATORE} &= [\text{espressione}]; \end{aligned}$$

dove *espressione* può contenere qualsiasi identificatore definito nel file delle quantità, costante numerica e operatore aritmetico.

In Bio-PEPA, la definizione dei processi avviene secondo un approccio *bottom-up*, definendo prima il comportamento dei singoli elementi del sistema e poi componendoli (tipicamente tramite operatore di composizione parallela) in un unico processo rappresentante l'intero modello. Per tenere conto delle quantità, le azioni sono modificate nella coppia (*azione*, *stechiometria*) (essendo il rate definito implicitamente nella sezione relativa alle cinetiche) e l'operatore prefisso d'azione è modificato nelle seguenti varianti:

reagente	$(\text{azione}, \text{stechiometria}) \downarrow P$ : se le quantità sono superiori a <i>stechiometria</i> , il processo diminuisce di tale quantità ed evolve in <i>P</i> ;
prodotto	$(\text{azione}, \text{stechiometria}) \uparrow P$ : il processo aumenta di <i>stechiometria</i> unità (eventualmente saturandosi fino a un massimo) ed evolve in <i>P</i> ;
attivatore	$(\text{azione}, \text{stechiometria}) \oplus P$ : il processo abilita <i>azione</i> se è presente in quantità superiori a <i>stechiometria</i> , poi evolve in <i>P</i> senza alterare di quantità;
inattivatore	$(\text{azione}, \text{stechiometria}) \ominus P$ : il processo disabilita <i>azione</i> se è presente in quantità superiori a <i>stechiometria</i> , poi evolve in <i>P</i> senza alterare di quantità;

modulatore generico  $(azione, stochiometria) \odot P$ : il processo evolve in  $P$  senza alterare di quantità, questo operatore coincide con il prefisso d'azione tradizionale.

Un modello esemplificativo, tratto da [20] (a cui si rimanda per i risultati ottenuti dall'analisi sul modello), sulla regolazione genica a feedback negativo basata su dimerizzazione in *E. coli* [15] è il seguente:

$$\begin{aligned}\alpha_1 &= \left[ \frac{\nu}{K_M + P2} \right]; \\ \alpha_2 &= [fMA(k_2)]; \\ \alpha_3 &= [fMA(k_3)]; \\ \alpha_4 &= [fMA(k_4)]; \\ \alpha_5 &= [fMA(k_5)]; \\ \alpha_{5i} &= [fMA(k_{5i})];\end{aligned}$$

$$\begin{aligned}M &= (\alpha_1, 1) \uparrow M + (\alpha_2, 1) \oplus M + (\alpha_3, 1) \downarrow M; \\ P &= (\alpha_2, 1) \uparrow P + (\alpha_4, 1) \downarrow P + (\alpha_5, 2) \downarrow P + (\alpha_{5i}, 2) \uparrow P; \\ P2 &= (\alpha_1, 1) \ominus P2 + (\alpha_5, 1) \uparrow P2 + (\alpha_{5i}, 1) \downarrow P2;\end{aligned}$$

$$M \bowtie_{\alpha_2} P \bowtie_{\{\alpha_5, \alpha_{5i}\}} P2. \quad (2.5)$$

M	0 nM
P	0 nM
P2	0 nM
$\nu$	$2.19 \text{ s}^{-1}$
$K_M$	356 nM
$k_2$	$0.043 \text{ s}^{-1}$
$k_3$	$0.0039 \text{ s}^{-1}$
$k_4$	$0.0007 \text{ s}^{-1}$
$k_5$	$0.025 \text{ s}^{-1} \text{ nM}^{-1}$
$k_{5i}$	$0.5 \text{ s}^{-1}$

Tabella 2.1: Parametri del modello sulla regolazione genica basata su dimerizzazione di *E. coli*

Il modello è composto da sei reazioni:

$\alpha_1$  trascrizione dell'mRNA ( $M$ ), inibita dal dimero di proteina ( $P2$ );

$\alpha_2$  traduzione dei monomeri di proteina ( $P$ ), promossa dall'mRNA;

$\alpha_3$  degradazione dell'mRNA;

$\alpha_4$  degradazione della proteina;

$\alpha_5$  dimerizzazione della proteina;

$\alpha_{5i}$  rottura del dimero di proteina.

Ad eccezione di  $\alpha_1$ , tutte seguono la legge di azione di massa ( $fMA$ ). Osservando i coefficienti stechiometrici, è evidente che siano tutte del primo ordine ( $fMA(k) = k \frac{[prodotti]}{[reagenti]}$ ), esclusa  $\alpha_5$ , del secondo ordine ( $fMA(k) = k \frac{[prodotti]}{[reagenti]^2}$ ).

## 2.5 Logiche temporali

Le logiche temporali sono un'estensione della logica dei predicati, basate su una relazione d'ordine di tipo "prima-dopo"; grazie a tale relazione risultano uno strumento particolarmente efficace per descrivere *percorsi* su un modello e, di conseguenza, proprietà verificabili tramite model checking.

Partendo da *predicati* che possono essere veri o falsi (atomici), è possibile costruirne di nuovi combinandoli con *connettivi logici*. I connettivi più comuni sono:

$\neg\phi$	vero solo se $\phi$ è falso;
$\phi_1 \wedge \phi_2$	vero se sia $\phi_1$ che $\phi_2$ sono veri;
$\phi_1 \vee \phi_2$	vero se almeno uno tra $\phi_1$ e $\phi_2$ è vero;
$\phi_1 \rightarrow \phi_2$	vero se $\phi_1$ e $\phi_2$ sono entrambi veri, oppure se $\phi_1$ è falso;
$\phi_1 \leftrightarrow \phi_2$	vero se $\phi_1$ e $\phi_2$ sono entrambi veri oppure entrambi falsi.

Aggiungendo delle variabili, è possibile *quantificare* i predicati:

$\forall x(\phi)$  vero se  $\phi$  è vero per tutti i possibili valori di  $x$ ;

$\exists x(\phi)$  vero se  $\phi$  è vero per almeno un valore di  $x$ .

Gli *operatori temporali* aggiungono la possibilità di indicare quando dovrà essere vero un predicato, i più comuni (rivolti al futuro) sono:

$\mathcal{X}\phi$  vero se  $\phi$  sarà vero nell'immediato futuro;

$\mathcal{F}\phi$  vero se prima o poi  $\phi$  sarà vero;

$\mathcal{G}\phi$  vero se  $\phi$  è sempre vero;



$\phi_1 \mathcal{U} \phi_2$  vero se  $\phi_1$  è vero finché non inizierà a valere  $\phi_2$ ;

$\phi_1 \mathcal{R} \phi_2$  vero se  $\phi_2$  è vero finché  $\phi_1$  rimane vero.

È possibile applicare le formule di una logica temporale a un modello a stati e transizioni (ad esempio una catena di Markov a tempo continuo), facendo coincidere l'insieme di predicati con l'insieme di etichette e definendo un'unica variabile relativa a un generico percorso  $\omega$ . Sulla catena di figura 2.2 è, ad esempio, possibile definire le proprietà  $A$  (“lo stato corrente è  $A$ ”),  $A \wedge \mathcal{X} B$  (“lo stato corrente è  $A$  e il prossimo sarà  $B$ ”) o  $\exists \omega(BUC)$  (compattato in  $\exists(BUC)$ , “esiste almeno un percorso in cui  $B$  è vero finché non inizierà a valere  $C$ ”).

La logica così costruita prende il nome di CTL\* ed è molto espressiva, non imponendo vincoli su come combinare gli operatori: a tale espressività corrisponde il fatto che il problema della *soddisfacibilità* (data una formula e un modello, dire se è vera o falsa per quel modello) sia *intrattabile* (classe di complessità  $2 - EXPTIME$  [86], sebbene in passato sia stato considerato addirittura *indecidibile* [28]).

Imponendo dei vincoli su una logica temporale, è possibile creare delle logiche meno espressive, ma di cui è possibile calcolare la soddisfacibilità, che si possono classificare in:

linear time      è possibile ragionare solo su un percorso alla volta, quindi i quantificatori ( $\forall$  e  $\exists$ ) sono eliminati; la logica linear time ottenuta a partire da CTL\* prende il nome di *linear temporal logic* (LTL);

branching time      è possibile ragionare solo sulle scelte disponibili, quindi gli operatori temporali ( $\mathcal{X}$ ,  $\mathcal{F}$ ,  $\mathcal{G}$ ,  $\mathcal{U}$  e  $\mathcal{R}$ ) devono essere sempre quantificati (“tra le scelte a disposizione, almeno una oppure tutte verificheranno una certa condizione”); la logica branching time ottenuta a partire da CTL\* prende il nome di *computation tree logic* (CTL, senza asterisco).

Probabilistic CTL (PCTL) [36] estende CTL con gli operatori:

$\mathbb{P}$  operatore di probabilità, vero se la probabilità che sia verificata la sottoformula a cui è applicato è limitata da un certo intervallo, considerando le possibili scelte;

$\mathbb{S}$  operatore di probabilità allo steady state, come  $\mathbb{P}$ , ma allo stato stazionario.

I due operatori possono essere usati sia per ottenere valori numerici, nella forma  $\mathbb{P}_{=?}[\phi]$  (“quale è la probabilità che  $\phi$  sia vera?”), che per ottenere valori di verità, nella forma  $\mathbb{P}_{[a,b]}[\phi]$  (“la probabilità che  $\phi$  sia vera è compresa tra  $a$  e  $b$ ?”). In PCTL i quantificatori risultano superflui, in quanto valgono le equivalenze  $\forall(\phi) = \mathbb{P}_{=1}[\phi]$  e  $\exists(\phi) = \mathbb{P}_{>0}[\phi]$ .

### 2.5.1 Continuous Stochastic Logic

Passando da un modello probabilistico a uno temporizzato esponenzialmente, è necessario poter catturare anche aspetti relativi al tempo trascorso. CSL [7] modifica gli operatori temporali di PCTL rendendoli *time-bounded*, ovvero imponendo un limite temporale entro cui devono essere verificati.

Gli operatori time-bounded si indicano con  $\mathcal{OP}^{[a,b]}$ , dove  $\mathcal{OP}$  è un'operatore temporale e  $[a,b]$  è l'intervallo temporale entro quale deve valere la sottoformula. È possibile ricavare gli operatori temporali illimitati (tradizionali) osservando che  $\mathcal{OP} = \mathcal{OP}^{<\infty}$ , quindi CSL è più espressivo di PCTL.

Un esempio di proprietà CSL da verificare sul modello 2.5 (dove, a seguito della conversione, il processo  $P$  corrisponde alla componente  $\_P$ ) può essere:

$$\mathbb{P}_{=?}[\mathcal{G}^{\leq 0.5} \_P2 > 10]$$

“Quale è la probabilità che il dimero di proteina sia *sempre* in quantità maggiore di 10 unità, nelle prime 0.5 unità di tempo?”

È importante notare che ciò che avviene fuori dall'intervallo in cui sono limitati gli operatori viene *ignorato* e non deve necessariamente essere falso (nell'esempio il calcolo della probabilità non è influenzato dal fatto che  $\_P2 > 10$  sia vero o meno dal tempo 0.51 al tempo 10).

Se il modello è arricchito da reward, è possibile estendere CSL con l'operatore  $\mathbb{R}$ , che permette di misurare il valore dei reward accumulati lungo un percorso: la logica così arricchita prende il nome di *continuous stochastic reward logic* (CSRL) [8].

PRISM definisce cinque varianti per l'operatore  $\mathbb{R}$  [66]:

$\mathbb{R}_{\{rew\}}[\mathcal{F}\phi]$  accumula i reward, definiti secondo la struttura *rew*, finché il predicato  $\phi$  non è verificato;

$\mathbb{R}_{\{rew\}}[\mathcal{C} \leq T]$  accumula i reward, definiti secondo *rew*, fino al tempo  $T$ ;

$\mathbb{R}_{\{rew\}}[\mathcal{C}]$  accumula i reward, definiti secondo *rew*, fino a un tempo infinito;

$\mathbb{R}_{\{rew\}}[\mathcal{I} = T]$  restituisce i reward, definiti secondo *rew*, all'istante  $T$ ;

$\mathbb{R}_{\{rew\}}[\mathcal{S}]$  restituisce i reward, definiti secondo *rew*, allo stato stazionario.

Come  $\mathbb{P}$  ed  $\mathbb{S}$ , anche  $\mathbb{R}$  può essere usato nelle varianti  $\mathbb{R}_{=?}$  e  $\mathbb{R}_{[a,b]}$ , permettendo il calcolo di un valore o la verifica che il valore sia entro un intervallo.

## 2.6 Simulazioni su catene di Markov a tempo continuo

In una CTMC, la matrice delle transizioni  $\mathbf{R}$  può anche essere vista come il prodotto tra la probabilità di transizione e il vettore dei tempi medi di

soggiorno ( $\mathbf{R}_{i,j} = \mathbf{P}_{i,j} \cdot \mathbf{r}_i$ ): astraendo da  $\mathbf{r}$  si ottiene una catena di Markov a tempo discreto, detta *embedded Markov chain* (EMC) [77].

L'algoritmo di Gillespie [32] (applicato inizialmente alla simulazione di un sistema di reazioni chimiche e successivamente generalizzato alle catene di Markov a tempo continuo) costituisce una procedura numerica esatta per la simulazione di eventi stocastici, basandosi su tecniche Monte Carlo.

Partendo da uno stato iniziale, estratto casualmente dalla CTMC secondo la distribuzione  $\mathcal{P}_{init}$ , si genera iterativamente un percorso, finché non si raggiunge un tempo di terminazione  $T$ ; ad ogni istante, se lo stato corrente è  $s_k$ , il tempo di soggiorno viene estratto casualmente secondo una distribuzione esponenziale con rate  $\mathbf{r}_k$  e lo stato successivo è scelto estraendo una variabile aleatoria con distribuzione pari alla riga della matrice delle probabilità  $\mathbf{P}_{k,-}$ . Qualora sia necessario ottenere valori esatti, la media di più percorsi avrà un andamento convergente a quello del fenomeno modellato.

**Data:**  $\mathcal{M} = (\mathbb{S}, \mathbf{P}, \mathbf{r}, \mathcal{P}_{init}), T$

**Result:** Percorso sulla catena fino al tempo  $T$

$s_0 \leftarrow$  stato iniziale estratto da  $\mathcal{P}_{init}$ ;

$k \leftarrow 0$ ;

$t \leftarrow 0$ ;

**while**  $t < T$  **do**

$P_{s_k} \leftarrow$  riga di  $\mathbf{P}$  corrispondente allo stato  $s_k$ ;

$s_{k+1}$  estratto casualmente secondo la distribuzione  $P_{s_k}$ ;

$t_k$  estratto casualmente secondo una distribuzione esponenziale con

        rate  $\mathbf{r}_{s_k}$ ;

$t \leftarrow t + t_k$ ;

$k \leftarrow k + 1$ ;

**end**

**return**  $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots \xrightarrow{t_{n-1}} s_n$ .

**Algorithm 1:** Algoritmo di Gillespie per la simulazione di un percorso su una CTMC fino al tempo  $T$

## 2.7 Model checking di catene di Markov a tempo continuo

CSL è formalmente definito da tutte le formule costruibili a partire dalla seguente grammatica:

$$\phi ::= \top \mid \text{atomo} \mid \phi \wedge \phi \mid \neg \phi \mid \mathbb{P}_{\bowtie p}[\psi] \mid \mathbb{S}_{\bowtie p}[\phi] \quad (2.6)$$

$$\psi ::= \mathcal{X}\phi \mid \phi \mathcal{U} \phi \mid \phi \mathcal{U}^{\leq t} \phi, \quad (2.7)$$

dove la produzione 2.6 viene espansa in una *formula di stato* e 2.7 in una *formula di percorso*. L'insieme di formule ottenuto è *funzionalmente completo*, è quindi possibile ottenere tutti gli altri operatori a partire da quelli già definiti

(ad esempio  $\phi_1 \vee \phi_2 = \neg(\neg\phi_1 \wedge \neg\phi_2)$ , oppure  $\mathcal{F}(\phi) = \top \mathcal{U} \phi$ ). Sebbene valga l'uguaglianza  $\phi_1 \mathcal{U} \phi_2 = \phi_1 \mathcal{U}^{<\infty} \phi_2$ , vengono indicate entrambe le varianti dell'operatore, essendo, all'atto pratico, conveniente utilizzare due algoritmi di model checking diversi.

La semantica associata alle formule di stato è la seguente:

$$s \models \top \quad \forall s \in \mathbb{S} \quad (2.8)$$

$$s \models \text{atomo} \iff \text{atomo} \in \mathcal{L}(s) \quad (2.9)$$

$$s \models \phi_1 \wedge \phi_2 \iff s \models \phi_1 \wedge s \models \phi_2 \quad (2.10)$$

$$s \models \neg\phi \iff s \not\models \phi \quad (2.11)$$

$$s \models \mathbb{P}_{\bowtie p}[\psi] \iff \text{Prob}_s(\{\omega \in \text{Path}_s \mid \omega \models \psi\}) \bowtie p \quad (2.12)$$

$$s \models \mathbb{S}_{\bowtie p}[\phi] \iff \sum_{s' \models \phi} \pi_s(s') \bowtie p, \quad (2.13)$$

e quella associata alle formule di percorso risulta:

$$\omega \models \mathcal{X}\phi \iff \exists \omega(1) \wedge \omega(1) \models \phi \quad (2.14)$$

$$\omega \models \phi_1 \mathcal{U} \phi_2 \iff \exists k \geq 0, \omega(k) \models \phi_2 \wedge \forall j < k, \omega(j) \models \phi_1 \quad (2.15)$$

$$\omega \models \phi_1 \mathcal{U}^{\leq t} \phi_2 \iff \exists x \in [0; t], \omega @ x \models \phi_2 \wedge \forall y \in [0; x), \omega @ y \models \phi_1. \quad (2.16)$$

Le regole 2.8–2.11 definiscono predicati e connettivi logici con il loro significato tradizionale. La regola 2.12 asserisce che  $\mathbb{P}_{\bowtie p}[\psi]$  è soddisfatto dallo stato  $s$  se e solo se la probabilità che tutti i percorsi  $\omega$  che si originano da  $s$  soddisfino  $\psi$  è in relazione  $\bowtie$  con  $p$  (ad esempio  $\leq 0.5$ ). La regola 2.13 afferma, invece, che  $\mathbb{S}_{\bowtie p}[\phi]$  è soddisfatto dallo stato  $s$  se e solo se la somma delle probabilità a regime delle transizioni da  $s$  a qualsiasi stato  $s'$  che soddisfi  $\phi$  è  $\bowtie p$ .

La regola 2.14 asserisce che  $\mathcal{X}\phi$  è soddisfatto dal percorso  $\omega$  se e solo se esiste uno stato successivo ( $\omega(1)$ ) a quello corrente e tale stato soddisfa  $\phi$ . Le regole 2.15 e 2.16 descrivono la semantica dell'operatore  $\mathcal{U}$  nei casi illimitato e limitato. Nel primo,  $\phi_1 \mathcal{U} \phi_2$  è soddisfatto da  $\omega$  se e solo se esiste uno stato futuro  $\omega(k)$  che soddisfa  $\phi_2$  e tutti gli stati precedenti a  $\omega(k)$  soddisfano  $\phi_1$ . Nel secondo caso,  $\phi_1 \mathcal{U}^{\leq t} \phi_2$  è soddisfatto da  $\omega$  se e solo se esiste un tempo  $x$  minore o uguale a  $t$  in cui lo stato occupato al tempo  $x$  soddisfa  $\phi_2$  e qualsiasi stato occupato in un tempo minore di  $x$  soddisfa  $\phi_1$ .

Il tempo richiesto per la terminazione degli algoritmi di model checking è lineare rispetto alla lunghezza della formula e polinomiale rispetto al numero di stati del modello. Più in dettaglio, la verifica della formula  $\mathbb{P}_{\bowtie p}[\phi_1 \mathcal{U} \phi_2]$  ha complessità asintotica  $O(|\mathbb{S}|^3 + |\phi|)$ , mentre la verifica di  $\mathbb{P}_{\bowtie p}[\phi_1 \mathcal{U}^{\leq t} \phi_2]$  ha complessità  $O(|\mathbb{S}|^2 + q \cdot t + |\phi|)$  (con  $q$  fattore di uniformizzazione utilizzato per il metodo di Jensen) [60].

In PRISM, la verifica degli operatori  $\mathbb{P}$ ,  $\mathbb{R}[\mathcal{I} = t]$  e  $\mathbb{R}[\mathcal{C} \leq t]$  avviene grazie al *metodo di Jensen* (o algoritmo di uniformizzazione), mentre per gli

operatori  $\mathbb{S}$ ,  $\mathbb{R}[\mathcal{F}\phi]$  e  $\mathbb{R}[\mathcal{S}]$  si utilizzano metodi basati sulla risoluzione di sistemi di equazioni lineari.

L'algoritmo di uniformizzazione fissa un fattore  $q$  pari al rate della transizione più veloce della CTMC e, scalando tutte le transizioni del fattore  $q$ , ottiene una DTMC che ne approssima il comportamento e su cui è possibile calcolare il vettore delle probabilità transitorie dello stato  $s$  dopo un tempo  $t$  tramite:

$$\pi_{s,t} = \pi_{s,0} \cdot \sum_{i=0}^{\infty} \mathcal{P}_{q \cdot t}(i) \cdot \mathbf{P}^i,$$

dove  $\pi_{s,0}$  è il vettore delle probabilità transitorie iniziali,  $\mathcal{P}_{\lambda}(n) = \frac{\lambda^n e^{-\lambda}}{n!}$  è la distribuzione di probabilità di Poisson e  $\mathbf{P}$  è la matrice delle probabilità di transizione ( $\mathbf{R}$  scalato del fattore  $q$ ). Metodi numerici, come quello di *Fox-Glynn*, approssimano il calcolo fissando un errore  $\varepsilon$  e troncando la sommatoria quando l'errore sulla probabilità stimata è inferiore a  $\varepsilon$ .

A partire dalla DTMC uniformizzata, sono utilizzati i tre algoritmi di model checking:

- $\mathbb{P}[\mathcal{X}\phi]$  si verifica banalmente calcolando la probabilità che  $\phi$  sia vero nei successori dello stato iniziale;
- $\mathbb{P}[\phi_1 \mathcal{U} \phi_2]$  eliminati gli stati che soddisfano la proprietà CTL  $\forall[\phi_1 \mathcal{U} \phi_2]$  e quelli che non soddisfano  $\exists[\phi_1 \mathcal{U} \phi_2]$ , si risolve il *problema del percorso stocastico più breve* sugli stati rimanenti [67];
- $\mathbb{P}[\phi_1 \mathcal{U}^{\leq t} \phi_2]$  eliminando dalla DTMC uniformizzata tutte le transizioni che escono dagli stati che soddisfano  $\phi_2$  e da tutti gli stati che non soddisfano la formula CTL  $\exists[\phi_1 \mathcal{U} \phi_2]$ , si calcola la probabilità di trovarsi in uno stato in cui valga  $\phi_2$  al tempo  $t$  [58].

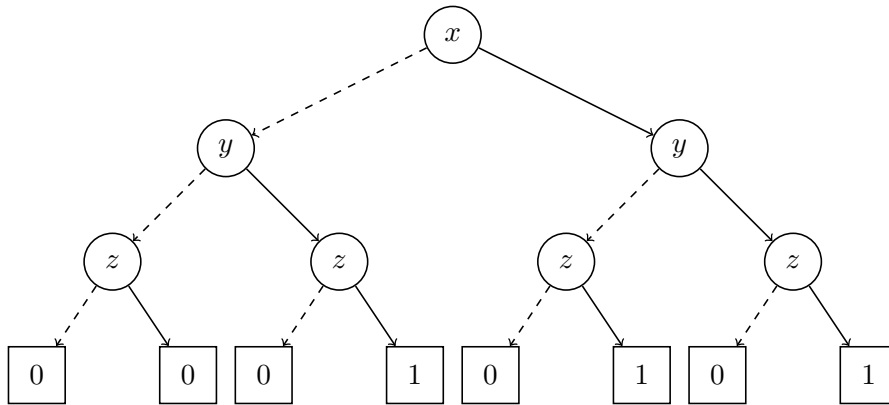
### 2.7.1 Model checking simbolico

Poiché la classe di complessità degli algoritmi di model checking è *polinomiale* rispetto allo spazio degli stati (ma questi crescono esponenzialmente al crescere del numero di componenti del sistema), è spesso opportuno effettuare il model checking utilizzando rappresentazioni simboliche della CTMC note come *multi terminal binary decision diagram* (MTBDD).

Una funzione logica è definita come  $f(x_1, x_2, \dots, x_n) : \mathbb{B}^n \mapsto \mathbb{B}$  e ad essa può essere associato un *albero di decisione* che la descrive graficamente. Ogni nodo dell'albero rappresenta una variabile e possiede due figli: uno raggiunto se la variabile è vera, l'altro se la variabile è falsa; le foglie costituiscono il valore della funzione.

In tabella 2.2 è esemplificata la tabella di verità della funzione logica  $(x \vee y) \wedge z$  e in figura 2.3 il corrispondente albero di decisione (con il ramo vero rappresentato da una linea continua e quello falso da una linea tratteggiata).

$x$	$y$	$z$	$(x \vee y) \wedge z$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabella 2.2: Tabella di verità della funzione logica  $(x \vee y) \wedge z$ Figura 2.3: Albero di decisione relativo alla funzione logica  $(x \vee y) \wedge z$ 

Come si può notare, per una funzione di  $n$  variabili, l'albero di decisione ha  $2^{n+1} - 1$  nodi.

Un *binary decision diagram* (BDD) [14] è un grafo diretto aciclico che collassa insieme più nodi dell'albero di decisione, ottenendo una rappresentazione compatta e canonica (due BDD della stessa funzione booleana sono sempre isomorfi). In figura 2.4 è rappresentato il BDD relativo all'albero di decisione di figura 2.3.

La costruzione di un BDD a partire dall'albero di decisione avviene attraverso le seguenti operazioni:

1. le foglie con lo stesso valore sono collassate;
2. i nodi isomorfi sono collassati;
3. i nodi ridondanti (dove sia il figlio vero che il figlio falso coincidono) sono eliminati.

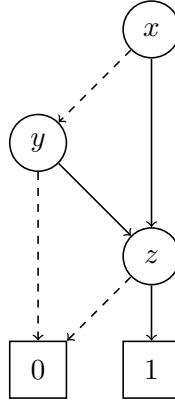


Figura 2.4: Binary decision diagram relativo alla funzione logica  $(x \vee y) \wedge z$

Nei BDD vale la *legge di Shannon*, che permette di espandere ricorsivamente la funzione logica in due sotto-BDD:

$$f_A = (\neg x \wedge f_{A|x=0}) \vee (x \wedge f_{A|x=1}),$$

dove  $f_A$  è la funzione logica di partenza,  $A|x=1$  è il sotto-BDD di  $A$  costruito a partire dal figlio vero della variabile  $x$  e, analogamente,  $A|x=0$  rappresenta il sotto-BDD costruito a partire dal figlio falso. Tale espansione permette di combinare più BDD tramite *operatori logici*, ottenendo una funzione logica (e quindi un BDD) pari alla combinazione delle funzioni di partenza. Per la logica dei predicati valgono le seguenti espansioni:

$$\begin{aligned} \neg f_A &= \neg x \wedge f_{A|x=1} \vee x \wedge f_{A|x=0} \\ f_A \odot f_B &= \neg x \wedge (f_{A|x=0} \odot f_{B|x=0}) \vee x \wedge (f_{A|x=1} \odot f_{B|x=1}) \\ \exists x f_A &= f_{A|x=0} \vee f_{A|x=1}. \end{aligned}$$

È possibile espandere un BDD per renderlo in grado di rappresentare funzioni con codominio reale ( $f(x_1, x_2, \dots, x_n) : \mathbb{B}^n \mapsto \mathbb{R}$ ), semplicemente inserendo una foglia per ogni valore mappato del codominio. La struttura così estesa prende il nome di *multi terminal binary decision diagram* (MTBDD). Le operazioni applicabili sugli MTBDD sono estese a tutti gli operatori *algebrici*.

Associando un identificatore intero ad ogni stato di una CTMC e utilizzando una funzione che ne rappresenta una codifica binaria di qualche tipo (ad esempio naturale,  $101 \mapsto 5$ , di Gray,  $111 \mapsto 5$ , o di altro tipo), è possibile rappresentare lo spazio degli stati con  $n$  variabili booleane. Utilizzando tale codifica, la matrice delle transizioni  $\mathbf{R} : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$  diventa una funzione booleana  $\mathbf{R} : \mathbb{B}^n \times \mathbb{B}^n \mapsto \mathbb{R} = \mathbb{B}^{2n} \mapsto \mathbb{R}$ , codificabile da un MTBDD [59].

I motori simbolici di PRISM, tuttavia, risparmiano memoria generando

l'MTBDD direttamente a partire dal modello, senza passare dalla CTMC intermedia. L'MTBDD relativo all'intero modello è costruito come somma degli MTBDD dei singoli moduli e ognuno di essi codifica una funzione logica con dominio costituito da variabili di riga (stato di partenza), variabili di colonna (stato di destinazione) e variabili logiche che codificano le variabili PRISM locali al modulo.

Dopo la costruzione dell'MTBDD, vengono eliminati tutti gli stati non raggiungibili a partire dallo stato iniziale. Tale operazione riduce la regolarità dell'MTBDD, aumentandone purtroppo il numero di nodi, tuttavia ciò si riflette, generalmente, in una maggiore efficienza del model checking (che fa uso degli stessi algoritmi descritti nel caso tradizionale, con il vantaggio di un numero di stati inferiore) [58].



## Capitolo 3

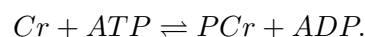
# Caso di studio: sviluppo di eritrociti ingegnerizzati per il trattamento della deficienza GAMT

In questo capitolo viene presentata come caso di studio una terapia in fase di sviluppo per la deficienza della guanidinoacetato metiltransferasi (sezione 3.1), basata su eritrociti ingegnerizzati per funzionare come bioreattore circolante nel flusso ematico. Dopo aver introdotto nella sezione 3.1 le caratteristiche del metabolismo della creatina e gli effetti della patologia su quest'ultimo e sull'organismo, viene descritto nella sezione 3.2 l'apparato in grado di incapsulare S-adenosilmetionina sintasi e guanidinoacetato metiltransferasi all'interno degli eritrociti. Infine vengono descritti gli strumenti di analisi utilizzati (sezioni 3.3 e 3.4) e lo stato attuale della sperimentazione *in vitro* del trattamento della deficienza GAMT (sezione 3.5).

### 3.1 Deficienza GAMT

#### 3.1.1 Metabolismo della creatina

La creatina (Cr) è un composto azotato presente nei tessuti ad elevato dispendio energetico (muscoli e neuroni) con funzioni di tampone e intermedio energetico, in grado di accumulare e rilasciare energia tramite la fosforilazione in fosfocreatina (PCr), con contemporanea defosforilazione di adenosintrifosfato (ATP) in adenosindifosfato (ADP):

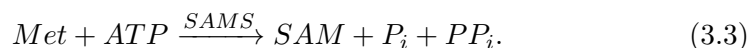


In condizioni fisiologiche, il pool di creatina dell'organismo è mantenuto a 6-50  $\mu\text{mol/l}$  nel siero e 11-244  $\text{mmol/l}$  nelle urine [56] da sintesi endogena, assunzione con la dieta e degradazione in creatinina (successivamente espulsa con le urine, insieme a una parte di creatina).

La via biosintetica della creatina avviene in due fasi. La prima, nei reni, produce *guanidinoacetato* e ornitina a partire da arginina e glicina tramite arginina-glicina amidinotransferasi. La seconda, nel fegato, trasferisce un gruppo metilico (usando *S-adenosil metionina* come donatore) sul guanidinoacetato per produrre creatina tramite *guanidinoacetato metiltransferasi*:

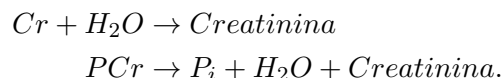


La S-adenosil metionina è prodotta dalle cellule a partire da metionina e ATP tramite *SAM sintasi*:



Altre reazioni possono interferire con la sintesi della creatina solo tramite sequestro o produzione dei precursori aminoacidici (arginina, glicina od ornitina) o di S-adenosil metionina, in quanto non esiste, nei mammiferi, nessun'altra reazione che utilizzi guanidinoacetato.

La produzione di creatinina avviene tramite idratazione della creatina o disidratazione della fosfocreatina:



I metaboliti della via della creatina sono trasportati a livello sistemico dal sangue e dal liquido cefalorachidiano. Gli scambi tra i due fluidi sono mediati dalle barriere *emato-encefalica* ed *emato-liquorale*.

A livello cellulare, gli scambi sono mediati dai seguenti trasportatori [80]:

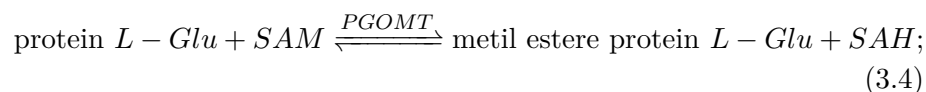
Famiglia SLC6 Solute Carrier 6, trasportatori attivi di guanidinocomposti (tra cui guanidinoacetato e creatina)  $\text{Na}^+$  e  $\text{Cl}^-$  dipendenti:

- Creatine Transporter 1 (CRT o SLC6A8) media l'ingresso di creatina e guanidinoacetato, sebbene il trasporto di quest'ultimo sia inibito dalla creatina (e quindi apprezzabile solo in pazienti GAMT deficienti),
- Taurine Transporter (TauT o SLC6A6) media il trasporto di taurina e l'efflusso di guanidinoacetato (con affinità un ordine di grandezza più basso rispetto alla taurina e inibizione ad opera di quest'ultima);

OCT3                      Organic Cationic Transporter 3, media l'efflusso di creatinina.

Gli eritrociti (che non sintetizzano nativamente creatina, ma che possono essere infusi con GAMT e SAMS, come descritto in sezione 3.5) costituiscono un ambiente con un numero molto limitato di interazioni, in quanto:

- non andando incontro a traduzione, non alterano il proprio pool aminoacidico in maniera significativa;
- pur possedendo SAMS endogena, non influiscono sensibilmente sull'aumento di SAM (l'attività dell'enzima risulta cinque ordini di grandezza più bassa rispetto alla variante espressa nel fegato);
- possiedono un solo enzima (protein glutammato O-metil transferasi) che consuma S-adenosil metionina, secondo la reazione:



- il substrato della protein glutammato O-metil transferasi è costituito dall'L-glutammato di proteine di membrana, modificato solo in maturazione, quindi l'enzima non opera negli eritrociti maturi.

Il trasporto di creatina mostra una cinetica bifasica che rallenta con l'età dei globuli rossi e risulta inibita da acido guanidinopropionico, indicando la presenza di SLC6A8 e di una diffusione passiva a bassa affinità, mentre quello di creatinina presenta una cinetica tradizionale [43]. Anche il trasporto di S-adenosil metionina è caratterizzato da una cinetica bifasica (la cinetica lenta sembra associata al metabolismo interno, più che a un trasporto vero e proprio, quella veloce suggerisce la presenza di due siti ad affinità diversa) [63, 79].

### 3.1.2 Deficienza dell'enzima guanidinoacetato metiltransferasi

Un malfunzionamento della guanidinoacetato metiltransferasi può causare rallentamento o blocco completo della reazione 3.2: tale malfunzionamento causa livelli bassi di creatina e accumulo di guanidinoacetato, non essendo utilizzato da nessun'altra reazione nell'intero organismo.

I bassi livelli di creatina causano deficit di tipo cognitivo e di sviluppo, mentre il guanidinoacetato, ad alte dosi, causa formazione di ossido nitrico e alterazione della membrana sinaptica (interferendo con le attività di ATPasi e ioni  $\text{Na}^+$  e  $\text{K}^+$ ) per antagonismo con il recettore  $\text{GABA}_A$ , risultando dunque neurotossico [33].

A causa della funzione di tampone energetico della creatina, sono presenti adattamenti metabolici (ma non istologici) secondari, atti a compensarne (tramite sovrapproduzione di ATP) la mancanza in pazienti GAMT deficienti. In

particolare, nei tessuti ad alto dispendio energetico, si osserva una sovraespressione degli enzimi coinvolti nella catena respiratoria (le quantità dei Complessi I, III e V, espressi nei mitocondri di fibroblasti incubati in assenza di creatina, si sono rivelate doppie rispetto alle colture di controllo), con conseguente aumento dell'attività mitocondriale [25].

Un elettroencefalogramma rileva convulsioni e iperattività di globo pallido, campo H di Forel, sostanza nera e corteccia, dopo iniezione intracisternale di guanidinocomposti in animali da laboratorio; ad eccezione del guanidinoacetato, gli altri guanidinocomposti causano inoltre formazione di radicali [54].

È stata dimostrata la natura genetica (autosomica recessiva) della deficienza GAMT, transfettando fibroblasti GAMT deficienti con vettori codificanti GAMT nativa sovraespressa e osservando un ripristino del normale metabolismo [4]. Topi knockout per il gene *gamt* presentano maggiore mortalità neonatale, ipotonia muscolare, fertilità maschile ridotta, perdita di peso non mediata da leptina e adattamenti metabolici analoghi a quelli riscontrati nei pazienti umani, rendendoli un modello biologico adeguato per lo studio della malattia [73].

### 3.1.3 Sintomi

La deficienza GAMT è caratterizzata da ritardo comportamentale e disabilità intellettuale, principalmente relativa ai domini del linguaggio e del comportamento, da attribuirsi alla carenza di creatina, e da epilessia e disturbi extrapiramidali (alterazione dei movimenti “espressivi” e di quelli involontari, come tremori, rallentamenti o spasmi), causati dall'accumulo di guanidinoacetato nel tessuto nervoso.

Nonostante il supporto scolastico, pazienti gravemente deficienti presentano funzionalità intellettuali estremamente limitate e mancano dell'indipendenza in età adulta; pazienti con sintomatologia lieve presentano invece funzionalità intellettuali sufficienti a raggiungere un certo livello d'indipendenza da adulti. La severità degli episodi epilettici è correlata alla gravità della deficienza e i disturbi di movimento non sono osservati in pazienti che non presentano anche fenomeni epilettici [78].

In [6, 31, 53, 56, 87] sono presentati alcuni casi clinici.

### 3.1.4 Diagnosi

Poiché i sintomi risultano aspecifici e gli effetti secondari sono analoghi all'encefalopatia mitocondriale [33], risulta necessario procedere ad esami diagnostici specifici (e ancora poco diffusi):

- sequenziamento del gene *gamt*;

- tomografia a risonanza magnetica, che rileva un aumento del segnale relativo al globo pallido;
- spettroscopia di risonanza magnetica  $^1H$  su urine e liquor, che rileva basse concentrazioni di creatina (assenza del picco singoletto a 3.05 ppm) e creatinina (assenza del picco singoletto a 3.13 ppm), e alte concentrazioni di guanidinoacetato (picco doppietto a 3.98 ppm nelle urine, non distinguibile nel liquor per la presenza di molecole simili) [29];
- spettroscopia di risonanza magnetica  $^{31}P$ , che rileva basse concentrazioni di fosfocreatina in cervello e muscoli, e alte concentrazioni di fosfoguanidinoacetato [68];
- reazione di Sakaguchi su campioni di urina: dopo una cromatografia a strato sottile, le piastre cromatografiche vengono prima immerse in una soluzione allo 0.1 % di idrossichinolina in acetone e poi spruzzate con bromo allo 0.3 % in soluzione 0.5 M di  $NaOH$ , il test è positivo (ma la diagnosi va confermata con altri metodi) se si formano macchie rosso-arancio, a causa della reazione tra bromo e guanidine monosostituite: tra queste l'unica che può essere presente nelle urine è il guanidinoacetato [75].

### 3.1.5 Terapie

Essendo la deficienza GAMT un difetto genetico, il trattamento può avvenire soltanto per integrazione di creatina ed eliminazione di guanidinoacetato.

Poiché alte dosi di creatina (1 g/kg/die nei ratti) riducono l'espressione di SLC6A8 [81], non è possibile eccedere con dosaggio o durata del trattamento. L'integrazione di creatina monoidrato segue i dosaggi previsti per i difetti del ciclo dell'urea (300-800 mg/kg/die) e può essere associata o meno a una dieta che riduca la formazione di guanidinoacetato (riducendo la disponibilità di arginina, per inibire la reazione 3.1) secondo i seguenti regimi [78]:

- dieta ipoproteica;
- dieta ipoproteica (0.6-1.8 mg/kg/die) e formula di aminoacidi non contenente arginina;
- dieta fortemente ipoproteica (0.2-0.5 mg/kg/die) con restrizione di arginina a 250 mg/kg/die e formula di aminoacidi non contenente arginina.

Ognuno dei regimi può essere ulteriormente integrato dall'aggiunta di ornitina (100 mg/kg/die), atta a spostare l'equilibrio della reazione 3.1 verso i reagenti. La restrizione dell'arginina non interferisce con le vie di detossificazione dei composti azotati, ma causa una riduzione del guanidinoacetato, associata a una scomparsa quasi completa di convulsioni e attività epileptogeniche [74]. Effettuando una risonanza magnetica  $^1H$  durante la terapia,

si osserva un recupero delle concentrazioni di creatina più rapido nel tessuto muscolare rispetto a quello nervoso [30].

Oltre a una dieta carente di arginina, è possibile inibire la reazione 3.1 somministrando benzoato di sodio (100 mg/kg/die), che sequestra la glicina formando acido ippurico (escreto rapidamente dai reni).

Un trattamento precoce (prenatale o nei primi mesi di vita) è in grado di prevenire o ridurre notevolmente i difetti di sviluppo, pertanto è importante uno screening tempestivo, prima dell'insorgenza dei sintomi [78].

## 3.2 Red Cell Loader

Red Cell Loader (RCL) [51] è un apparato, sviluppato dall'Università di Urbino, in grado di incapsulare sostanze non diffusibili all'interno dei globuli rossi tramite processi osmotici reversibili. Il sangue così arricchito costituisce una nuova via di somministrazione a rilascio prolungato di farmaci o mezzi di contrasto, e la possibilità di creare un bioreattore in grado di circolare *in vivo*, sequestrando dal circolo ematico substrati (ad esempio sostanze tossiche o farmaci inattivi) e rilasciando prodotti (ad esempio farmaci attivati) [10, 49, 71].

RCL presenta vantaggi rispetto a tecniche di incapsulamento preesistenti basate su dialisi ad alto ematocrito [35], quali apirogenicità, sterilità ed emocompatibilità (che lo rendono adatto per utilizzi *in vivo*), la ridotta quantità di sangue necessario (50 ml contro un'unità di sangue, 450 ml), la velocità della procedura (circa due ore) e l'alta resa (sopravvivenza del 35-50 % dei globuli rossi e incapsulamento del 30 % di sostanza). Lo stesso apparato permette inoltre di ingegnerizzare l'"invecchiamento" degli eritrociti, modulando quindi il rate di fagocitosi (e conseguentemente il rate di rilascio della sostanza) ad opera dei macrofagi.

Gli eritrociti caricati dal RCL sono già stati sperimentati nei seguenti ambiti clinici:

- rilascio prolungato di azidotimidina ed etambutolo per il trattamento di infezioni disseminate di *Mycobacterium avium* in pazienti con AIDS allo stadio avanzato [48, 69];
- rilascio di competitori di nucleosidi fosforilati all'interno di monociti e macrofagi infetti da HIV [50];
- rilascio prolungato e localizzato di dexametasone in pazienti con broncopneumopatia cronica ostruttiva [70];
- incapsulamento di mezzi di contrasto per risonanza magnetica per aumentarne l'emivita (in forma libera sono eliminati dai reni in poche ore) [5].

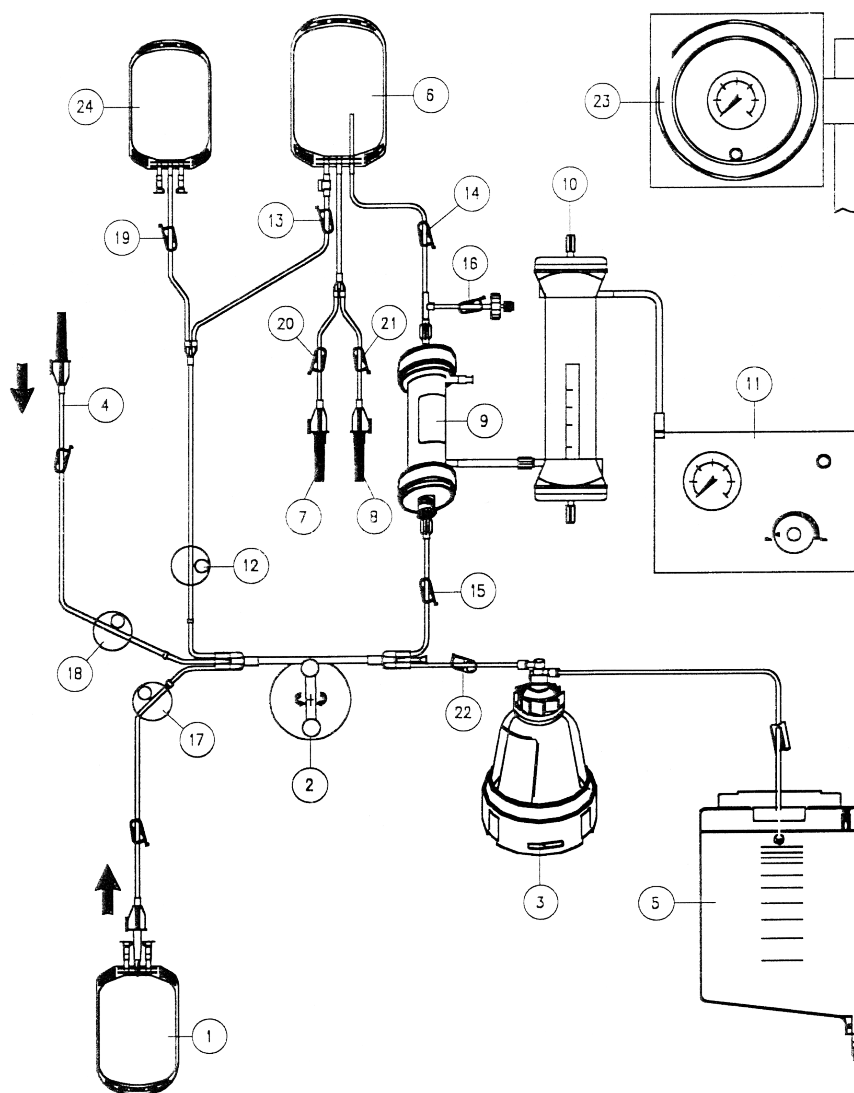


Figura 3.1: Red Cell Loader

### 3.3 Cromatografia liquida ad alte prestazioni

La cromatografia liquida ad alte prestazioni (HPLC) è una tecnica di separazione e analisi delle componenti di un liquido. Come per la cromatografia tradizionale, la separazione avviene tramite il passaggio della *fase mobile* (da analizzare) attraverso una colonna analitica al cui interno è contenuta la *fase stazionaria*, costituita da particelle sferiche in grado di interagire con la fase mobile trattenendola (ad esempio perché porose o cariche elettrostaticamente). Componenti diverse della fase mobile avranno differente affinità per la fase stazionaria e conseguentemente la attraverseranno a velocità diverse. Ponendo uno spettrometro all'uscita della colonna, è possibile analizzare le singole componenti man mano che ne escono.

A differenza della tecnica tradizionale, l'HPLC utilizza alte pressioni (anziché la semplice forza di gravità) per spingere la fase mobile attraverso la colonna analitica, diminuendo drasticamente il tempo di analisi e aumentando la precisione. L'utilizzo di una pompa garantisce un flusso costante e le alte pressioni permettono l'utilizzo di colonne più sottili che impediscono alla fase mobile di diffondersi trasversalmente, alterando il tempo di uscita dalla colonna.

### 3.4 Spettrometria di massa a ionizzazione per elettrospray

L'elettrospray è una tecnica di ionizzazione di macromolecole che permette la desolvatazione (la separazione del soluto dal solvente) senza il rischio di frammentazione delle molecole. L'applicazione di alta tensione causa il passaggio dalla fase liquida alla dispersione in aerosol per repulsione elettrostatica. Il ridotto volume delle microgocce causa un brusco aumento della densità di carica, che si riflette su un ulteriore aumento delle forze repulsive, le quali a loro volta innescano una separazione in microgocce più piccole. Il fenomeno continua finché le singole molecole ionizzate non sono espulse dalla goccia (venendo effettivamente separate dal solvente) e disperse in fase gassosa in una camera a vuoto.

La procedura specifica per la misura di guanidinoacetato e creatina da gocce di sangue essiccate su cartone parte dall'estrazione dal cartone con una soluzione di acqua, metanolo e N-metil-D<sub>3</sub>-creatina (creatina marcata con deuterio) e viene effettuata risospesando in acqua, metanolo e acido acetico, dopo l'evaporazione del primo solvente. L'analisi impiega circa un minuto e presenta range di misura lineari sia per guanidinoacetato che per creatina, con una precisione tale da rendere l'elettrospray una tecnica diagnostica promettente per le deficienze GAMT e AGAT [16].



### 3.5 Stato dell'arte del trattamento della deficienza GAMT

È in fase di sperimentazione l'utilizzo del red cell loader per incapsulare negli eritrociti GAMT nativa dei pazienti, che sopperisca alla mancanza della GAMT difettosa. Gli esperimenti programmati appartengono ai seguenti gruppi:

1. misura dell'uptake di guanidinoacetato ad opera delle membrane degli eritrociti;
2. incapsulamento di GAMT negli eritrociti e misura della sintesi di creatina;
3. incapsulamento di GAMT e SAMS negli eritrociti e misura della sintesi di creatina.

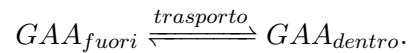
Allo stato attuale, la sperimentazione *in vitro* sta procedendo con la seconda fase. Le simulazioni *in silico* (capitolo 5) seguono lo stesso ordine e sono state invece portate a termine.

Il workflow generale per ogni esperimento procede nel seguente modo:

incapsulamento	gli eritrociti sono separati dal sangue di un donatore e incapsulati tramite RCL con enzimi ed eventualmente substrati;
analisi HPLC	un'aliquota di eritrociti ingegnerizzati viene lisata e analizzata tramite HPLC per verificare le quantità effettivamente incapsulate;
incubazione	vengono aggiunti i metaboliti esterni (tra cui guanidinoacetato marcato con $^{13}C$ , HEPES come tampone a pH fisiologico e PIGPA per il ringiovanimento degli eritrociti [84]) e la soluzione viene incubata per un tempo prestabilito;
analisi elettrospray	ogni reazione è fermata per essiccazione su cartone e le gocce di sangue vengono analizzate per elettrospray, quantificando il guanidinoacetato e la creatina marcati con $^{13}C$ .

#### 3.5.1 Esperimento sull'uptake del guanidinoacetato

Poiché gli eritrociti ingegnerizzati tramite RCL sostituiscono il fegato relativamente alla reazione 3.2, è necessario caratterizzare la cinetica di trasporto del guanidinoacetato tra plasma e interno degli eritrociti:



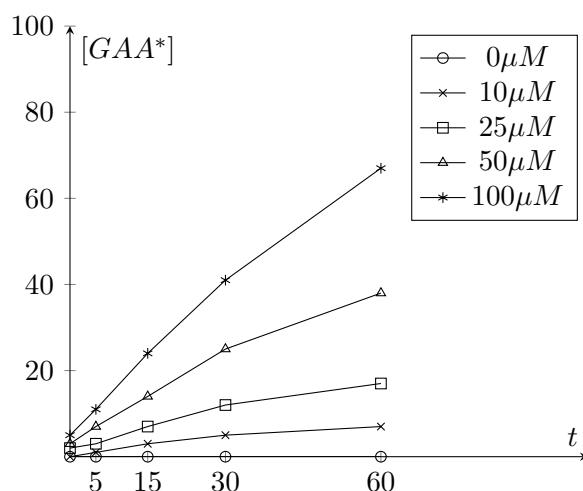


Figura 3.2: Rate di ingresso del guanidinoacetato negli eritrociti a diverse concentrazioni

Per verificare che la membrana degli eritrociti sia effettivamente permeabile al guanidinoacetato e per osservarne la cinetica, sono state incubate a  $37^\circ C$  quantità costanti di eritrociti (al 15% di ematocrito) con quantità variabili di guanidinoacetato  $^{13}C$ .

A intervalli di tempo prefissati (0, 5, 15, 30 e 60 minuti) sono state prelevate aliquote da cui sono stati separati gli eritrociti per centrifugazione a 9600 g per 3 minuti in presenza di 1-bromo-dodecano. Al termine della centrifugazione sono stati ottenuti pellet al 50% circa di ematocrito, che sono stati essiccati su cartone e analizzati tramite elettrospray.

La figura 3.2 riassume i dati ottenuti in un grafico della concentrazione del guanidinoacetato intracellulare nel tempo, partendo da diverse concentrazioni di guanidinoacetato extracellulare. L'andamento dei grafici è coerente con una diffusione passiva del guanidinoacetato attraverso la membrana.

Per meglio caratterizzare la cinetica, è stato realizzato il diagramma dei doppi reciproci in figura 3.3, ponendo in ascissa l'inverso delle concentrazioni e in ordinata l'inverso delle velocità, calcolate come pendenza della retta di regressione lineare ottenuta sui punti di ognuna delle curve, ad eccezione della curva corrispondente a  $0\mu M$  di guanidinoacetato esterno. È stata effettuata un'ulteriore regressione lineare sui punti per ottenere la retta di regressione:

$$y = 48.9333x + 0.0853333, \quad (R^2 = 0.998832),$$

da cui è stata calcolata una costante  $v\_uptake = 0.02\mu M/min$ .

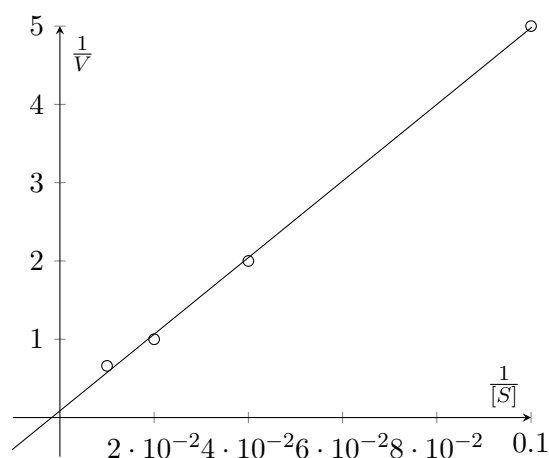


Figura 3.3: Diagramma dei doppi reciproci dell'ingresso di guanidinoacetato negli eritrociti

### 3.5.2 Esperimenti sul dosaggio della creatina prodotta

In un paziente GAMT deficiente, l'elevata disponibilità di guanidinoacetato nel plasma si riflette in una cinetica potenzialmente elevata della GAMT incapsulata negli eritrociti. Oltre all'uptake di guanidinoacetato, l'altro elemento limitante può essere costituito dall'S-adenosil metionina, che rischia di non essere prodotta a velocità sufficientemente elevata dalla SAMS eritrocitaria.

Per verificare se sia necessario incapsulare, oltre alla GAMT, SAMS clonata da *E. coli*, si sono effettuati esperimenti di dosaggio della creatina prodotta, a partire da quantità preformate di S-adenosil metionina.

Il sangue è stato centrifugato fino a ottenere tre aliquote di  $780 \mu\text{l}$  al 90 % di ematocrito. Una delle aliquote è stata addizionata a  $220 \mu\text{l}$  di HEPES e funge da controllo, le altre due sono state addizionate ciascuna a  $220 \mu\text{l}$  di soluzione contenente  $660 \mu\text{g}$  di GAMT. Da ognuna sono stati prelevati  $50 \mu\text{l}$  e alle aliquote contenenti GAMT sono state aggiunte rispettivamente  $2.5 \mu\text{l}$  e  $10 \mu\text{l}$  di soluzione  $5 \text{ nM}$  di S-adenosil metionina. Le due aliquote sono state processate tramite RCL per incapsulare GAMT e S-adenosil metionina, e, insieme all'aliquota controllo, sono state diluite fino a ottenere un ematocrito al 50 %.

Dalle tre soluzioni sono stati prelevati  $250 \mu\text{l}$  ed è stata verificata tramite HPLC la quantità di S-adenosil metionina effettivamente incapsulata, ottenendo una concentrazione di circa  $25 \mu\text{M}$  per l'aliquota trattata con  $2.5 \mu\text{l}$  e  $125 \mu\text{M}$  per l'aliquota trattata con  $10 \mu\text{l}$ .

Sono stati aggiunti  $7 \mu\text{l}$  di soluzione contenente guanidinoacetato  $^{13}\text{C}$  a concentrazione  $5 \text{ nM}$  e si sono incubate le soluzioni a  $37^\circ\text{C}$ . Ai tempi 0, 1, 2, 3 e 20 ore sono stati prelevati  $50 \mu\text{l}$  di soluzione ed essiccati su cartoncino per bloccare le reazioni.

L'analisi elettrospray ha prodotto le seguenti tabelle:

Tempo (ore)	$[GAA]$ ( $\mu M$ )	$[Cr]$ ( $\mu M$ )
0	0.13	59.80
1	0.15	72.20
2	0.16	75.25
3	0.20	75.86
20	0.27	76.31

Tabella 3.1: Concentrazioni di guanidinoacetato e creatina in eritrociti non ingegnerizzati.

Tempo (ore)	$[GAA]$ ( $\mu M$ )	$[Cr]$ ( $\mu M$ )
0	2.61	66.01
1	3.10	61.90
2	4.62	63.37
3	5.11	67.50
20	5.26	59.31

Tabella 3.2: Concentrazioni di guanidinoacetato e creatina in eritrociti ingegnerizzati con GAMT e 25  $\mu M$  di SAM.

Tempo (ore)	$[GAA]$ ( $\mu M$ )	$[Cr]$ ( $\mu M$ )
0	7.32	66.32
1	10.52	64.55
2	13.87	65.87
3	12.18	61.58
20	15.13	59.88

Tabella 3.3: Concentrazioni di guanidinoacetato e creatina in eritrociti ingegnerizzati con GAMT e 125  $\mu M$  di SAM.

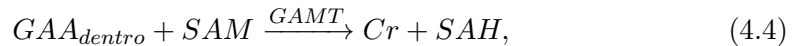
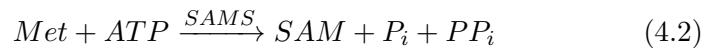
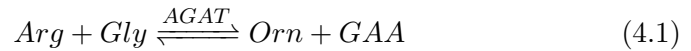
## Capitolo 4

# Costruzione del modello per il trattamento della deficienza GAMT

In questo capitolo viene presentata la fase di descrizione formale del caso di studio, riassumendo brevemente le caratteristiche specifiche del metabolismo (sezione 4.1) e mostrando il modello Bio-PEPA (sezione 4.2) che lo descrive e i modelli PRISM generati a partire da quest'ultimo (sezione 4.3).

### 4.1 Via metabolica da modellare

Riassumendo i dati raccolti nel capitolo 3, si parte dalla seguente via:



Assumendo che la produzione di guanidinoacetato avvenga molto più lentamente della sua degradazione, si può astrarre dalla reazione 4.1, considerando una quantità iniziale di guanidinoacetato libera nel plasma che non viene rifornita. Vista l'elevata efficacia dei meccanismi di regolazione, si può assumere che i metabolismi energetici e di biosintesi riescano a compensare ampiamente l'aumentato fabbisogno di metionina e ATP degli eritrociti, di conseguenza si possono assumere livelli costanti di questi ultimi nella reazione 4.2. L'equilibrio della reazione 4.3 risulta essere spostato verso i prodotti, a causa del sequestro immediato del guanidinoacetato intracellulare ad opera della reazione 4.4,

di conseguenza è possibile semplificare la reazione rendendola unidirezionale (uptake).

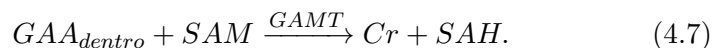
Le cinetiche coinvolte nelle tre reazioni rimaste sono:

SAMS catalisi con formazione di complessi ternari ad ordine di associazione obbligata, inibita da S-adenosil metionina;

GAMT catalisi con formazione di complessi ternari ad ordine di associazione obbligata, inibita da S-adenosil omocisteina;

trasporto diffusione passiva attraverso la membrana, dal plasma, verso l'interno degli eritrociti (equazione 2.3).

Per limitare l'esplosione dello spazio degli stati del modello, è necessario eliminare tutto ciò che non è rilevante ai fini delle cinetiche (ad eccezione della creatina, di cui è richiesta la quantificazione), di conseguenza le reazioni sono semplificate come segue:

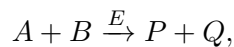


#### 4.1.1 Cinetiche di reazione

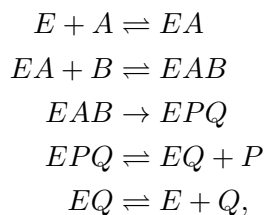
Per modellare fedelmente il caso di studio, è necessario ricavare le velocità a cui dovranno avvenire le azioni che modellano le reazioni precedentemente descritte. Di seguito vengono ricavate le cinetiche utili alla modellazione del caso di studio.

#### Catalisi enzimatica con formazione di complessi ternari

Gli enzimi che catalizzano le reazioni 4.5 e 4.7 sono nella forma:



che costituisce in realtà una rappresentazione astratta (che considera solo gli istanti iniziale e finale, ignorando i passaggi intermedi) della sequenza di reazioni più complessa:



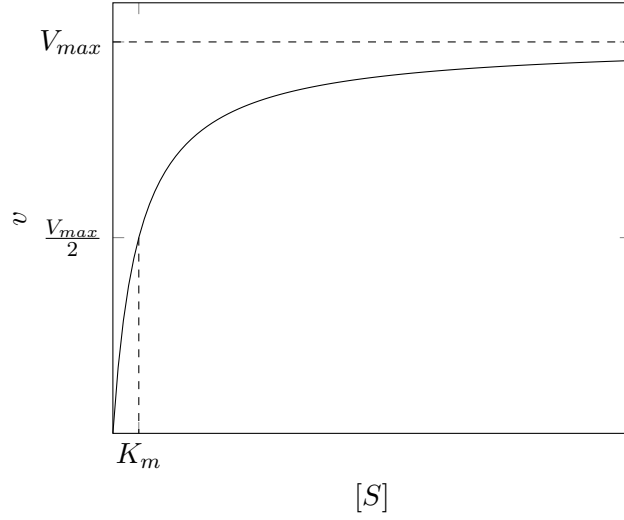


Figura 4.1: Andamento della velocità di reazione in funzione della concentrazione di substrato in un enzima di Michaelis-Menten

dove i due substrati  $A$  e  $B$  sono convertiti nei due prodotti  $P$  e  $Q$  ed è possibile un solo ordine di associazione (prima si forma il complesso binario  $EA$  e poi quello ternario  $EAB$ , analogamente, prima si dissocia il prodotto  $P$  e poi quello  $Q$ ).

Ai fini del calcolo della cinetica, risulta essere più conveniente ragionare sui singoli passi della reazione, anziché sulla rappresentazione astratta. Una volta ricavata l'equazione della cinetica, si può applicare quest'ultima direttamente alla reazione astratta, per semplificare il modello.

Utilizzando metodi come quello di King-Altman [42], che semplificano il calcolo della cinetica nonostante l'elevato numero di reazioni coinvolte, si ottiene l'equazione:

$$v = \frac{\frac{V_{max}}{km_B + [B]} \cdot [A]}{\frac{ki \cdot km_B + km_A \cdot [B]}{km_B + [B]} + [A]}. \quad (4.8)$$

Risulta evidente che:

$$V_{max}^{app} = \frac{V_{max}}{km_B + [B]}$$

$$K_M^{app} = \frac{ki \cdot km_B + km_A \cdot [B]}{km_B + [B]},$$

dove  $k_i$  indica un'eventuale costante di inibizione,  $km_A$  e  $km_B$  indicano le affinità dell'enzima rispetto ad  $A$  e  $B$ , rispettivamente, e  $V_{max}$  indica la velocità massima dell'enzima.

Come precedentemente affermato, la funzione 4.8 definisce l'andamento della velocità delle reazioni catalizzate da SAMS e GAMT.

## 4.2 Modello Bio-PEPA

Il modello Bio-PEPA per il trattamento della deficienza GAMT a base di eritrociti ingegnerizzati, prodotto a partire dalle reazioni 4.5, 4.6 e 4.7, è il seguente:

$$\begin{aligned}
 sams &= \left[ 10 \cdot \frac{\frac{v_{sams}}{km_{sams\_atp+atp}} \cdot met}{\frac{ki_{sams\_sam} \cdot km_{sams\_atp+atp} + km_{sams\_met\_atp}}{km_{sams\_atp+atp}} + met} \right]; \\
 gamt &= \left[ 10 \cdot \frac{\frac{v_{gamt}}{km_{gamt\_sam} + \frac{SAM}{10}} \cdot \frac{GAA\_INT}{10}}{\frac{ki_{gamt\_sah} \cdot km_{gamt\_sam} + km_{gamt\_gaa\_int} \cdot \frac{SAM}{10}}{km_{gamt\_sam} + \frac{SAM}{10}} + \frac{GAA\_INT}{10}} \right]; \\
 uptake &= \left[ v_{uptake} \cdot \frac{GAA\_EXT - GAA\_INT}{10} \right]; \\
 UM\_SAM &= \left[ \frac{SAM}{10} \right]; \\
 UM\_SAH &= \left[ \frac{SAH}{10} \right]; \\
 UM\_GAA\_INT &= \left[ \frac{GAA\_INT}{10} \right]; \\
 UM\_GAA\_EXT &= \left[ \frac{GAA\_EXT}{10} \right]; \\
 UM\_CR &= \left[ \frac{CR}{10} \right];
 \end{aligned} \tag{4.9}$$

$$\begin{aligned}
 SAM &= sams \uparrow SAM + gamt \downarrow SAM; \\
 SAH &= gamt \uparrow SAH; \\
 GAA\_INT &= uptake \uparrow GAA\_INT + gamt \downarrow GAA\_INT; \\
 GAA\_EXT &= uptake \downarrow GAA\_EXT; \\
 CR &= gamt \uparrow CR;
 \end{aligned} \tag{4.10}$$

$$(SAM \underset{gamt}{\bowtie} SAH \underset{gamt}{\bowtie} GAA\_INT \underset{gamt}{\bowtie} CR) \underset{uptake}{\bowtie} GAA\_EXT. \tag{4.11}$$

La porzione 4.9 descrive rate e contatori definiti sul modello. I rate sono associati alle tre azioni definite, che coincidono con le reazioni da modellare, ognuno è definito in base alla cinetica coinvolta ed è espresso in  $10^{-7} M \cdot min^{-1}$  (mentre le costanti sono definite in  $\mu M$  e  $\mu M \cdot min^{-1}$ , motivo per cui è necessario includere fattori di conversione). I contatori sono utilizzati per convertire le concentrazioni delle specie chimiche del modello in  $\mu M$ .



Le costanti di processo sono definite nella porzione 4.10 e ognuna rappresenta il comportamento di ogni specie chimica, osservata in aliquote da  $10^{-7}$  *M*. Il comportamento dei processi è definito in termini di aumenti e diminuzioni a seguito di ogni reazione. Il processo *SAM*, ad esempio, può aumentare a seguito della partecipazione alla reazione 4.5 (catalizzata da *SAMS*), oppure diminuire partecipando alla reazione 4.7 (catalizzata da *GAMT*).

Il modello vero e proprio è costituito dalla porzione 4.11, che sincronizza tutti i processi. La semantica dell'operatore di composizione parallela modella fedelmente il comportamento reale, poiché ogni reazione è costituita dalla trasformazione simultanea di tutte le specie coinvolte, ma, allo stesso tempo, due specie non coinvolte nella stessa reazione possono trasformarsi indipendentemente l'una dall'altra. Essendo il risultato della composizione di due processi un nuovo processo, è possibile costruire il modello incrementalmente, sincronizzando man mano le singole specie, si può dunque immaginare la porzione 4.11 costruita come:

1. *SAM* e *SAH* sincronizzati in un unico processo *reagenti\_gamt*;
2. *GAA\_INT* e *CR* sincronizzati nel processo *prodotti\_gamt*;
3. *reagenti\_gamt* e *prodotti\_gamt* sincronizzati nel processo *reazione\_gamt* (dove l'azione *gamt* non richiede ulteriori sincronizzazioni, dato che tutti i processi che la utilizzano sono stati accorpati);
4. *reazione\_gamt* e *GAA\_EXT* sincronizzati nel modello completo (dato che l'azione *uptake* va sincronizzata solo tra *GAA\_INT*, che è incluso in *reazioni\_gamt* e *GAA\_EXT* e dato che l'azione *sams* non ha bisogno di sincronizzazioni, essendo eseguita dal solo processo *SAM*).

Dallo stesso modello è possibile analizzare diversi aspetti della terapia semplicemente modificando il valore delle quantità iniziali e delle costanti. Le tabelle seguenti indicano i parametri utilizzati per ogni analisi.

### 4.2.1 Ingresso di guanidinoacetato negli eritrociti

In questa serie di istanze del modello si replica l'esperimento descritto nella sezione 3.5.1, per validare la fedeltà del modello rispetto ai dati sperimentali. Di queste istanze verrà fatto un utilizzo esclusivamente simulativo.

Parametro	Valore	Unità	Note
<i>SAM</i>	0	$10^{-7}$ M	
<i>SAH</i>	0	$10^{-7}$ M	
<i>GAA_INT</i>	5	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	100	$10^{-7}$ M	valore sperimentale
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	0	$\mu$ M	
<i>atp</i>	0	$\mu$ M	
<i>v_sams</i>	0	$\mu$ M/min	
<i>v_gamt</i>	0	$\mu$ M/min	
<i>km_sams_atp</i>	0	$\mu$ M	
<i>km_sams_met</i>	0	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	0	$\mu$ M	
<i>km_gamt_gaa_int</i>	0	$\mu$ M	
<i>ki_gamt_sah</i>	0	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	valore sperimentale

Tabella 4.1: Esperimento di uptake con  $10\mu$ M di guanidinoacetato extracellulare

Parametro	Valore	Unità	Note
<i>SAM</i>	0	$10^{-7}$ M	
<i>SAH</i>	0	$10^{-7}$ M	
<i>GAA_INT</i>	12	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	250	$10^{-7}$ M	valore sperimentale
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	0	$\mu$ M	
<i>atp</i>	0	$\mu$ M	
<i>v_sams</i>	0	$\mu$ M/min	
<i>v_gamt</i>	0	$\mu$ M/min	
<i>km_sams_atp</i>	0	$\mu$ M	
<i>km_sams_met</i>	0	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	0	$\mu$ M	
<i>km_gamt_gaa_int</i>	0	$\mu$ M	
<i>ki_gamt_sah</i>	0	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.2: Esperimento di uptake con  $25\mu$ M di guanidinoacetato extracellulare

Parametro	Valore	Unità	Note
<i>SAM</i>	0	$10^{-7}$ M	
<i>SAH</i>	0	$10^{-7}$ M	
<i>GAA_INT</i>	25	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	500	$10^{-7}$ M	valore sperimentale
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	0	$\mu$ M	
<i>atp</i>	0	$\mu$ M	
<i>v_sams</i>	0	$\mu$ M/min	
<i>v_gamt</i>	0	$\mu$ M/min	
<i>km_sams_atp</i>	0	$\mu$ M	
<i>km_sams_met</i>	0	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	0	$\mu$ M	
<i>km_gamt_gaa_int</i>	0	$\mu$ M	
<i>ki_gamt_sah</i>	0	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.3: Esperimento di uptake con  $50\mu$ M di guanidinoacetato extracellulare

Parametro	Valore	Unità	Note
<i>SAM</i>	0	$10^{-7}$ M	
<i>SAH</i>	0	$10^{-7}$ M	
<i>GAA_INT</i>	52	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	1000	$10^{-7}$ M	valore sperimentale
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	0	$\mu$ M	
<i>atp</i>	0	$\mu$ M	
<i>v_sams</i>	0	$\mu$ M/min	
<i>v_gamt</i>	0	$\mu$ M/min	
<i>km_sams_atp</i>	0	$\mu$ M	
<i>km_sams_met</i>	0	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	0	$\mu$ M	
<i>km_gamt_gaa_int</i>	0	$\mu$ M	
<i>ki_gamt_sah</i>	0	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.4: Esperimento di uptake con  $100\mu$ M di guanidinoacetato extracellulare

### 4.2.2 Sintesi di creatina in eritrociti incapsulati con SAM esogena

In questa serie di istanze del modello si replica l'esperimento descritto nella sezione 3.5.2, per validare la fedeltà del modello rispetto ai dati sperimentali. Anche queste istanze verranno sottoposte esclusivamente ad analisi simulativa.

Parametro	Valore	Unità	Note
<i>SAM</i>	100	$10^{-7}$ M	valore sperimentale
<i>SAH</i>	13	$10^{-7}$ M	[55]
<i>GAA_INT</i>	660	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	1000	$10^{-7}$ M	valore sperimentale
<i>CR</i>	26	$10^{-7}$ M	valore sperimentale
<i>met</i>	12	$\mu$ M	[55]
<i>atp</i>	1100	$\mu$ M	[55]
<i>v_sams</i>	0.033	$\mu$ M/min	[41]
<i>v_gamt</i>	33	$\mu$ M/min	valore sperimentale
<i>km_sams_atp</i>	80	$\mu$ M	[55]
<i>km_sams_met</i>	2	$\mu$ M	[55]
<i>ki_sams_sam</i>	2	$\mu$ M	[55]
<i>km_gamt_sam</i>	14.8	$\mu$ M	[39]
<i>km_gamt_gaa_int</i>	78	$\mu$ M	[39]
<i>ki_gamt_sah</i>	0.4	$\mu$ M	[12]
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.5: Esperimento di sintesi della creatina con  $10\mu$ M di S-adenosil metionina incapsulata

Parametro	Valore	Unità	Note
<i>SAM</i>	500	$10^{-7}$ M	valore sperimentale
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	663	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	1000	$10^{-7}$ M	
<i>CR</i>	73	$10^{-7}$ M	valore sperimentale
<i>met</i>	12	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	0.033	$\mu$ M/min	
<i>v_gamt</i>	33	$\mu$ M/min	
<i>km_sams_atp</i>	80	$\mu$ M	
<i>km_sams_met</i>	2	$\mu$ M	
<i>ki_sams_sam</i>	2	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.6: Esperimento di sintesi della creatina con  $50 \mu$ M di S-adenosil metionina incapsulata

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	[55]
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	598	$10^{-7}$ M	valore sperimentale
<i>GAA_EXT</i>	1000	$10^{-7}$ M	
<i>CR</i>	1	$10^{-7}$ M	valore sperimentale
<i>met</i>	12	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	0.033	$\mu$ M/min	
<i>v_gamt</i>	0	$\mu$ M/min	non incapsulata
<i>km_sams_atp</i>	80	$\mu$ M	
<i>km_sams_met</i>	2	$\mu$ M	
<i>ki_sams_sam</i>	2	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.7: Esperimento di controllo della sintesi della creatina con S-adenosil metionina endogena

### 4.2.3 Predizione della sintesi di creatina in eritrociti incapsulati con SAM sintasi di *E. coli*

In questa serie di istanze del modello si prevede l'andamento di esperimenti non ancora effettuati, dove si valuta la quantità di SAM sintasi di *E. coli* (con affinità e costanti di equilibrio diverse rispetto a quella eritrocitaria, usata come controllo) da incapsulare per fare in modo che l'S-adenosil metionina non sia il reagente limitante del metabolismo. Queste istanze verranno utilizzate sia per l'applicazione delle tecniche simulative che di verifica.

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	livelli patologici [40]
<i>GAA_EXT</i>	500	$10^{-7}$ M	concentrazione massima ipotizzata
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	12	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	0.033	$\mu$ M/min	
<i>v_gamt</i>	15	$\mu$ M/min	assumendo un incapsulamento del 30% circa di 1 mg/ml
<i>km_sams_atp</i>	2	$\mu$ M	
<i>km_sams_met</i>	80	$\mu$ M	
<i>ki_sams_sam</i>	2	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.8: Esperimento di controllo della sintesi della creatina con SAM sintasi endogena

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	20	$\mu$ M/min	[13], 0.05 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	[13]
<i>km_sams_met</i>	75	$\mu$ M	[13]
<i>ki_sams_sam</i>	0	$\mu$ M	[13]
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.9: Esperimento di sintesi della creatina con 0.05 mg di SAM sintasi di *E. coli* incapsulati

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	40	$\mu$ M/min	0.1 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	
<i>km_sams_met</i>	75	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.10: Esperimento di sintesi della creatina con 0.1 mg di SAM sintasi di *E. coli* incapsulati

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	200	$\mu$ M/min	0.5 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	
<i>km_sams_met</i>	75	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.11: Esperimento di sintesi della creatina con 0.5 mg di SAM sintasi di *E. coli* incapsulati

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	400	$\mu$ M/min	1 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	
<i>km_sams_met</i>	75	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.12: Esperimento di sintesi della creatina con 1 mg di SAM sintasi di *E. coli* incapsulati



Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	1000	$\mu$ M/min	2.5 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	
<i>km_sams_met</i>	75	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.13: Esperimento di sintesi della creatina con 2.5 mg di SAM sintasi di *E. coli* incapsulati

Parametro	Valore	Unità	Note
<i>SAM</i>	35	$10^{-7}$ M	
<i>SAH</i>	13	$10^{-7}$ M	
<i>GAA_INT</i>	37	$10^{-7}$ M	
<i>GAA_EXT</i>	500	$10^{-7}$ M	
<i>CR</i>	0	$10^{-7}$ M	
<i>met</i>	20	$\mu$ M	
<i>atp</i>	1100	$\mu$ M	
<i>v_sams</i>	2000	$\mu$ M/min	5 mg incapsulati
<i>v_gamt</i>	15	$\mu$ M/min	
<i>km_sams_atp</i>	73	$\mu$ M	
<i>km_sams_met</i>	75	$\mu$ M	
<i>ki_sams_sam</i>	0	$\mu$ M	
<i>km_gamt_sam</i>	14.8	$\mu$ M	
<i>km_gamt_gaa_int</i>	78	$\mu$ M	
<i>ki_gamt_sah</i>	0.4	$\mu$ M	
<i>v_uptake</i>	0.02	$\mu$ M/min	

Tabella 4.14: Esperimento di sintesi della creatina con 5 mg di SAM sintasi di *E. coli* incapsulati

### 4.3 Modelli PRISM

I modelli PRISM compilati a partire dalle istanze del modello Bio-PEPA per effettuare l'analisi sono tutti della stessa forma, cambiando soltanto le costanti definite a inizio file e le quantità iniziali dei moduli.

Per semplicità viene mostrato un unico modello parametrizzato, indicando in corsivo gli identificatori Bio-PEPA con cui sostituire i parametri (ad esempio, nel modello 4.1 è necessario sostituire *met* con 0, nel modello 4.5 con 12 e così via). Per ragioni di spazio, le cinetiche sono indicate con *sams*, *gamt* e *uptake* (anziché con le formule già indicate nel modello Bio-PEPA), sebbene siano le stesse per ogni modello.

In questo modello parametrico vengono definiti un modulo per ogni specie (dove nel caso di un aumento viene controllato che non sia raggiunta la saturazione e nel caso di una diminuzione viene controllato che la specie non sia esaurita), e un modulo complessivo per tutti i rate. Vengono inoltre definite tutte le strutture di reward che verranno utilizzate per l'analisi e che corrispondono al numero di reazioni effettuate, alla quantità di ogni substrato, al quadrato della quantità e ai contatori Bio-PEPA definiti.

```

1  ctmc
2    const double _met = met;
3    const double _atp = atp;
4    const double _v_sams = v_sams;
5    const double _v_gamt = v_gamt;
6    const double _km_sams_atp = km_sams_atp;
7    const double _km_sams_met = km_sams_met;
8    const double _ki_sams_sam = ki_sams_sam;
9    const double _km_gamt_sam = km_gamt_sam;
10   const double _km_gamt_gaa_int = km_gaa_int;
11   const double _ki_gamt_sah = km_gamt_sah;
12   const double _v_uptake = v_uptake;
13
14   module Rates
15     [_sams] sams : true;
16     [_gamt] gamt : true;
17     [_uptake] uptake : true;
18   endmodule
19
20   const int MAX = SAM + SAH + GAA_INT + GAA_EXT +
      CR;
21
22   module _SAM
23     _SAM : [0..MAX] init SAM;
24     [_sams] (_SAM + 1 ≤ MAX) → 1 : (_SAM' = _SAM + 1);

```

```

25     [_gamt] ( $\_SAM \geq 1$ )  $\rightarrow 1 : (\_SAM' = \_SAM - 1)$ ;
26 endmodule
27
28 module  $\_SAH$ 
29      $\_SAH : [0..MAX]$  init  $SAH$ ;
30     [_gamt] ( $\_SAH + 1 \leq MAX$ )  $\rightarrow 1 : (\_SAH' = \_SAH + 1)$ ;
31 endmodule
32
33 module  $\_GAA\_INT$ 
34      $\_GAA\_INT : [0..MAX]$  init  $GAA\_INT$ ;
35     [_uptake] ( $\_GAA\_INT + 1 \leq MAX$ )  $\rightarrow 1 : (\_GAA\_INT' =$ 
36          $\_GAA\_INT + 1)$ ;
37     [_gamt] ( $\_GAA\_INT \geq 1$ )  $\rightarrow 1 : (\_GAA\_INT' = \_GAA\_INT$ 
38          $- 1)$ ;
39 endmodule
40
41 module  $\_GAA\_EXT$ 
42      $\_GAA\_EXT : [0..MAX]$  init  $GAA\_EXT$ ;
43     [_uptake] ( $\_GAA\_EXT \geq 1$ )  $\rightarrow 1 : (\_GAA\_EXT' =$ 
44          $\_GAA\_EXT - 1)$ ;
45 endmodule
46
47 module  $\_CR$ 
48      $\_CR : [0..MAX]$  init  $CR$ ;
49     [_gamt] ( $\_CR + 1 \leq MAX$ )  $\rightarrow 1 : (\_CR' = \_CR + 1)$ ;
50 endmodule
51
52 rewards " $\_sams$ "
53     [ $\_sams$ ] true : 1;
54 endrewards
55
56 rewards " $\_gamt$ "
57     [ $\_gamt$ ] true : 1;
58 endrewards
59
60 rewards " $\_uptake$ "
61     [ $\_uptake$ ] true : 1;
62 endrewards
63
64 rewards " $\_SAM$ "
65     true :  $\_SAM$ ;
66 endrewards

```

```
66   rewards "_SAM_squared"
67     true : _SAM * _SAM;
68   endrewards
69
70   rewards "_SAH"
71     true : _SAH;
72   endrewards
73
74   rewards "_SAH_squared"
75     true : _SAH * _SAH;
76   endrewards
77
78   rewards "_GAA_INT"
79     true : _GAA_INT;
80   endrewards
81
82   rewards "_GAA_INT_squared"
83     true : _GAA_INT * _GAA_INT;
84   endrewards
85
86   rewards "_GAA_EXT"
87     true : _GAA_EXT;
88   endrewards
89
90   rewards "_GAA_EXT_squared"
91     true : _GAA_EXT * _GAA_EXT;
92   endrewards
93
94   rewards "_CR"
95     true : _CR;
96   endrewards
97
98   rewards "_CR_squared"
99     true : _CR * _CR;
100  endrewards
101
102  rewards "_UM_SAM"
103    true : (_SAM / 10.0);
104  endrewards
105
106  rewards "_UM_SAH"
107    true : (_SAH / 10.0);
108  endrewards
109
```

```
110  rewards "_UM_GAA_INT"  
111    true : (_GAA_INT / 10.0);  
112  endrewards  
113  
114  rewards "_UM_GAA_EXT"  
115    true : (_GAA_EXT / 10.0);  
116  endrewards  
117  
118  rewards "_UM_CR"  
119    true : (_CR / 10.0);  
120  endrewards
```

## Capitolo 5

# Analisi simulativa del trattamento della deficienza GAMT

In questo capitolo viene presentato l'esito delle simulazioni effettuate sui modelli sotto forma di grafici relativi all'andamento delle varie specie in funzione del tempo. Più in dettaglio, sono stati ripetuti gli esperimenti già effettuati *in vitro*, relativamente all'uptake di guanidinoacetato (sez. 5.1) e al dosaggio della creatina prodotta (sez. 5.2), e gli esperimenti non ancora effettuati relativi all'incapsulamento di SAMS e GAMT (sez. 5.3).

### 5.1 Simulazione dell'esperimento sull'uptake del guanidinoacetato

I modelli 4.1-4.4 sono stati utilizzati per simulare l'esperimento di uptake di guanidinoacetato (sezione 3.5.1). Ogni modello è stato simulato per 60 minuti, con un campionamento degli eventi ogni 10 minuti. La figura 5.1 raggruppa le quattro simulazioni e le sovrappone ai dati sperimentali.

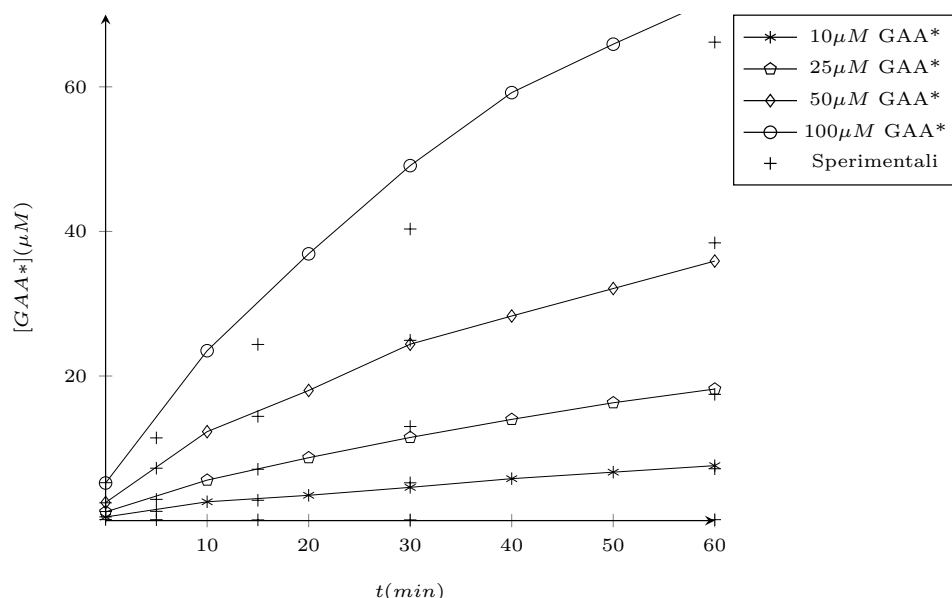


Figura 5.1: Validazione dell'esperimento di uptake di guanidinoacetato

Si può notare come la fedeltà delle simulazioni rispetto ai dati sperimentali sia molto alta per tutte le concentrazioni iniziali considerate, ad eccezione del caso dell'incubazione con  $100\mu M$  di guanidinoacetato. Tale curva presenta una leggera sovrastima rispetto ai dati sperimentali da attribuire, probabilmente, a una sottospecifica del modello. Il volume degli eritrociti è infatti finito ed è ragionevole pensare che si presentino fenomeni di saturazione non dipendenti dal meccanismo di trasporto (che essendo una diffusione passiva non può essere, per sua natura, saturabile), bensì dal poco spazio a disposizione all'interno degli eritrociti per contenere grandi quantità di guanidinoacetato.

Questa sovrastima è comunque accettabile, considerando il fatto che le aliquote di eritrociti ingegnerizzati utilizzate per questa serie di esperimenti risulta molto bassa rispetto a quelle utilizzate per gli esperimenti successivi.

## 5.2 Simulazione dell'esperimento sulla sintesi di creatina

I modelli 4.5-4.7 simulano l'esperimento di dosaggio della creatina prodotta da eritrociti ingegnerizzati con GAMT (sezione 3.5.2). Ogni modello è stato simulato per 10 ore, con un campionamento ogni minuto. Nei grafici 5.2 e 5.3 sono mostrati gli andamenti delle concentrazioni del guanidinoacetato intracellulare e della creatina nel tempo. Il numero di marker è stato ridotto per aumentare la leggibilità dei grafici.

Si può osservare dalle curve “Unloaded”, relative al modello 4.5, che gli eritrociti nativi non sono dotati di alcuna attività di sintesi della creatina, quindi si limitano ad accumulare guanidinoacetato al loro interno.

Le altre curve presentano invece due regioni separate da un brusco cambiamento di tendenza che coincide con l'esaurimento di S-adenosil metionina incapsulata. Nella prima si osservano una decrescita del guanidinoacetato e una crescita della creatina prodotta, indicando l'utilizzo dell'S-adenosil metionina incapsulata. La seconda mostra invece un andamento costante della creatina e un accumulo di guanidinoacetato, indicando che una volta esaurita l'S-adenosil metionina, la SAMS nativa non è in grado di produrne di nuova a una velocità sufficientemente elevata da far riprendere la reazione di sintesi della creatina.



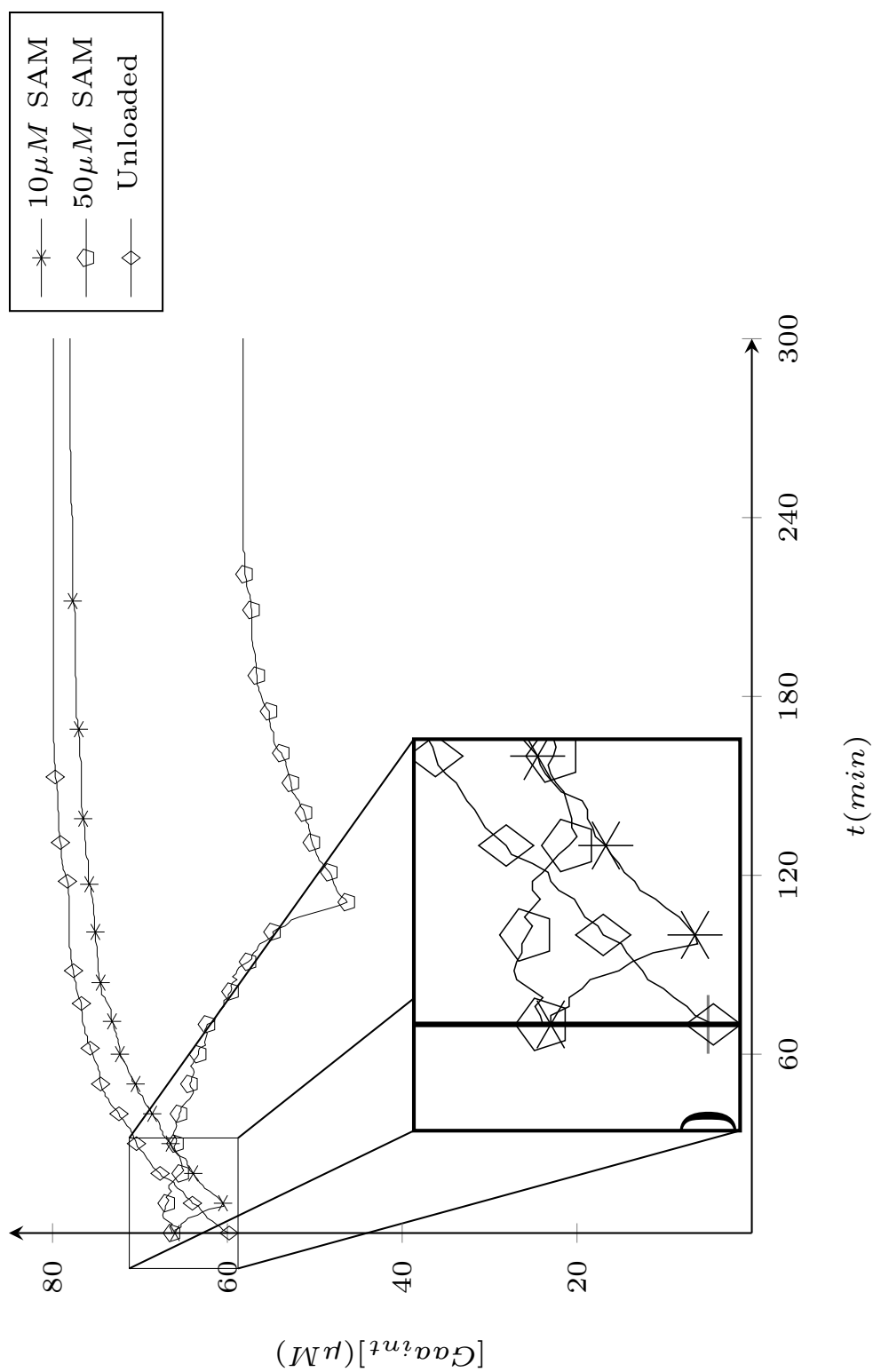


Figura 5.2: Validazione dell'esperimento di sintesi di creatina: andamento del guanidinoacetato intracellulare

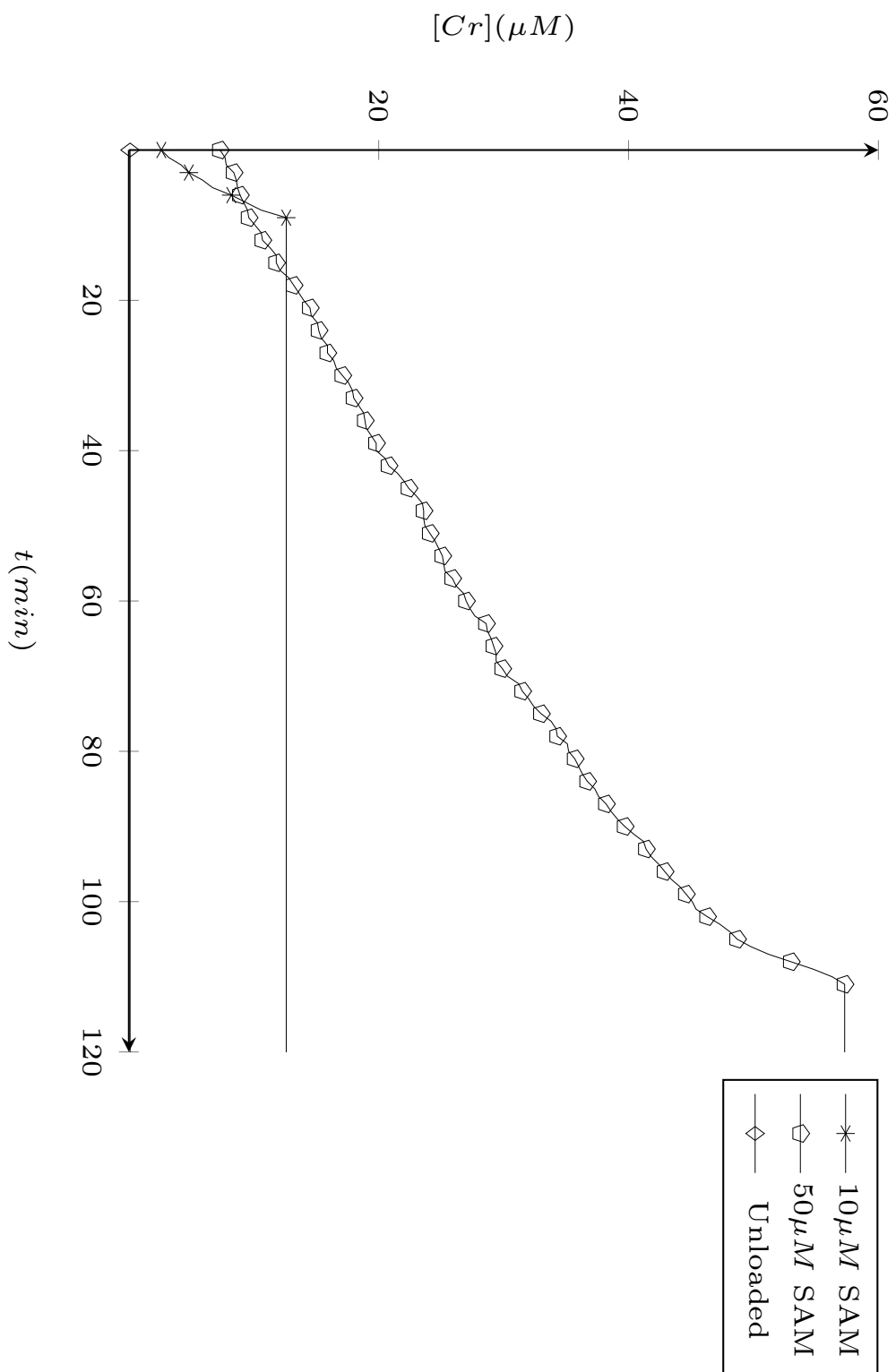


Figura 5.3: Validazione dell'esperimento di sintesi di creatina: andamento della creatina

### 5.3 Simulazione di eritrociti ingegnerizzati con GAMT e SAMS

I modelli 4.8-4.14 simulano esperimenti non ancora effettuati che prevedono l'incapsulamento di una quantità fissa di GAMT e di una quantità variabile di SAMS clonata da *E. coli*. Ogni modello è stato simulato per 10 ore, con un campionamento ogni 10 minuti. Le figure 5.4-5.10 mostrano l'andamento di tutte le specie per ogni simulazione, mentre le figure 5.11 e 5.12 confrontano l'andamento, rispettivamente, della sintesi di creatina e del consumo di guanidinoacetato tra tutte le simulazioni, ad eccezione del modello 4.14 che, andando in saturazione, presenta dati non realistici.

Si può osservare che il cambiamento di tendenza sulle curve del guanidinoacetato, appena visibile in figura 5.4 (e coincidente con l'esaurimento dell'S-adenosil metionina), viene mascherato nelle altre figure dall'attività della SAMS. L'S-adenosil metionina prodotta viene immediatamente sequestrata dalla GAMT, quindi la sintesi di creatina e il consumo di guanidinoacetato sono in grado di proseguire, nonostante i livelli di S-adenosil metionina siano mantenuti apparentemente a zero.

L'attività della GAMT risulta essere proporzionale all'attività della SAMS incapsulata in un intervallo molto ampio. Nelle figure 5.9 e 5.10 si osserva invece una cinetica della SAMS che supera quella della GAMT, causando un accumulo di S-adenosil metionina, fino a raggiungere la saturazione.

Le figure 5.11 e 5.12 evidenziano l'andamento proporzionale della cinetica della GAMT rispetto alla quantità di SAMS incapsulata (e quindi della sua cinetica).

### 66 5.3 Simulazione di eritrociti ingegnerizzati con GAMT e SAMS

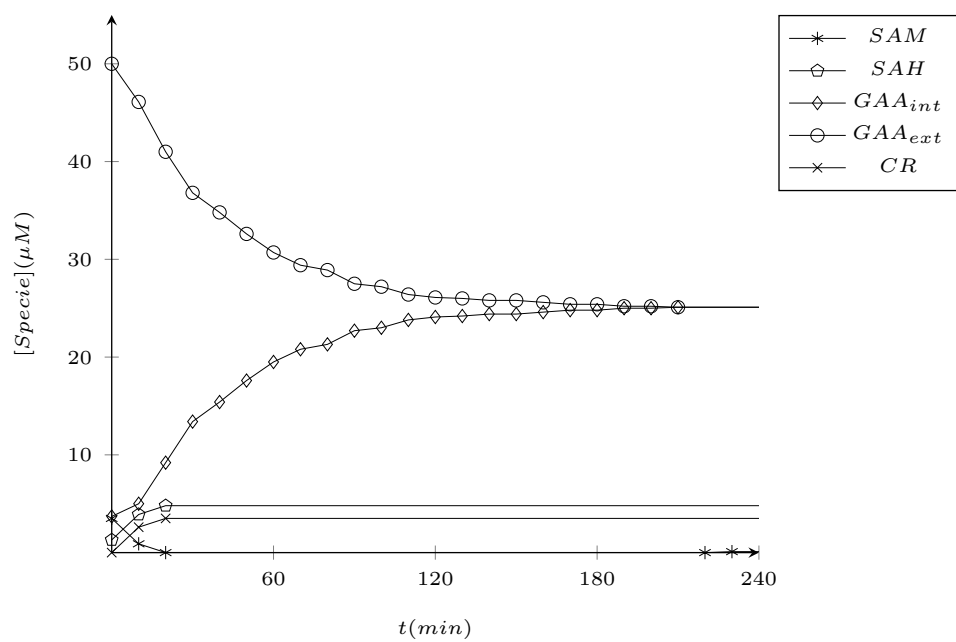


Figura 5.4: Simulazione di controllo con SAMS nativa (modello 4.8)

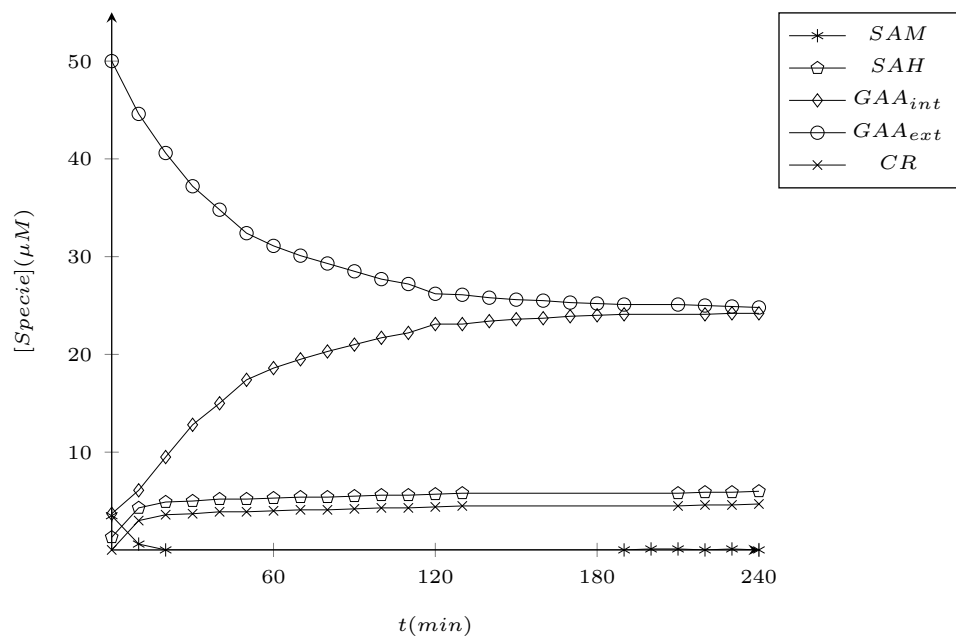


Figura 5.5: Simulazione dell'incapsulamento di 0.05 mg di SAMS (modello 4.9)

### 5.3 Simulazione di eritrociti ingegnerizzati con GAMT e SAMS 67

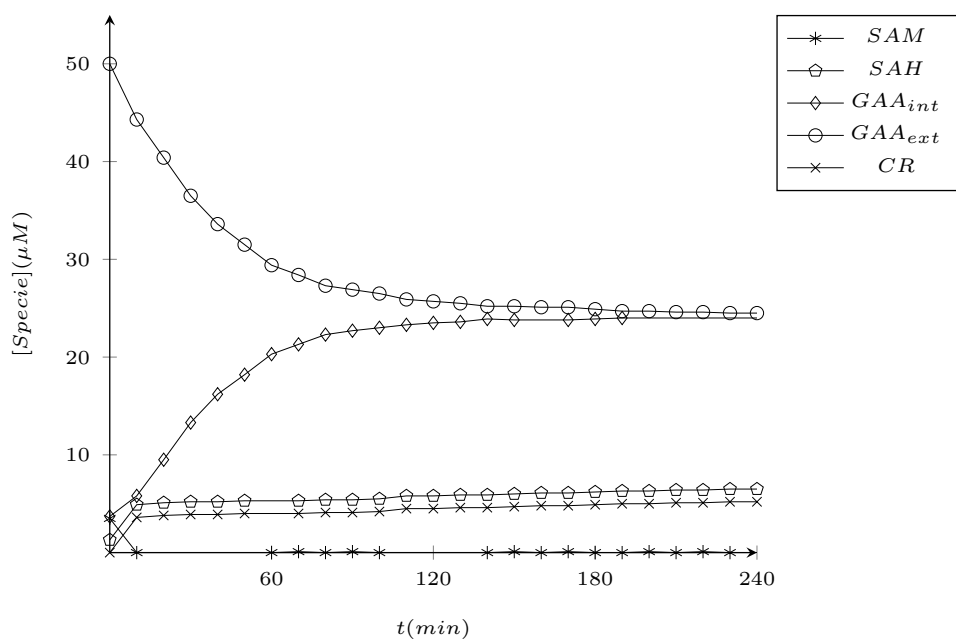


Figura 5.6: Simulazione dell'incapsulamento di 0.1 mg di SAMS (modello 4.10)

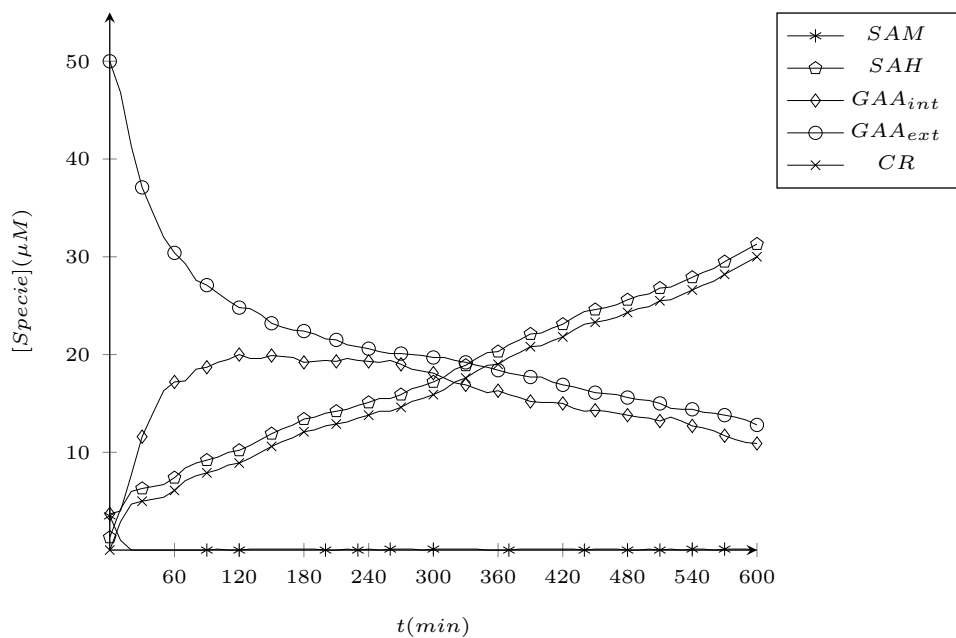


Figura 5.7: Simulazione dell'incapsulamento di 0.5 mg di SAMS (modello 4.11)

### 68 5.3 Simulazione di eritrociti ingegnerizzati con GAMT e SAMS

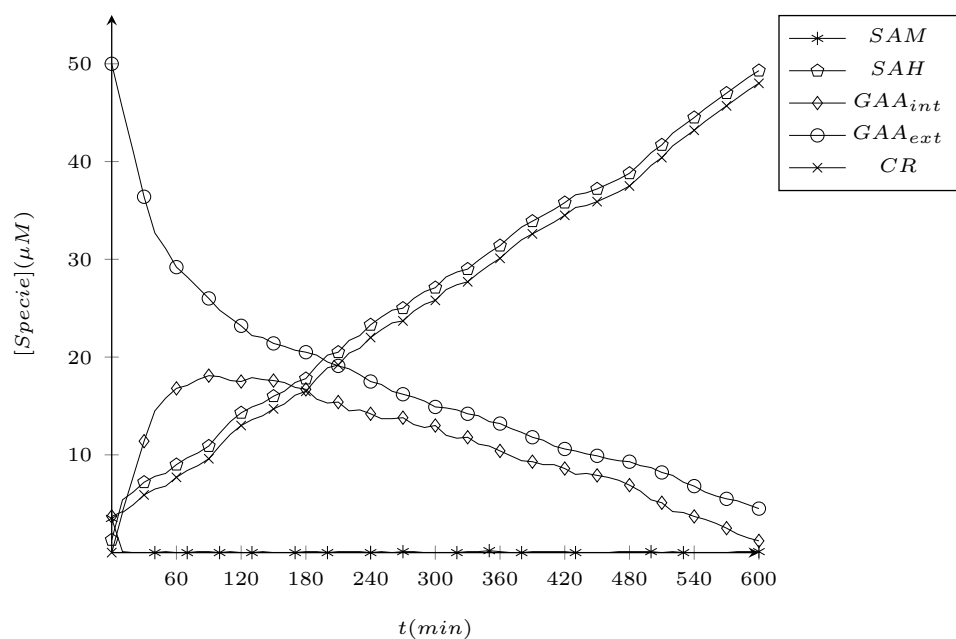


Figura 5.8: Simulazione dell'incapsulamento di 1 mg di SAMS (modello 4.12)

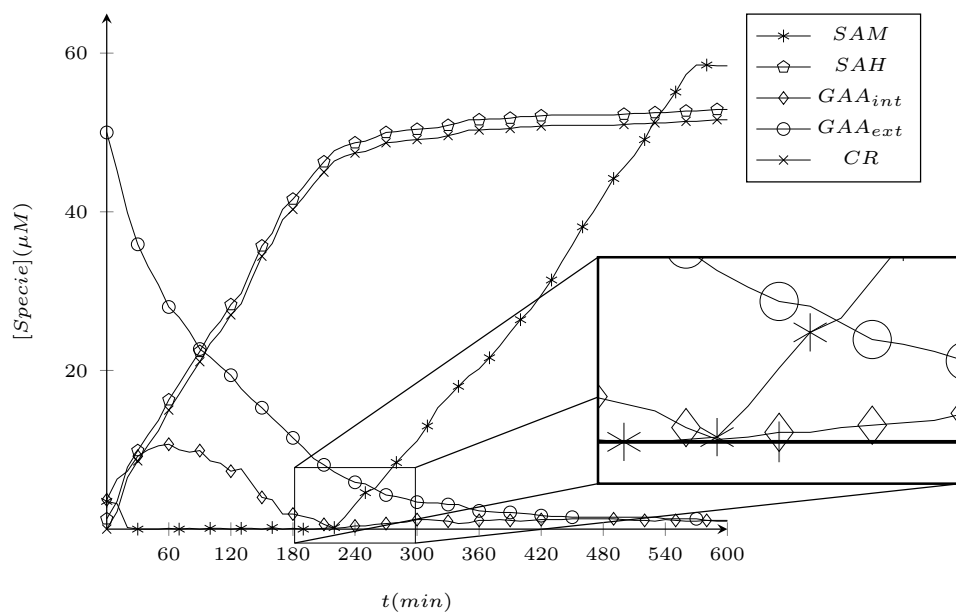


Figura 5.9: Simulazione dell'incapsulamento di 2.5 mg di SAMS (modello 4.13)

### 5.3 Simulazione di eritrociti ingegnerizzati con GAMT e SAMS 69

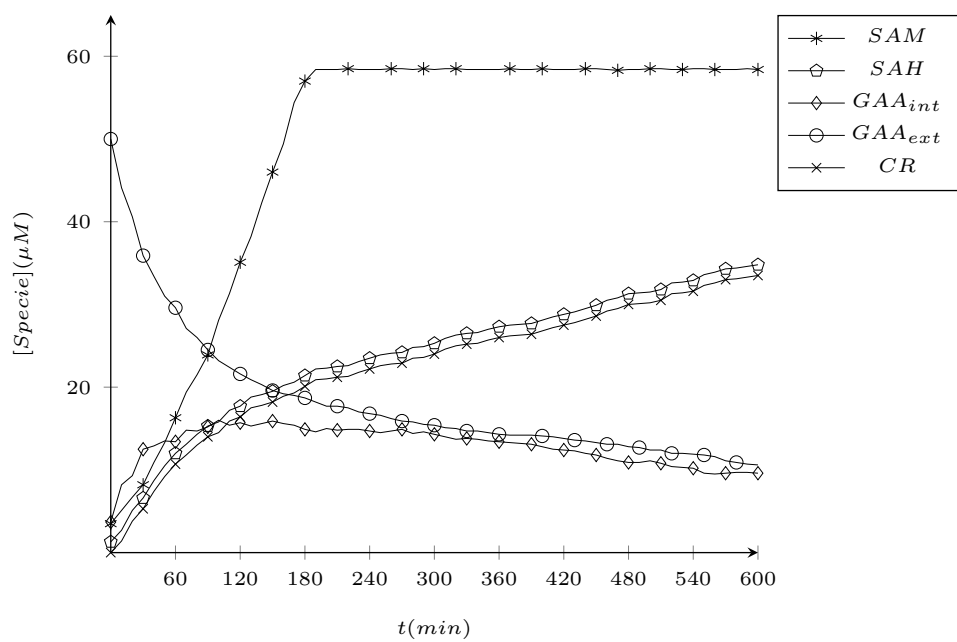


Figura 5.10: Simulazione dell'incapsulamento di 5 mg di SAMS (modello 4.14)

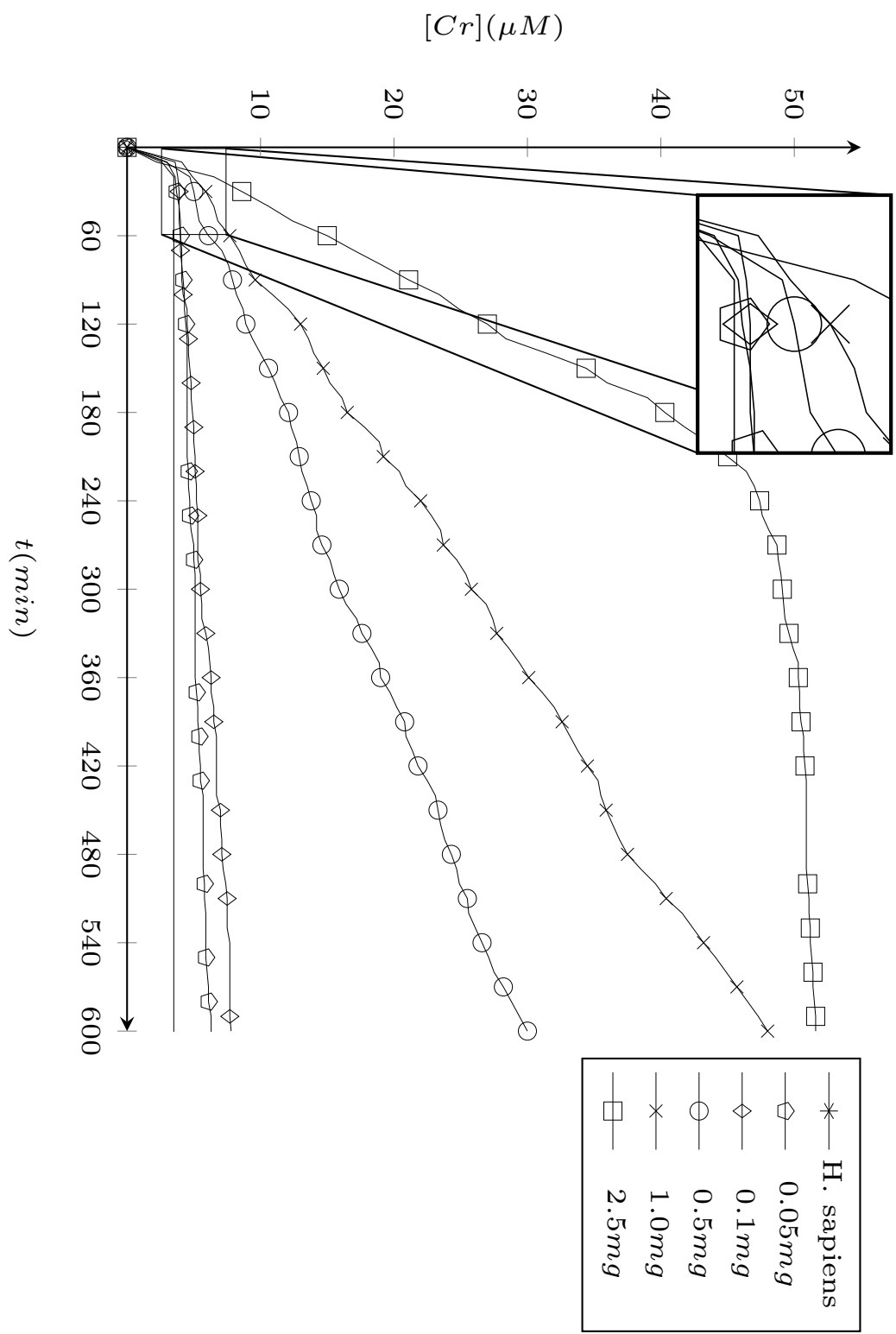


Figura 5.11: Sintesi di creatina negli eritrociti ingegnerizzati con quantità variabili di SAMS di *E. coli*



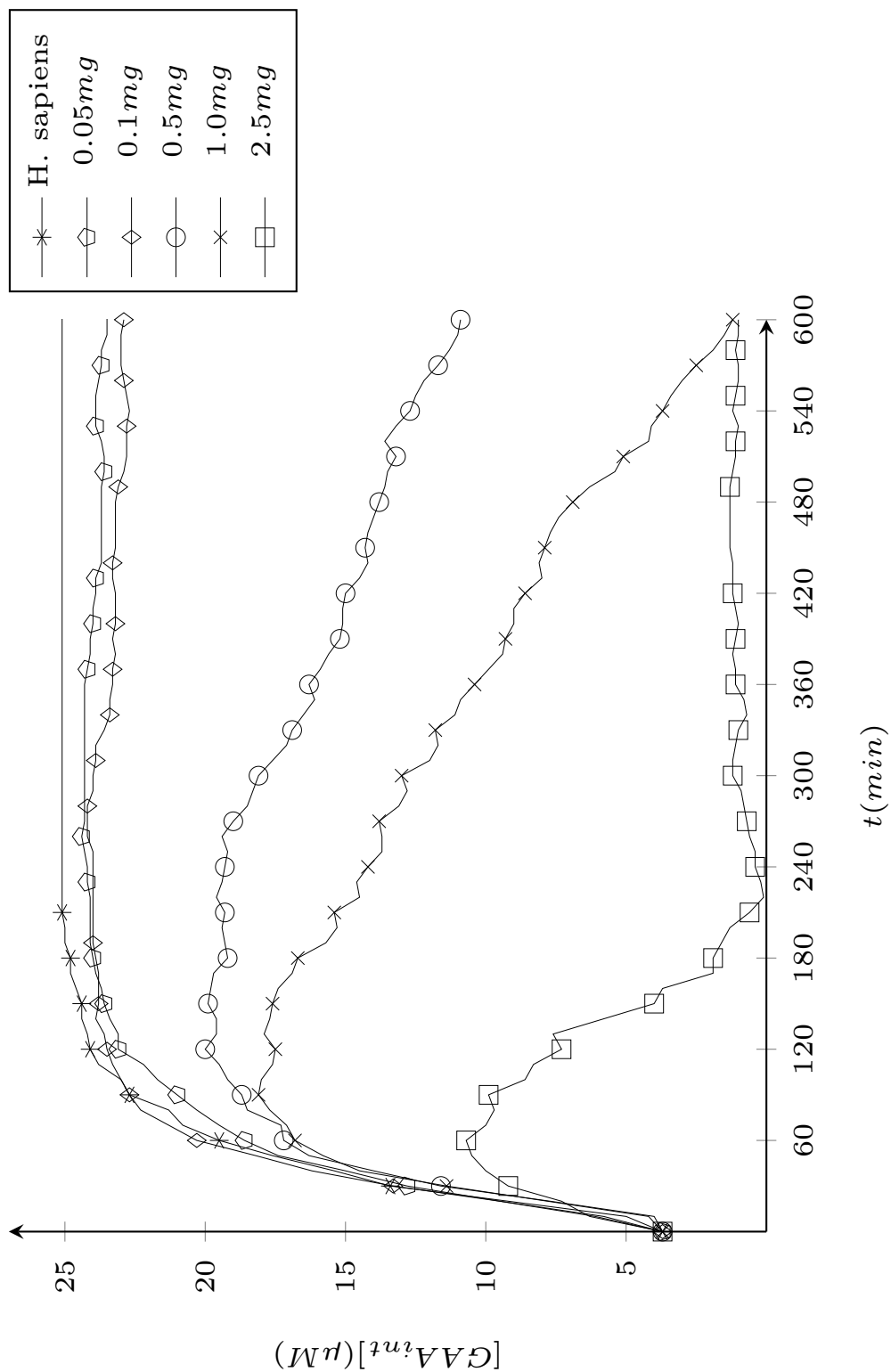


Figura 5.12: Consumo di guanidinoacetato negli eritrociti ingegnerizzati con quantità variabili di SAMS di *E. coli*

## Capitolo 6

# Model checking del trattamento della deficienza GAMT

In questo capitolo vengono descritte le difficoltà tecniche riscontrate durante il model checking (sez. 6.1), le proprietà CSL verificate sui modelli PRISM e l'esito del model checking per ognuna di esse (sez. 6.2 per le proprietà generate da Bio-PEPA e sez. 6.3 per le proprietà definite manualmente).

### 6.1 Limiti e requisiti del model checking

Poiché la sperimentazione del trattamento per la deficienza GAMT è ancora in fase iniziale, le simulazioni si sono rivelate sufficienti a riprodurre gli esperimenti effettuati. Sebbene possa essere interessante confrontare gli esiti simulativi con quelli di verifica formale, l'elevata dimensione del modello rende impraticabile tale confronto.

In termini predittivi, risulta di particolare interesse un'analisi di tipo prestazionale, che identifichi le reazioni limitanti della via metabolica, e di tipo temporale, per valutare l'efficacia della terapia. Oltre a tali proprietà, può essere interessante, a scopo didattico, verificare le proprietà predefinite da Bio-PEPA, le quali pur essendo di tipo quantitativo, tuttavia non forniscono informazioni aggiuntive rispetto alle simulazioni.

I modelli PRISM, generati a partire dal modello Bio-PEPA, vengono convertiti in CTMC di dimensioni intrattabili per un personal computer. Si è pertanto tentato di costruire le CTMC in macchine via via più ricche di memoria principale:

- macchina virtuale con 4 GB di RAM,
- macchina fisica con 16 GB di RAM,

- server virtuale con 64 GB di RAM,
- server virtuale con 128 GB di RAM.

Utilizzando l'engine *hybrid* (model checking ibrido) sul modello 4.14, le prime tre macchine non sono riuscite a portare a termine la costruzione del modello, andando in crash per memoria insufficiente. La quarta è invece riuscita a costruire un MTBDD da 94 GB in 7325.603 secondi, costituito da 38814405 nodi (di cui 409378 raggiungibili), equivalente a una CTMC con 42156649 stati e 125726277 transizioni, ma si è dovuto interrompere il calcolo dopo circa quindici giorni di blocco.

Un tentativo con l'engine *explicit* (model checking tradizionale) si è rivelato più fruttuoso, costruendo la matrice delle transizioni da 71 GB in 699.519 secondi. Tuttavia non è attualmente supportata la verifica di proprietà allo steady state (contenenti gli operatori  $\mathbb{S}$  o  $\mathbb{R}[\mathbb{S}]$ ) o contenenti operatori innestati (ad esempio  $\mathcal{FG}\phi$ ) per l'engine explicit.

## 6.2 Proprietà generate da Bio-PEPA

Relativamente al trattamento per la deficienza GAMT, sono stati generati automaticamente da Bio-PEPA:

- reward transitori relativi alle reazioni (*\_sams*, *\_gamt* e *\_uptake*), utili per contare quante volte avvengono;
- reward stazionari relativi alle specie (*\_SAM*, *\_SAH*, *\_GAA\_INT*, *\_GAA\_EXT* e *\_CR*), che permettono di quantificarle;
- reward quadratici (ad esempio *\_SAM\_squared* = *\_SAM*<sup>2</sup>), utili per il calcolo di varianze e scarti quadratici;
- reward basati sui contatori definiti (*\_UM\_SAM* = *\_SAM*/10, *\_UM\_SAH* = *\_SAH*/10, *\_UM\_GAA\_EXT* = *\_GAA\_EXT*/10, *\_UM\_GAA\_INT* = *\_GAA\_INT*/10 e *\_UM\_CR* = *\_CR*/10).

Sono stati inoltre generati i predicati:

- specie al massimo (ad esempio *\_SAM\_at\_maximum* : *\_SAM* = *MAX*), che rendono più leggibili le proprietà relative alla saturazione della specie;
- specie esaurite (ad esempio *\_SAM\_depleted* : *\_SAM* = 0), che rendono più leggibili le proprietà relative all'esaurimento.

La variabile *MAX* è definita, nei modelli PRISM, come la somma di tutte le quantità iniziali, per evitare di generare modelli infiniti. Vengono generate altre tre variabili, nel file di proprietà, da inizializzare a valori arbitrari prima della verifica formale:

- $T$  istante temporale;
- $i$  quantità di una specie;
- $rew$  reward generico.

Vista la ridotta quantità di informazioni aggiuntive ottenibili, rispetto alle simulazioni, dalle proprietà predefinite sono state inizializzate le variabili allo stesso valore per ogni proprietà:  $T = 600$  (10 ore),  $i = 10$  ( $1\mu\text{M}$ ),  $rew$  non utilizzata.

Di seguito vengono elencate le proprietà, il tempo di verifica e l'esito, relativamente al solo modello 4.14, che ha prodotto una CTMC da 42156649 stati, 125726277 transizioni e fattore di uniformizzazione pari a 17.4583; il coefficiente d'errore per l'algoritmo di Fox-Glynn è pari a  $1.25 \cdot 10^{-7}$ . I risultati vengono troncati alla quinta cifra decimale.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_SAM = i] \quad (6.1)$$

“Qual è la probabilità che ci siano  $i$  molecole di S-adenosil metionina al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 84309.627 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_SAM\_at\_maximum] \quad (6.2)$$

“Qual è la probabilità che l'S-adenosil metionina sia satura esattamente al tempo  $T$ ?”

Risultato: 0.92757

Tempo di elaborazione: 57043.331 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}\_SAM\_at\_maximum] \quad (6.3)$$

“Qual è la probabilità che l'S-adenosil metionina sia satura entro tempo  $T$ ?”

Risultato: 0.99999

Tempo di elaborazione: 67152.519 secondi.

$$\mathbb{R}_{\{\_SAM\}=?}[\mathcal{I} = T] \quad (6.4)$$

“Quanta S-adenosil metionina è presente al tempo  $T$ ?”

Risultato: 584.92193

Tempo di elaborazione: 65439.354 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_SAH = i] \quad (6.5)$$

“Qual è la probabilità che ci siano  $i$  molecole di S-adenosil omocisteina al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65624.93 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_SAH\_at\_maximum] \quad (6.6)$$

“Qual è la probabilità che l’S-adenosil omocisteina sia satura esattamente al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65311.243 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}\_SAH\_at\_maximum] \quad (6.7)$$

“Qual è la probabilità che l’S-adenosil omocisteina sia satura entro tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65306.103 secondi.

$$\mathbb{R}_{\{\_SAH\}=?}[\mathcal{I} = T] \quad (6.8)$$

“Quanta S-adenosil omocisteina è presente al tempo  $T$ ?”

Risultato: 345.72163

Tempo di elaborazione: 65339.864.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_GAA\_INT = i] \quad (6.9)$$

“Qual è la probabilità che ci siano  $i$  molecole di guanidinoacetato intracellulare al tempo  $T$ ?”

Risultato:  $2.69092 \cdot 10^{-13}$

Tempo di elaborazione: 65357.881 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_GAA\_INT\_at\_maximum] \quad (6.10)$$

“Qual è la probabilità che il guanidinoacetato intracellulare sia saturo esattamente al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65071.259 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}\_GAA\_INT\_at\_maximum] \quad (6.11)$$

“Qual è la probabilità che il guanidinoacetato intracellulare sia saturo entro il tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65088.718 secondi.

$$\mathbb{R}_{\{\_GAA\_INT\}=?}[\mathcal{I} = T] \quad (6.12)$$

“Quanto guanidinoacetato intracellulare è presente al tempo  $T$ ?”

Risultato: 96.15829

Tempo di elaborazione: 65317.685 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_GAA\_EXT = i] \quad (6.13)$$

“Qual è la probabilità che ci siano  $i$  molecole di guanidinoacetato plasmatico al tempo  $T$ ?”

Risultato:  $4.63631 \cdot 10^{-17}$

Tempo di elaborazione: 66352.333 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_GAA\_EXT\_at\_maximum] \quad (6.14)$$

“Qual è la probabilità che il guanidinoacetato plasmatico sia saturo esattamente al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65089.469 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}\_GAA\_EXT\_at\_maximum] \quad (6.15)$$

“Qual è la probabilità che il guanidinoacetato plasmatico sia saturo entro tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65337.322 secondi.

$$\mathbb{R}_{\{\_GAA\_EXT\}=?}[\mathcal{I} = T] \quad (6.16)$$

“Quanto guanidinoacetato plasmatico è presente al tempo  $T$ ?”

Risultato: 108.12003

Tempo di elaborazione: 66145.619 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_CR = i] \quad (6.17)$$

“Qual è la probabilità che ci siano  $i$  molecole di creatina al tempo  $T$ ?”

Risultato:  $4.09024 \cdot 10^{-161}$

Tempo di elaborazione: 65139.591 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{[T,T]}\_CR\_at\_maximum] \quad (6.18)$$

“Qual è la probabilità che la creatina sia satura esattamente al tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65127.878 secondi.

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}\_CR\_at\_maximum] \quad (6.19)$$

“Qual è la probabilità che la creatina sia satura entro il tempo  $T$ ?”

Risultato: 0

Tempo di elaborazione: 65093.444 secondi.

$$\mathbb{R}_{\{\_CR\}=?}[\mathcal{I} = T] \quad (6.20)$$

“Quanta creatina è presente al tempo  $T$ ?”

Risultato: 332.72163

Tempo di elaborazione: 65481.934 secondi.

$$\mathbb{R}_{\{\_UM\_SAM\}=?}[\mathcal{I} = T] \quad (6.21)$$

“Quanta S-adenosil metionina è presente al tempo  $T$ , in  $\mu M$ ?”

Risultato: 58.49219

Tempo di elaborazione: 65468.829 secondi.

$$\mathbb{R}_{\{\_UM\_SAH\}=?}[\mathcal{I} = T] \quad (6.22)$$

“Quanta S-adenosil omocisteina è presente al tempo  $T$ , in  $\mu M$ ?”

Risultato: 34.57216

Tempo di elaborazione: 66046.642 secondi.

$$\mathbb{R}_{\{\_UM\_GAA\_INT\}=?}[\mathcal{I} = T] \quad (6.23)$$

“Quanto guanidinoacetato intracellulare è presente al tempo  $T$ , in  $\mu M$ ?”

Risultato: 9.61583

Tempo di elaborazione: 65381.08 secondi.

$$\mathbb{R}_{\{\_UM\_GAA\_EXT\}=?}[\mathcal{I} = T] \quad (6.24)$$

“Quanto guanidinoacetato plasmatico è presente al tempo  $T$ , in  $\mu M$ ?”

Risultato: 10.812000

Tempo di elaborazione: 65364.411 secondi.

$$\mathbb{R}_{\{\_UM\_CR\}=?}[\mathcal{I} = T] \quad (6.25)$$

“Quanta creatina è presente al tempo  $T$ , in  $\mu M$ ?”

Risultato: 33.27216

Tempo di elaborazione: 65343.736 secondi.

$$\mathbb{R}_{\{\_sams\}=?}[\mathcal{C} \leq T] \quad (6.26)$$

“Quante volte è avvenuta la reazione ad opera della SAM sintasi entro il tempo  $T$ ?”

Risultato: N/A

Tempo di elaborazione: N/A

Risultato ottenuto per via simulativa: 882.566.

$$\mathbb{R}_{\{\_gamt\}=?}[\mathcal{C} \leq T] \quad (6.27)$$

“Quante volte è avvenuta la reazione ad opera della guanidinoacetato metil-transferasi entro il tempo  $T$ ?”

Risultato: N/A

Tempo di elaborazione: N/A

Risultato ottenuto per via simulativa: 332.971.

$$\mathbb{R}_{\{\_uptake\}=?}[C \leq T] \quad (6.28)$$

“Quante volte il guanidinoacetato è entrato dentro gli eritrociti entro il tempo  $T$ ?”

Risultato: N/A

Tempo di elaborazione: N/A

Risultato ottenuto per via simulativa: 392.443.

### 6.3 Proprietà definite

Di seguito sono descritte proprietà più interessanti di quelle generate automaticamente (anche in questo caso verificate per  $T = 600$  e *soglia* = 10).

Essendo l'S-adenosil metionina il reagente limitante, è necessario controllare e quantificare i casi in cui l'intera via metabolica vada in *starvation*, bloccando la produzione di creatina:

$$\mathbb{P}_{=?}[\mathcal{FG\_SAM} = 0] \quad (6.29)$$

“Qual è la probabilità che prima o poi l'S-adenosil metionina si esaurisca e rimanga a zero per sempre?”

Risultato: 0

Tempo di elaborazione: 342.842 secondi.

Anche nei casi in cui l'S-adenosil metionina non si esaurisca, è importante quantificare pure eventuali cali prestazionali e capacità di recupero da cali temporanei:

$$\mathbb{P}_{=?}[\mathcal{FG\_SAM} \leq \textit{soglia}] \quad (6.30)$$

“Qual è la probabilità che prima o poi l'S-adenosil metionina scenda sotto la *soglia* e vi rimanga per sempre?”

Risultato: 0

Tempo di elaborazione: 292.23 secondi.

$$\mathbb{P}_{=?}[\mathcal{G}(\_SAM = 0 \rightarrow \mathcal{F}^{\leq T}\_SAM > \textit{soglia})] \quad (6.31)$$

“Qual è la probabilità che, se si esaurisce l'S-adenosil metionina, questa riesca sempre a tornare oltre la *soglia* entro un tempo  $T$ ?”

Risultato: Proprietà non verificabile con l'engine explicit, né per via simulativa

Tempo di elaborazione: N/A.



Vista la citotossicità del guanidinoacetato, è interessante quantificare eventuali accumuli negli eritrociti (è importante tuttavia notare che il modello non rappresenta una possibile fuoriuscita del guanidinoacetato e che quindi l'esito della proprietà seguente potrebbe non essere significativo):

$$\mathbb{P}_{=?}[\mathcal{FG}^{\leq T}_{GAA\_INT} > soglia] \quad (6.32)$$

“Qual è la probabilità che il guanidinoacetato intracellulare rimanga sopra la *soglia* almeno per un tempo  $T$ ?”

Risultato: Proprietà non verificabile con l'engine explicit, né per via simulativa  
Tempo di elaborazione: N/A.

Per valutare l'efficacia della cura, è utile predire il tempo necessario a ripristinare livelli fisiologici di guanidinoacetato nel plasma:

$$\mathbb{P}_{=?}[\mathcal{F}^{\leq T}_{GAA\_EXT} \leq soglia] \quad (6.33)$$

“Qual è la probabilità che il guanidinoacetato plasmatico scenda sotto la *soglia* entro un tempo  $T$ ?”

Risultato:  $5.06063 \cdot 10^{-17}$

Tempo di elaborazione: 64474.518 secondi.

## Capitolo 7

# Conclusioni

### 7.1 Lavoro svolto

In questa tesi è stato descritto un approccio formale per l'analisi *in silico* di reti metaboliche. Sono stati presentati strumenti matematici in grado di descrivere fedelmente con un modello ogni aspetto di un metabolismo generico, con particolare enfasi su catene di Markov a tempo continuo, algebre dei processi e logiche temporali, e le tecniche per utilizzare tali modelli tramite simulazioni e verifiche formali.

È stata proposta un'applicazione di tale approccio allo studio della deficienza GAMT e di una cura in fase sperimentale, utilizzando gli strumenti software Bio-PEPA e PRISM. La cura presa in esame utilizza eritrociti ingegnerizzati da un apparato noto come Red Cell Loader per incapsulare l'enzima guanidinoacetato metiltransferasi allo scopo di sopperire alla mancanza della GAMT epatica dei pazienti affetti. Essendo la sperimentazione *in vitro* non ancora completata, l'analisi *in silico* proposta in questa tesi si è rivelata utile a indirizzare gli esperimenti futuri.

Più in dettaglio, l'analisi simulativa effettuata ha dimostrato la necessità di incapsulare, oltre alla GAMT, anche S-adenosil metionina sintasi clonata da *E. coli*, al fine di sopperire alla scarsa affinità della SAMS ematica e che porterebbe a un limitato effetto terapeutico, a causa della poca S-adenosil metionina a disposizione all'interno degli eritrociti. La verifica formale ha dimostrato che, una volta incapsulate sia SAMS che GAMT, non c'è rischio che il bioreattore costituito dagli eritrociti smetta di funzionare entro poche ore dall'infusione per esaurimento di S-adenosil metionina.

### 7.2 Necessità di un approccio formale integrato

L'approccio simulativo:

- + è ampiamente diffuso grazie alla sua semplicità di utilizzo;

- + è computazionalmente semplice, quindi produce risultati in tempi rapidi;
- + richiede un numero di risorse limitato;
- non è in grado di catturare tutti i comportamenti del sistema in maniera esaustiva;
- può accumulare errori in simulazioni particolarmente lunghe;
- richiede una media di più simulazioni per aumentare la precisione;
- è sensibile al tempo da simulare, aumentando le risorse richieste proporzionalmente al tempo;
- rende difficile l'analisi di casi particolari, che potrebbero addirittura passare inosservati se non sono noti a priori.

L'approccio basato su verifica:

- + analizza esaustivamente tutti i comportamenti del sistema, rilevando anche casi particolari non noti a priori;
- + è esente da errori di approssimazione;
- + è computazionalmente indipendente dal limite temporale imposto per la soddisfacibilità delle proprietà, richiedendo le stesse risorse per una proprietà istantanea o che dovrà verificarsi dopo miliardi di operazioni;
- è poco diffuso a causa della sua difficoltà di utilizzo;
- è computazionalmente complesso, richiedendo tempi elevati per ottenere un esito;
- è avido di risorse, a meno dell'utilizzo di tecniche simboliche, non sempre efficaci;

Risulta quindi utile (se non auspicabile) utilizzare un approccio che integri sia la componente simulativa che quella basata su verifica. I formalismi di modellazione si prestano agevolmente ad entrambi gli approcci e la visione d'insieme fornita è spesso in grado di rivelare caratteristiche non osservabili con uno solo dei due approcci.

## 7.3 Sviluppi futuri

Relativamente al caso di studio presentato, si prospetta la necessità di verificare la fedeltà delle simulazioni con gli esperimenti che verranno effettuati in futuro. Inoltre non è esclusa la possibilità di investigare in maniera formale nuovi aspetti che verranno portati alla luce dai nuovi esperimenti.

Per poter verificare le proprietà non modellabili dall'engine explicit di PRISM, può essere interessante introdurre approssimazioni che rendano i modelli di dimensione sufficientemente bassa da permettere la verifica tramite engine hybrid. In alternativa, quando l'engine explicit sarà in uno stato più maturo, potrà rivelarsi utile ripetere la verifica sui modelli non approssimati.

# Bibliografia

- [1] O. Akman, F. Ciocchetta, A. Degasperi e M. Guerriero. Modelling biological clocks with Bio-PEPA: stochasticity and robustness for the *Neurospora Crassa* circadian network. In *Computational Methods in Systems Biology*, pagine 52–67. Springer, 2009.
- [2] O. E. Akman, M. L. Guerriero, L. Loewe e C. Troein. Complementary approaches to understanding the plant circadian clock. *arXiv preprint arXiv:1002.4661*, 2010.
- [3] A. Aldini e M. Bernardo. Mixing logics and rewards for the component-oriented specification of performance measures. *Theoretical Computer Science*, 382(1):3–23, 2007.
- [4] L. S. Almeida, E. H. Rosenberg, C. Martinez-Munoz, N. M. Verhoeven, L. Vilarinho, C. Jakobs e G. S. Salomons. Overexpression of GAMT restores GAMT activity in primary GAMT-deficient fibroblasts. *Molecular Genetics and Metabolism*, 89(4):392–394, 2006.
- [5] A. Antonelli, C. Sfara, E. Manuali, I. J. Bruce e M. Magnani. Encapsulation of superparamagnetic nanoparticles into red blood cells as new carriers of MRI contrast agents. *Nanomedicine*, 6(2):211–223, 2011.
- [6] H. C. Araújo, W. Smit, N. M. Verhoeven, G. S. Salomons, S. Silva, R. Vasconcelos, H. Tomás, I. Almeida, C. Jakobs e M. Duran. Guanidinoacetate methyltransferase deficiency identified in adults and a child with mental retardation. *American Journal of Medical Genetics Part A*, 133(2):122–127, 2005.
- [7] C. Baier, B. Haverkort, H. Hermanns e J. P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on software engineering*, 29(6):524–541, 2003.
- [8] C. Baier, B. Haverkort, H. Hermanns e J. P. Katoen. On the logical characterisation of performability properties. In *ICALP*, volume 1853, pagine 780–792. Springer, 2000.
- [9] M. Bernardo e S. Botta. A survey of modal logics characterising behavioural equivalences for non-deterministic and stochastic systems. *Mathematical Structures in Computer Science*, 18(1):29–55, 2008.

- [10] S. Biagiotti, M. F. Paoletti, A. Fraternale, L. Rossi e M. Magnani. Drug delivery by red blood cells. *IUBMB Life*, 63(8):621–631, 2011.
- [11] D. Bošnjacki, T. E. Pronk e E. P. de Vink. In silico modelling and analysis of ribosome kinetics and aa-tRNA competition. *Transactions on Computational Systems Biology*:69–89, 2009.
- [12] BRENDA - Information on EC 2.1.1.2 - guanidinoacetate N-methyltransferase. URL: <http://brenda-enzymes.org/enzyme.php?ecno=2.1.1.2>.
- [13] BRENDA - Information on EC 2.5.1.6 - methionine adenosyltransferase. URL: <http://brenda-enzymes.org/enzyme.php?ecno=2.5.1.6>.
- [14] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *Computers, I. E. E. E. Transactions on*, 100(8):677–691, 1986.
- [15] R. Bundschuh, F. Hayot e C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophysical Journal*, 84(3):1606–1615, 2003.
- [16] C. Carducci, S. Santagata, V. Leuzzi, C. Carducci, C. Artiola, T. Giovanniello, R. Battini e I. Antonozzi. Quantitative determination of guanidinoacetate and creatine in dried blood spot by flow injection analysis-electrospray tandem mass spectrometry. *Clinica Chimica Acta*, 364(1):180–187, 2006.
- [17] T. Chen, M. Diciolla, M. Kwiatkowska e A. Mereacre. Quantitative verification of implantable cardiac pacemakers. In *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*, pagine 263–272. IEEE, 2012.
- [18] F. Ciocchetta, A. Degasperi, J. Heath e J. Hillston. Modelling and analysis of the NF- $\kappa$ B pathway in Bio-PEPA. *Transactions on Computational Systems Biology XII*:229–262, 2010.
- [19] F. Ciocchetta, A. Duguid e M. L. Guerriero. A compartmental model of the cAMP/PKA/MAPK pathway in Bio-PEPA. *arXiv preprint arXiv:0911.4984*, 2009.
- [20] F. Ciocchetta e J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, 2009.
- [21] F. Ciocchetta e J. Hillston. Bio-PEPA for epidemiological models. *Electronic Notes in Theoretical Computer Science*, 261:43–69, 2010.
- [22] E. M. Clarke e J. M. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4):626–643, 1996.
- [23] R. Cleaveland, J. Parrow e B. Steffen. The Concurrency Workbench: A semantics-based tool for the verification of concurrent systems. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 15(1):36–72, 1993.

- [24] F. Dannenberg, M. Kwiatkowska, C. Thachuk e A. J. Turberfield. DNA walker circuits: Computational potential, design, and verification. In *International Workshop on DNA-Based Computers*, pagine 31–45. Springer, 2013.
- [25] A. M. Das, K. Ullrich e D. Isbrandt. Upregulation of respiratory chain enzymes in guanidinoacetate methyltransferase deficiency. *Journal of Inherited Metabolic Disease*, 23(4):375–377, 2000.
- [26] M. P. Dobay, A. Dobay, J. Bantang e E. Mendoza. How many trimers? Modeling influenza virus fusion yields a minimum aggregate size of six trimers, three of which are fusogenic. *Molecular BioSystems*, 7(10):2741–2749, 2011.
- [27] W. M. Elseaidy, R. Cleaveland e J. W. Baugh. Modeling and verifying active structural control systems. *Science of Computer Programming*, 29(1-2):99–122, 1997.
- [28] E. A. Emerson e C. S. Jutla. The complexity of tree automata and logics of programs. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pagine 328–337. IEEE, 1988.
- [29] U. F. H. Engelke, M. Tassini, J. Hayek, M. de Vries, A. Bilos, A. Vivi, G. Valensin, S. Buoni, R. Zannolli, W. Brussel, B. Kremer, G. S. Salomons, M. J.B. M. Veendrick-Meekes, L. A. J. Kluijtmans, E. Morava e R. A. Wevers. Guanidinoacetate methyltransferase (GAMT) deficiency diagnosed by proton NMR spectroscopy of body fluids. *NMR in Biomedicine*, 22(5):538–544, 2009.
- [30] R. Ensenauer, T. Thiel, K. O. Schwab, U. Tacke, S. Stöckler-Ipsiroglu, A. Schulze, J. Hennig e W. Lehnert. Guanidinoacetate methyltransferase deficiency: differences of creatine uptake in human brain and muscle. *Molecular Genetics and Metabolism*, 82(3):208–213, 2004.
- [31] V. Ganesan, A. Johnson, A. Connelly, S. Eckhardt e R. A. H. Surtees. Guanidinoacetate methyltransferase deficiency: new clinical features. *Pediatric Neurology*, 17(2):155–157, 1997.
- [32] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
- [33] N. Gordon. Guanidinoacetate methyltransferase deficiency (GAMT). *Brain and Development*, 32(2):79–81, 2010.
- [34] M. Guerriero. Qualitative and quantitative analysis of a Bio-PEPA model of the gp130/jak/stat signalling pathway. *Transactions on Computational Systems Biology XI*:90–115, 2009.
- [35] S. L. Hamarat Baysal e A. H. Uslan. Encapsulation of Urease and PEG-Urease in erythrocyte. *Artificial Cells, Blood Substitutes, and Biotechnology*, 28(3):263–271, 2000.

- [36] H. Hansson e B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
- [37] Z. Har’El e R. P. Kurshan. Software for analytical development of communications protocols. *Bell Labs Technical Journal*, 69(1):45–59, 1990.
- [38] J. Heath, M. Kwiatkowska, G. Norman, D. Parker e O. Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theoretical Computer Science*, 391(3):239–257, 2008.
- [39] J. Ilas, A. Mühl e S. Stöckler-Ipsiroglu. Guanidinoacetate methyltransferase (GAMT) deficiency: non-invasive enzymatic diagnosis of a newly recognized inborn error of metabolism. *Clinica Chimica Acta*, 290(2):179–188, 2000.
- [40] T. Kikuchi, Y. Orita, A. Ando, H. Mikami, M. Fujii, A. Okada e H. Abe. Liquid-chromatographic determination of guanidino compounds in plasma and erythrocyte of normal persons and uremic patients. *Clinical Chemistry*, 27(11):1899–1902, 1981.
- [41] S. Kim. S-adenosylmethionine: protein-carboxyl methyltransferase from erythrocyte. *Archives of Biochemistry and Biophysics*, 161(2):652–657, 1974.
- [42] E. L. King e C. Altman. A schematic method of deriving the rate laws for enzyme-catalyzed reactions. *The Journal of Physical Chemistry*, 60(10):1375–1378, 1956.
- [43] C. P. Ku e H. Passow. Creatine and creatinine transport in old and young human red blood cells. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 600(1):212–227, 1980.
- [44] R. P. Kurshan. The complexity of verification. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, pagine 365–371. ACM, 1994.
- [45] M. Kwiatkowska, G. Norman e D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *International Conference on Computer Aided Verification*, pagine 585–591. Springer, 2011.
- [46] M. Kwiatkowska, G. Norman e D. Parker. Probabilistic Model Checking: Advances and Applications. In *Formal System Verification*, pagine 73–121. Springer, 2018.
- [47] M. Kwiatkowska, G. Norman e D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)*, 6(2):128–142, 2004.



- [48] M. Magnani, A. Casabianca, A. Fraternale, G. Brandi, S. Gessani, R. Williams, M. Giovine, G. Damonte, A. De Flora e U. Benatti. Synthesis and targeted delivery of an azidothymidine homodinucleotide conferring protection to macrophages against retroviral infection. *Proceedings of the National Academy of Sciences*, 93(9):4403–4408, 1996.
- [49] M. Magnani, F. Pierigè e L. Rossi. Erythrocytes as a novel delivery vehicle for biologics: from enzymes to nucleic acid-based therapeutics. *Therapeutic Delivery*, 3(3):405–414, 2012.
- [50] M. Magnani, L. Rossi, G. Brandi, G. F. Schiavano, M. Montroni e G. Piedimonte. Targeting antiretroviral nucleoside analogues in phosphorylated form to macrophages: in vitro and in vivo studies. *Proceedings of the National Academy of Sciences*, 89(14):6477–6481, 1992.
- [51] M. Magnani, L. Rossi, M. D’ascenzo, I. Panzani, L. Bigi e A. Zanella. Erythrocyte engineering for drug delivery and targeting. *Biotechnology and Applied Biochemistry*, 28(1):1–6, 1998.
- [52] K. L. McMillan. Symbolic model checking. In *Symbolic Model Checking*, pagine 25–60. Springer, 1993.
- [53] A. G. Mikati, I. A. Gheida, A. Shamseddine, M. A. Mikati e P. E. Karam. Epileptic and electroencephalographic manifestations of guanidinoacetate-methyltransferase deficiency. *Epileptic Disorders*, 15(4):407–416, 2013.
- [54] A. Mori. Biochemistry and neurotoxicology of guanidino compounds. *Integrative Physiological and Behavioral Science*, 22(3):85–94, 1987.
- [55] K. L. Oden e S. Clarke. S-adenosyl-L-methionine synthetase from human erythrocytes: role in the regulation of cellular S-adenosylmethionine levels. *Biochemistry*, 22(12):2978–2986, 1983.
- [56] D. J. O’Rourke, S. Ryan, G. Salomons, C. Jakobs, A. Monavari e M. D. King. Guanidinoacetate methyltransferase (GAMT) deficiency: late onset of movement disorder and preserved expressive language. *Developmental Medicine & Child Neurology*, 51(5):404–407, 2009.
- [57] N. D. L. Owens, J. Timmis, A. Greensted e A. Tyrrell. Modelling the tunability of early t cell signalling events. *Lecture Notes in Computer Science*, 5132:12–23, 2008.
- [58] D. A. Parker. *Implementation of symbolic model checking for probabilistic systems*. Tesi di dottorato, University of Birmingham, 2003.
- [59] Part 10 - Implementation of Probabilistic Model Checking. URL: <http://www.prismmodelchecker.org/lectures/biss07/10-implementation.pdf>.
- [60] Part 5 - Continuous-time Markov chains. URL: <http://www.prismmodelchecker.org/lectures/biss07/05-ctmcs.pdf>.

- [61] D. Peled. Combining partial order reductions with on-the-fly model-checking. In *Computer Aided Verification*, pagine 377–390. Springer, 1994.
- [62] PEPA. URL: <http://www.dcs.ed.ac.uk/pepa>.
- [63] C. Pezzoli, G. Stramentinoli, M. Galli-Kienle e E. Pfaff. Uptake and metabolism of S-adenosyl-L-methionine by isolated rat hepatocytes. *Biochemical and Biophysical Research Communications*, 85(3):1031–1038, 1978.
- [64] PRISM - Case Studies - Cell Cycle Control in Eukaryotes. URL: <http://www.prismmodelchecker.org/casestudies/cyclin.php>.
- [65] PRISM - Tutorial - Circadian clock. URL: <http://www.prismmodelchecker.org/tutorial/circadian.php>.
- [66] PRISM Manual | Property Specification / Reward-based Properties. URL: <http://www.prismmodelchecker.org/manual/PropertySpecification/Reward-basedProperties>.
- [67] Probabilistic verification and synthesis. URL: <http://www.prismmodelchecker.org/courses/kth15/marta-kth15-p3.pdf>.
- [68] W. Renema, A. Schmidt, J. J. A. van Asten, F. Oerlemans, K. Ullrich, B. Wieringa, D. Isbrandt e A. Heerschap. MR spectroscopy of muscle and brain in guanidinoacetate methyltransferase (GAMT)-deficient mice: Validation of an animal model to study creatine deficiency. *Magnetic Resonance in Medicine*, 50(5):936–943, 2003.
- [69] L. Rossi, G. Brandi, G. F. Schiavano, S. Scarfi, E. Millo, G. Damonte, U. Benatti, A. De Flora e M. Magnani. Heterodimer-loaded erythrocytes as bioreactors for slow delivery of the antiviral drug azidothymidine and the antimycobacterial drug ethambutol. *AIDS Research and Human Retroviruses*, 15(4):345–353, 1999.
- [70] L. Rossi, S. Serafini, L. Cenerini, F. Picardi, L. Bigi, I. Panzani e M. Magnani. Erythrocyte-mediated delivery of dexamethasone in patients with chronic obstructive pulmonary disease. *Biotechnology and Applied Biochemistry*, 33(2):85–89, 2001.
- [71] L. Rossi, S. Serafini, F. Pierigé, A. Antonelli, A. Cerasi, A. Fraternale, L. Chiarantini e M. Magnani. Erythrocyte-based drug delivery. *Expert Opinion on Drug Delivery*, 2(2):311–322, 2005.
- [72] K. R. Sanft, S. Wu, M. Roh, J. Fu, R. K. Lim e L. R. Petzold. StochKit2: software for discrete stochastic simulation of biochemical systems with events. *Bioinformatics*, 27(17):2457–2458, 2011.

- [73] A. Schmidt, B. Marescau, E. A. Boehm, W. K. J. Renema, R. Peco, A. Das, R. Steinfeld, S. Chan, J. Wallis, M. Davidoff, R. Waldschütz, A. Heerschap, P. P. De Deyn, S. Neubauer e D. Isbrandt. Severely altered guanidino compound levels, disturbed body weight homeostasis and impaired fertility in a mouse model of guanidinoacetate N-methyltransferase (GAMT) deficiency. *Human Molecular Genetics*, 13(9):905–921, 2004.
- [74] A. Schulze, F. Ebinger, D. Rating e E. Mayatepek. Improving treatment of guanidinoacetate methyltransferase deficiency: reduction of guanidinoacetic acid in body fluids by arginine restriction and ornithine supplementation. *Molecular Genetics and Metabolism*, 74(4):413–419, 2001.
- [75] A. Schulze, E. Mayatepek, D. Rating e H. J. Bremer. Sakaguchi reaction: A useful method for screening guanidinoacetate-methyltransferase deficiency. *Journal of Inherited Metabolic Disease*, 19(5):706–706, 1996.
- [76] Session 3: simulation vs. Formal Verification. URL: <https://embedded.eecs.berkeley.edu/research/vis/ttc/lecDir/html/session3>.
- [77] Simulating Markov chains. URL: <http://www.columbia.edu.ks20/4703-Sigman/4703-07-Notes-MC.pdf>.
- [78] S. Stockler-Ipsiroglu, C. van Karnebeek, N. Longo, G. C. Korenke, S. Mercimek-Mahmutoglu, I. Marquart, B. Barshop, C. Grolik, A. Schlune, B. Angle, H. C. Araújo, T. Coskun, L. Diogo, M. Geraghty, G. Haliloglu, V. Konstantopoulou, V. Leuzzi, A. Levtova, J. MacKenzie, B. Maranda, A. A. Mhanni, G. Mitchell, A. Morris, T. Newlove, D. Renaud, F. Scaglia, V. Valayannopoulos, F. J. van Spronsen, K. T. Verbruggen, N. Yuskiv, W. Nyhan e A. Schulze. Guanidinoacetate methyltransferase (GAMT) deficiency: outcomes in 48 individuals and recommendations for diagnosis, treatment and monitoring. *Molecular Genetics and Metabolism*, 111(1):16–25, 2014.
- [79] G. Stramentinoli, C. Pezzoli e M. G. Kienle. Uptake of S-adenosyl-L-methionine by rabbit erythrocytes. *Biochemical Pharmacology*, 27(10):1427–1430, 1978.
- [80] M. Tachikawa e K. Hosoya. Transport characteristics of guanidino compounds at the blood-brain barrier and blood-cerebrospinal fluid barrier: relevance to neural disorders. *Fluids and Barriers of the CNS*, 8(1):13, 2011.
- [81] M. Tarnopolsky, G. Parise, M. Fu, A. Brose, A. Prasad, O. Speer e T. Wallimann. Acute and moderate-term creatine monohydrate supplementation does not affect creatine transporter mRNA or protein content in either young or elderly humans. *Molecular and Cellular Biochemistry*, 244(1-2):159–166, 2003.

- 
- [82] The exponential distribution has the memoryless (forgetfulness) property. URL: <http://www.math.wm.edu/leemis/chart/UDR/PDFs/ExponentialForgetfulness.pdf>.
  - [83] The geometric distribution has the memoryless (forgetfulness) property. URL: <http://www.math.wm.edu/leemis/chart/UDR/PDFs/GeometricF.pdf>.
  - [84] R. T. Usry, G. L. Moore e F. W. Manalo. Morphology of stored, rejuvenated human erythrocytes. *Vox Sanguinis*, 28(3):176–183, 1975.
  - [85] M. Y. Vardi e P. Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*, pagine 322–331. IEEE Computer Society, 1986.
  - [86] W. Visser e H. Barringer. Practical CTL\* model checking: Should SPIN be extended? *International Journal on Software Tools for Technology Transfer*, 2(4):350–365, 2000.
  - [87] J. Vodopiutz, C. B. Item, M. Häusler, H. Korall e O. A. Bodamer. Severe speech delay as the presenting symptom of guanidinoacetate methyltransferase deficiency. *Journal of Child Neurology*, 22(6):773–774, 2007.

# Ringraziamenti

Si ringraziano i professori Jane Hillston e Stephen Gilmore per il loro aiuto sulla sintassi di Bio-PEPA.

Si ringraziano i dott.ri Marco Cappellacci, Paolo Cecchini e Andrea Seraghiti e la Sezione di Scienze e Tecnologie dell'Informazione dell'Università di Urbino per aver messo a disposizione una macchina sufficientemente potente per l'applicazione delle tecniche descritte nel capitolo 6. Si ringrazia infine la prof.ssa Simona Scacchi per la correzione delle bozze.