NNT/NL : 2020AIXM0609/070ED352

DOCTORAL THESIS

Defended at the University of Camerino (UNICAM)

as part of the cotutelle with the Aix-Marseille University (AMU)

on 28 December 2020 by

Stefano Maestri

Thesis title:

PROCESS-BASED MODELLING OF RNA AND PROTEIN INTERACTIONS: A FORMAL APPROACH

Modelling the effects of long-range forces in biological systems to better understand the global behaviour of molecular interactions

Discipline
Science and Technology (UNICAM)
Physique et Sciences de la Matière (AMU)

Speciality
Computer Science (UNICAM)
Physique Théorique et Mathématique (AMU)

PhD school
School of Advanced Studies (UNICAM)
Physique et sciences de la matière - 352 (AMU)

Laboratory/Research partners
Bioshape and Data Science Lab (UNICAM)
CPT - Centre de Physique Théorique (AMU)

Jury members

Paul Bourgine CNRS	Referee/Examiner
Jack TUSZYNSKI University of Alberta	Referee
Elena Floriani AMU	Examiner
Andrea Omicini University of Bologna	Examiner
Sandra Pucciarelli UNICAM	Examiner
Emanuela Merelli UNICAM	Thesis supervisor
Marco Pettini AMU	Thesis co-supervisor

NNT/NL : 2020AIXM0609/070ED352

THÈSE DE DOCTORAT

Soutenue à l'Université de Camerino (UNICAM)

dans le cadre d'une cotutelle avec Aix-Marseille Université (AMU)

le 28 décembre 2020 par

Stefano MAESTRI

Titre de la thèse:

MODÉLISATION PAR PROCESSUS DES INTERACTIONS ENTRE ARN ET PROTÉINES: UNE APPROCHE FORMELLE

Modélisation des effets des forces à longue portée dans les systèmes biologiques pour mieux comprendre le comportement global des interactions moléculaires

Discipline
Science and Technology (UNICAM)
Physique et Sciences de la Matière (AMU)

Spécialité
Computer Science (UNICAM)
Physique Théorique et Mathématique (AMU)

École doctorale
School of Advanced Studies (UNICAM)
Physique et sciences de la matière - 352 (AMU)

Laboratoire/Partenaires de recherche
Bioshape and Data Science Lab (UNICAM)
CPT - Centre de Physique Théorique (AMU)

Composition du jury

Paul BOURGINE CNRS	Rapporteur/Examinateur
Jack TUSZYNSKI Université de l'Alberta	Rapporteur
Elena FLORIANI AMU	Examinatrice
Andrea OMICINI Université de Bologne	Examinateur
Sandra PUCCIARELLI UNICAM	Examinatrice
Emanuela MERELLI UNICAM	Directrice de thèse
Marco PETTINI AMU	Co-directeur de thèse

Affidavit

I, undersigned, Stefano Maestri, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific direction of Emanuela Merelli and Marco Pettini, in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body, except where part of an authored or co-authored publication has been included. In the latter case, my contribution and those of any other authors to the work have been explicitly indicated. I declare that appropriate credit has been given within this manuscript where reference has been made to the work of others.

Camerino, 25/11/2020

Abstract

Process algebras and agent-based models have proven to be effective methods for studying biological systems. Our research employs such techniques to investigate the behaviours that characterise biological macromolecules and reveal the global properties of biochemical processes resulting from local molecular interactions. This dissertation consists of two parts. In the first one, we use formal methods, such as the Calculus of Communicating Systems, to demonstrate the existence of a congruence level at which the folding of RNA molecules is behaviourally equivalent to that of proteins. This finding allows us to hypothesise the role that RNA functional complexity played during the evolutionary process that led proteins to emerge as the primary catalysts in modern cells. We also rely on such a representation to model how an error in the genetic code–i.e., a mutation–can propagate through each step of the synthesis of a new protein, ultimately affecting its folded conformation. We formally prove that the different complexity of RNA and protein folding results in significantly dissimilar impacts that a single nucleotide mutation can have on the structures of proteins compared to those of RNAs. In the second part of this manuscript, we describe an agent-based approach that we specially designed to investigate the global behaviour of long-distance electrodynamic interactions among biomolecules. Agents are software entities that can perceive their environment and operate on it autonomously. Using agent-oriented programming, we created a software replica of glycolysis–the metabolic process that provides energy to cells through glucose oxidation. The ability of agents to reproduce molecular behaviours makes it possible to study biochemical processes in a virtual environment and interpret them as the result of underlying molecular interactions. Furthermore, the generated agent interaction matrix can be filtered using topological data analysis, allowing us to investigate the role of 2-simplex formation in biochemical reactions. Our goal is to understand how specific types of molecular interactions influence glycolysis effectiveness, particularly in cancer cells. The two parts that make up our work represent the main phases of the engineering life cycle for the simulation of enzyme behaviour; they are intended as the preliminary steps in the development of a computational framework able to contribute to cancer studies performed on experimental data. This research sheds new light on how biomolecules interact and lays the groundwork for in silico personalised and precision medicine.

Résumé

Les algèbres de processus et les modèles à base d'agents se sont révélés être des méthodes efficaces pour étudier les systèmes biologiques. Notre recherche utilise de telles techniques pour étudier les comportements qui caractérisent les macromolécules biologiques et révéler les propriétés globales des processus biochimiques résultant des interactions moléculaires locales. Cette thèse se compose de deux parties. Dans la première, nous utilisons des méthodes formelles, telles que le calcul des systèmes communicants, pour prouver l'existence d'un niveau de congruence auquel le repliement de l'ARN est comportementalement équivalent à celui des protéines. Cette découverte nous permet d'émettre l'hypothèse du rôle que la complexité fonctionnelle de l'ARN a joué au cours du processus évolutif qui a conduit les protéines à émerger en tant que catalyseurs primaires dans les cellules modernes. Nous nous appuyons également sur une telle représentation pour modéliser comment une erreur dans le code génétique – c'est-à-dire une mutation – peut se propager à chaque étape de la synthèse d'une nouvelle protéine, affectant finalement sa conformation repliée. Nous démontrons formellement que la complexité différente du repliement de l'ARN et des protéines entraîne un impact significativement différent qu'une seule mutation de nucléotide peut avoir sur les structures des protéines par rapport à celles des ARN. Dans la seconde partie de ce manuscrit, nous décrivons une approche à base d'agents que nous avons spécialement conçue pour étudier le comportement global des interactions électrodynamiques à longue distance entre les biomolécules. Les agents sont des entités logicielles capables de percevoir leur environnement et d'y opérer de manière autonome. À l'aide d'une programmation orientée agent, nous avons créé une réplique logicielle de la glycolyse – le processus métabolique qui fournit de l'énergie aux cellules par l'oxydation du glucose. La capacité des agents à reproduire des comportements moléculaires permet d'étudier des processus biochimiques dans un environnement virtuel et de les interpréter comme le résultat d'interactions moléculaires sous-jacentes. De plus, la matrice d'interaction d'agent générée peut être filtrée à l'aide de l'analyse topologique de données, ce qui nous permet d'étudier le rôle de la formation de 2-simplexes dans les réactions biochimiques. Notre objectif est de comprendre comment des types spécifiques d'interactions moléculaires influencent l'efficacité de la glycolyse, en particulier dans les cellules cancéreuses. Les deux parties qui composent notre travail représentent les principales phases du cycle de vie de l'ingénierie pour la simulation du comportement enzymatique ; elles sont conçues comme les étapes préliminaires du développement d'un cadre informatique capable de contribuer aux études sur le cancer réalisées sur des données expérimentales. Cette recherche apporte un nouvel éclairage sur l'interaction des biomolécules et jette les bases d'une médecine in silico personnalisée et de précision.

Acknowledgements

First and foremost, I would like to thank my supervisors, Prof. Emanuela Merelli and Prof. Marco Pettini, for their invaluable guidance, unwavering support, and patience throughout my PhD. Their vast knowledge and wealth of experience have inspired me both in my academic studies and in daily life. I would also like to thank Prof. Sandra Pucciarelli for her assistance with some of the biological issues that came up during my research. I extend my sincere thanks to everyone at the Bioshape and Data Science Lab, especially Michela Quadrini, who has been a true friend and a precious guide through the many unwritten rules of university life. I am also grateful to my aunt Cinzia for her advice on English grammar and style and to my friend Fulvia for helping me with graphics and web design (and also for putting up with my whining about my PhD troubles). Finally, I would like to express my heartfelt gratitude to my parents, my sister, and my beloved cat; I would not have been able to complete my PhD without their wonderful understanding and support over the years.

Contents

List of Figures

List of Tables

Introduction

Our understanding of biological systems is often hampered by the complexity of the underlying molecular interactions, which blurs the relationships that link basic phenomena to a process as a whole when investigated from a top-down perspective. To use Aristotle's words, "*in the case of all things which have several parts and in which the totality is not, as it were, a mere heap, but the whole is something besides the parts, there is a cause*" [7]. If we take a reductionist approach, this cause may remain hidden. We can undoubtedly acquire relevant knowledge about each considered structure by recursively decomposing a complex biological system down to its primary elements. However, the behaviour of the entire system is understandable only if we are able to grasp its global properties [4, 53]. According to this view, sophisticated biological functions originate from simple local rules that govern how the system's basic components interact.

This dissertation builds on such premises to examine the behaviours that characterise biological macromolecules, ranging from the steps that lead some of them to a three-dimensional conformation to the way they interact with one another. We use algebraic modelling to provide a formal definition of the local interactions carried out by nucleotides in an RNA molecule and amino acids in a protein's polypeptide chain; identifying their collective properties in the expression of a fully functional macromolecule reveals congruences and dissimilarities, which, in some cases, can be associated with genetic pathologies. We also analyse the global behaviour of long-distance electrodynamic interactions in metabolic pathways through a specifically designed agent-based approach. Experimental evidence proves that random encounters and short-range potentials might not be sufficient to explain the high efficiency of biochemical reactions in living cells [45, 81, 95]. However, while the latest in vitro studies are limited by present-day technology, agent-based simulations provide an in silico support to the outcomes hitherto obtained and elucidate behaviours not yet well understood. The core idea of our work is to show how algebraic and agent-based methods are well suited to uncover complex phenomena in biological systems and shed new light on the interpretation of genetic diseases.

The following sections provide a brief overview and contextualisation of the topics covered in the remainder of this dissertation, which is divided into two main parts: the first focuses on the algebraic models of RNAs and proteins, while the second describes our agent-based study on biomolecular interactions. Although these two approaches can be related to each other (as shown in Chapter 4), we separate them to help the reader distinguish the work conducted mainly within the context of the University of Camerino (Part I) from the results of the collaboration with the Aix-Marseilles University (Part II).

Algebraic Modelling of RNA and Proteins

The relationship between structure and function is a relevant topic in biology, whose investigation received a significant contribution from different computational approaches [13, 27, 69, 73, 89]. In particular, formal languages and graph grammars have been successfully applied in modelling the properties that correlate the functions expressible by RNA molecules and the specific substructures involved in their folding [68, 96]. The latter plays a fundamental role in this analysis because it allows a linear biopolymer to reach a three-dimensional conformation (by forming hydrogen bonds between nonconsecutive monomers).

In this manuscript, we push that idea further and prove that the complexity of RNA functions can be traced back to the inner potentiality of each nucleotide to interact with others in the same sequence. This result is obtained by comparing the folding of RNA with that performed by proteins to identify an abstraction level at which these two classes of molecules show the same structural and functional complexity. We refer to this level as the *congruence level*; its characterisation is possible due to the expressiveness of process algebras [1], through which we model the folding of both RNA and proteins.

During the second half of the last century, the investigation of the reasons for the existence of such similar molecules led to the formulation of the RNA world hypothesis: RNA might be a "fossil" of an RNA world that existed on Earth before modern cells appeared, in which RNA fulfilled the roles of both DNA and proteins [44]. This theory is still highly debated, as, beyond their similarities, proteins and RNAs show profound structural differences that affect how they perform their functions [93]. In the first part of this dissertation, we formally compare the folding process of proteins with that of RNAs. We focus our study on the interactions carried out by the elementary units that make up RNAs and proteins, describing the whole folding as the resulting behaviour of such interactions; by highlighting their fundamental properties, we aim to identify clues to the validity of the RNA world hypothesis.

We then concentrate on a class of pathologies that impact the folding process. This part of our study starts with a formal description of how such pathologies originate as an error in the genetic code (a mutation, in biological terms) and can propagate through each step of gene expression, affecting both RNA and protein structures. We model how the mutation of even a single nucleotide (point mutation) can alter the final conformation of a protein while it is harmless to the structure of RNAs; we also show that a well-known genetic disease, sickle cell anaemia, can be considered a global behaviour of both amino acid and nucleotide interactions.

We finally take another step forward by hypothesising the biological functions that characterise the *congruence level* mentioned above and further exploring the applicability of process algebras to describe its properties. The resulting models will ultimately form the basis for an agent-based simulation [57]. Agents are discrete software elements whose interactions correspond to those performed by the components of the modelled system fairly faithfully to the actual behaviour of a biological process [75]. In process algebras, processes are concurrent, autonomous, and reactive; all these properties are also shared by agents, making process algebras suitable specification languages for agent-based systems.

Agent-based Modelling and Simulation of Biomolecular Interactions

The second part of this manuscript describes an agent-based simulator developed to study the molecular interactions that characterise metabolic pathways and analyse their global properties [22]. We simulated complete enzymatic reactions by modelling the involved molecules (enzymes, metabolites, and complexes) as autonomous and interactive agents.

In vitro studies show that a biological macromolecule behaves like an oscillating dipole, and long-range forces can be activated between two resonant molecular systems because a charge that oscillates at high frequency (in the range of $10^{10} - 10^{11}$ Hz) is not affected by Debye screening [45, 81]. We aim to provide an in silico validation to these experiments through agent-based simulations, where each molecule is represented by an agent able to perceive the environment and the cognate partners with which it can interact. A similar result may be obtained through a molecular dynamics model; however, this method often places the analysis at the atomic level, and related simulations require a priori knowledge of a large number of experimental parameters. Instead, agent-based models and simulations allow the study to be conducted at an abstraction level that can be represented with a reasonably small amount of empirical data without loss of accuracy when reproducing a macromolecular behaviour. We explore the simulator's ability to deal with the long-distance electrodynamic interactions that shape the behaviour of biomolecular systems, thus allowing us to analyse their effect on the evolution of a metabolic pathway, such as yeast glycolysis.

However, understanding and representing as a whole the agent dynamics characterising a reaction made by a large number of molecules still constitutes a considerable challenge. For this reason, we define a new visualisation paradigm based on the concept of *interaction-as-perception*: whenever a molecule perceives a cognate partner, a potential link between the two is established. In this way, we can derive the graph of perceptions at each simulation time step; on those graphs, we apply the topological data analysis to capture the 3-body interactions by interpreting 2-simplices–convex hulls of three points–as observable structures. We use 2-simplex formation as a valid semantic to represent the global dynamics of the system.

Organisation of the Manuscript

Each of the two main parts of this manuscript is correlated with an introductory chapter (Chapters 1 and 5, respectively), which describes the basic biological and theoretical concepts needed to better understand our study and the methodology adopted. The following chapters go into detail on the results we have obtained.

The first part comprises Chapters 1 to 4:

- In Chapter 1, we provide some basic knowledge on gene expression and RNA and protein folding; we also introduce the formal methods adopted for modelling these biological processes: Calculus of Communicating Systems (CCS), labelled transition systems (LTSs), and Hennessy-Milner logic (HML).
- In Chapter 2, we take advantage of CCS and LTS to model the folding of both RNA and proteins and demonstrate how it is possible to formally define a level of abstraction in which such processes show behavioural equivalence (*congruence level*). Its definition allows us to hypothesise some of the reasons that led the evolution of life to form proteins and use them as the main catalysts in biological processes.
- Chapter 3 analyses a class of pathologies that affect the folding processes to study how the dissimilarities between the structural components of proteins and RNAs cause different responses to an alteration of the correct folding pathway.
- In Chapter 4, we explore the expressiveness of CCS in modelling the functions representing the behaviour of non-coding RNA molecules, intended as a characterisation of the congruence level defined in Chapter 2. Based on these results, we propose a suitable methodology to generate an algebraic specification for agent-based simulations.

The second part of this manuscript consists of Chapters 5 to 7:

- Chapter 5 introduces the fundamental steps of glycolysis and the agent-based simulator, Orion, we developed for studying this process in terms of molecular interactions.
- Chapter 6 describes how we adapted Orion to simulate long-distance molecular interactions in metabolic reactions and analyse how they affect glycolysis efficiency.
- In Chapter 7, we take a step forward by using agent-based simulations to reproduce three-body dynamics in a biochemical reaction, thus visualising and understanding its global behaviour; this is possible by applying the *interaction-as-perception* paradigm.

Introduction (en français)

Notre compréhension des systèmes biologiques est souvent entravée par la complexité des interactions moléculaires sous-jacentes, qui brouille les relations reliant les phénomènes de base à un processus dans son ensemble lorsqu'il est étudié dans une perspective descendante (top-down). Pour reprendre les mots d'Aristote, « Il y a une cause à l'unité de ce qui a plusieurs parties dont la réunion n'est point une sorte de monceau, de tout ce dont l'ensemble est quelque chose indépendamment des parties » [6]. Si nous adoptons une approche réductionniste, cette cause peut rester cachée. On peut sans doute acquérir des connaissances pertinentes sur chaque structure considérée en décomposant récursivement un système biologique complexe jusqu'à ses éléments primaires. Cependant, le comportement de l'ensemble du système n'est compréhensible que si nous sommes capables de saisir ses propriétés globales [4, 53]. Selon ce point de vue, fonctions biologiques sophistiquées proviennent de règles locales simples qui régissent la manière dont les composants de base du système interagissent.

Cette thèse s'appuie sur ces prémisses pour examiner les comportements qui caractérisent les macromolécules biologiques, allant des étapes qui conduisent certaines d'entre elles à une conformation tridimensionnelle à la façon dont elles interagissent les unes avec les autres. Nous utilisons la modélisation algébrique pour fournir une définition formelle des interactions locales réalisées par les nucléotides dans une molécule d'ARN et les acides aminés dans la chaîne polypeptidique d'une protéine ; l'identification de leurs propriétés collectives dans l'expression d'une macromolécule pleinement fonctionnelle révèle des congruences et des dissemblances, qui, dans certains cas, peuvent être associées à des pathologies génétiques. Nous analysons également le comportement global des interactions électrodynamiques à longue distance dans les voies métaboliques grâce à une approche à base d'agents spécifiquement conçue. Des preuves expérimentales montrent que les rencontres aléatoires et les potentiels à courte portée pourraient ne pas être suffisants pour expliquer la grande efficacité des réactions biochimiques dans les cellules vivantes [45, 81, 95]. Cependant, alors que les dernières études in vitro sont limitées par la technologie actuelle, les simulations à base d'agents fournissent un support in silico aux résultats obtenus jusqu'à présent et élucident des comportements pas encore bien compris. L'idée centrale de notre travail est de montrer comment les méthodes algébriques et à base d'agents sont bien adaptées pour découvrir des phénomènes complexes dans les systèmes biologiques et apporter un nouvel éclairage sur l'interprétation des maladies génétiques.

Les sections suivantes fournissent un bref aperçu et une contextualisation des sujets abordés dans le reste de cette thèse, qui est divisée en deux parties principales : la première se concentre sur les modèles algébriques des ARN et des protéines, tandis que la seconde décrit notre étude à base d'agents sur interactions biomoléculaires. Bien que ces deux approches puissent être liées l'une à l'autre (comme le montre le Chapitre 4), nous les séparons pour aider le lecteur à distinguer les travaux menés principalement dans le cadre de l'Université de Camerino (Part I) des résultats de la collaboration avec l'Université d'Aix-Marseille (Part II).

Modélisation algébrique de l'ARN et des protéines

La relation entre structure et fonction est un sujet pertinent en biologie, dont l'étude a reçu une contribution significative de différentes approches informatiques [13, 27, 69, 73, 89]. En particulier, les langages formels et les grammaires de graphes ont été appliqués avec succès pour modéliser les propriétés qui corrèlent les fonctions exprimables par les molécules d'ARN et les sous-structures spécifiques impliquées dans leur repliement [68, 96]. Cette dernière joue un rôle fondamental dans cette analyse, car elle permet à un biopolymère linéaire d'atteindre une conformation tridimensionnelle (en formant des liaisons hydrogène entre des monomères non consécutifs).

Dans ce manuscrit, nous poussons cette idée plus loin et prouvons que la complexité des fonctions de l'ARN peut être attribuée à la potentialité interne de chaque nucléotide à interagir avec d'autres dans la même séquence. Ce résultat est obtenu en comparant le repliement de l'ARN avec celui effectué par les protéines pour identifier un niveau d'abstraction auquel ces deux classes de molécules présentent la même complexité structurale et fonctionnelle. Nous appelons ce niveau le *niveau de congruence* ; sa caractérisation est possible grâce à l'expressivité des algèbres de processus [1], à travers lesquelles nous modélisons le repliement de l'ARN et des protéines.

Au cours de la seconde moitié du siècle dernier, l'enquête sur les raisons de l'existence de telles molécules similaires a conduit à la formulation de l'hypothèse du monde à ARN (« RNA world ») : l'ARN pourrait être un « fossile » d'un monde d'ARN qui existait sur Terre avant l'apparition des cellules modernes, dans lequel l'ARN remplissait en même temps les rôles d'ADN et de protéines [44]. Cette théorie est encore très débattue, car, au-delà de leurs similitudes, les protéines et les ARN présentent de profondes différences structurelles qui affectent la façon dont ils remplissent leurs fonctions [93]. Dans la première partie de cette thèse, nous comparons formellement le processus de repliement des protéines avec celui des ARN. Nous concentrons notre étude sur les interactions réalisées par les unités élémentaires qui composent les ARN et les protéines, décrivant l'ensemble du repliement comme le comportement résultant de ces interactions ; en mettant en évidence leurs propriétés fondamentales, nous visons à identifier des indices sur la validité de l'hypothèse du monde à ARN.

Nous nous concentrons ensuite sur une classe de pathologies affectant le processus de repliement. Cette partie de notre étude commence par une description formelle de la façon dont

ces pathologies proviennent d'une erreur dans le code génétique (une mutation, en termes biologiques) et peuvent se propager à chaque étape de l'expression génique, affectant à la fois les structures de l'ARN et des protéines. Nous modélisons comment la mutation d'un seul nucléotide (mutation ponctuelle) peut altérer la conformation finale d'une protéine alors qu'elle est inoffensive pour la structure des ARN ; nous montrons également qu'une maladie génétique bien connue, la drépanocytose, peut être considérée comme un comportement global des interactions des acides aminés et des nucléotides.

Nous faisons enfin un autre pas en avant en supposant les fonctions biologiques qui caractérisent le *niveau de congruence* mentionné ci-dessus et en explorant plus avant l'applicabilité des algèbres de processus pour décrire ses propriétés. Les modèles résultants formeront éventuellement la base d'une simulation à base d'agents [57]. Les agents sont des éléments logiciels discrets dont les interactions correspondent à celles réalisées par les composants du système modélisé assez fidèlement au comportement réel d'un processus biologique [75]. Dans les algèbres de processus, les processus sont concurrents, autonomes et réactifs ; toutes ces propriétés sont également partagées par les agents, ce qui fait des algèbres de processus des langages de spécification appropriés pour les systèmes à base d'agents.

Modélisation à base d'agents et simulation des interactions biomoléculaires

La seconde partie de ce manuscrit décrit un simulateur à base d'agents développé pour étudier les interactions moléculaires qui caractérisent les voies métaboliques et analyser leurs propriétés globales [22]. Nous avons simulé des réactions enzymatiques complètes en modélisant les molécules impliquées (enzymes, métabolites et complexes) comme des agents autonomes et interactifs.

Des études in vitro montrent qu'une macromolécule biologique se comporte comme un dipôle oscillant et que des forces à longue portée peuvent être activées entre deux systèmes moléculaires résonnants. Cela se produit parce qu'une charge qui oscille à haute fréquence (de l'ordre de $10^{10} - 10^{11}$ Hz) n'est pas affecté par l'écrantage de Debye [45, 81]. Notre objectif est de fournir une validation in silico à ces expériences par le biais de simulations à base d'agents, où chaque molécule est représentée par un agent capable de percevoir l'environnement et les partenaires apparentés avec lesquels il peut interagir. Un résultat similaire peut être obtenu grâce à un modèle de dynamique moléculaire ; cependant, cette méthode place souvent l'analyse au niveau atomique et les simulations associées nécessitent la connaissance a priori d'un grand nombre de paramètres expérimentaux. Au lieu de cela, les modèles et les simulations à base d'agents permettent de mener l'étude à un niveau d'abstraction qui peut être représenté avec une quantité raisonnablement petite de données empiriques sans perte de précision lors de la reproduction d'un comportement macromoléculaire. Nous explorons la capacité du simulateur à traiter les interactions électrodynamiques à longue distance qui façonnent le comportement des systèmes biomoléculaires, nous permettant ainsi d'analyser leur effet sur l'évolution d'une voie métabolique, telle que la glycolyse de la levure.

Cependant, comprendre et représenter dans son ensemble la dynamique des agents caractérisant une réaction faite par un grand nombre de molécules constitue encore un défi considérable. Pour cette raison, nous définissons un nouveau paradigme de visualisation fondé sur le concept de *interaction-comme-perception* : chaque fois qu'une molécule perçoit un partenaire apparenté, un lien potentiel entre les deux est établi. On peut ainsi dériver le graphe des perceptions à chaque pas de temps de simulation ; sur ces graphes, nous appliquons l'analyse des données topologiques pour capturer les interactions à 3-corps en interprétant les 2-simplexes – coques convexes de trois points – comme des structures observables. Nous utilisons la formation des 2-simplexes comme sémantique valide pour représenter la dynamique globale du système.

Organisation du manuscrit

Chacune des deux parties principales de ce manuscrit est corrélée à un chapitre d'introduction (Chapitres 1 et 5, respectivement), qui décrit les concepts biologiques et théoriques de base nécessaires pour mieux comprendre notre étude et la méthodologie adoptée. Les chapitres suivants détaillent les résultats que nous avons obtenus.

La première partie comprend les chapitres 1 à 4 :

- Dans le chapitre 1, nous fournissons quelques connaissances de base sur l'expression des gènes et le repliement de l'ARN et des protéines ; nous introduisons également les méthodes formelles adoptées pour modéliser ces processus biologiques, précisément le calcul des systèmes communicants (CCS, Calculus of Communicating Systems), les systèmes de transition étiquetés (LTS, Labelled Transition Systems) et la logique Hennessy-Milner (HML, Hennessy-Milner Logic).

- Dans le chapitre 2, nous utilisons CCS et LTS pour modéliser le repliement de l'ARN et des protéines, et démontrer comment il est possible de définir formellement un niveau d'abstraction dans lequel de tels processus montrent équivalence comportementale (*niveau de congruence*). Sa définition nous permet d'émettre des hypothèses sur certaines des raisons qui ont conduit l'évolution de la vie à former des protéines et à les utiliser comme principaux catalyseurs dans les processus biologiques.

- Chapitre 3 analyse une classe de pathologies qui affectent les processus de repliement pour étudier comment les dissemblances entre les composants structurels des protéines et des ARN provoquent différentes réponses à une altération de la voie de repliement correcte.

- Au chapitre 4, nous explorons l'expressivité du CCS dans la modélisation des fonctions représentant le comportement des molécules d'ARN non codantes, conçues comme une caractérisation du niveau de congruence défini au chapitre 2. Sur la base de ces résultats, nous proposons une méthodologie appropriée pour générer une spécification algébrique pour les simulations à base d'agents.

La deuxième partie de ce manuscrit comprend les chapitres 5 à 7 :

- Le chapitre 5 présente les étapes fondamentales de la glycolyse et le simulateur à base d'agents, Orion, que nous avons développé pour étudier ce processus en termes d'interactions moléculaires.
- Le chapitre 6 décrit comment nous avons adapté Orion pour simuler les interactions moléculaires à longue distance dans les réactions métaboliques et analyser comment elles affectent l'efficacité de la glycolyse.
- Dans le chapitre 7, nous faisons un pas en avant en utilisant des simulations à base d'agents pour reproduire la dynamique à trois corps dans une réaction biochimique, visualisant et comprenant ainsi son comportement global ; cela est possible en appliquant le paradigme *interaction-comme-perception*.

Part I

Algebraic Models

Chapter 1

Background and Methods for Part I

1.1 Introduction

This chapter is intended to provide the reader with the basic concepts, biological and theoretical, needed to comprehend the models described in Part I of this manuscript.

Section 1.2 gives an overview of the processes that underpin protein folding and gene expression; we also introduce the RNA world hypothesis, discussed in Chapter 2. Finally, we briefly describe haemoglobin, a protein that we analyse in Chapter 3 to model the behaviour of sickle-cell anaemia. The content of this section is mainly based on well-established biological and biochemical knowledge [2, 65, 119].

In Section 1.3, we describe the key formalisms at the basis of our modelling approaches. We cover the fundamentals of Calculus of Communicating Systems, labelled transition systems, and Hennessy-Milner logic; we also introduce the concept of software agent, which is used in Chapter 4, even though we go into detail about agent-based modelling and simulation in the second part of this manuscript.

This chapter contains no original content, except for section 1.3.5, where we show how the models of the dissertation's first and second parts can be linked together in a consistent discourse.

1.2 Fundamentals of Molecular Biology

1.2.1 The structure of DNA and the replication process

A deoxyribonucleic acid (*DNA*) consists of two strands of *nucleotides*, molecules made of a *sugar-phosphate group* covalently linked to a *nucleobase* (or simply, *base*). The sugar of a nucleotide is composed of five carbon atoms, each identified by a number followed by a prime mark (e.g., 5'-carbon). Two nucleotides in the same DNA strand are linked through a covalent bond between the sugar's $3'$-hydroxyl (-OH) group of one of them and the $5'$-phosphate (-PO4) of the other.

Only the base differs in each nucleotide and can be one of four possible types: *adenine* (*A*), *guanine* (*G*), *cytosine* (*C*) or *thymine* (*T*). Adenine and guanine are two-rings bases (*purines*), while cytosine and thymine are single-ring bases (*pyrimidines*).

The two nucleotide strands of a DNA molecule are held together by hydrogen bonds, connecting the bases of one strand to those of the other. Adenine always pairs with thymine, while guanine always pairs with cytosine (that is, a purine always pairs with a pyrimidine); they are often called *Watson-Crick base pairs*. A detail important for the models provided in the subsequent chapters is that, in this coupling process, adenine and thymine–or uracil in *ribonucleic acids* (*RNAs*), as we will see later–pair through two hydrogen bonds, while guanine and cytosine form three hydrogen bonds. As a consequence of this complementary base pairing, each strand of a DNA molecule contains a sequence of nucleotides that is exactly complementary to the sequence of the other strand. DNA strands run antiparallel to each other (i.e., oriented in opposite polarities), twisted into a double helix (Figure 1.1).

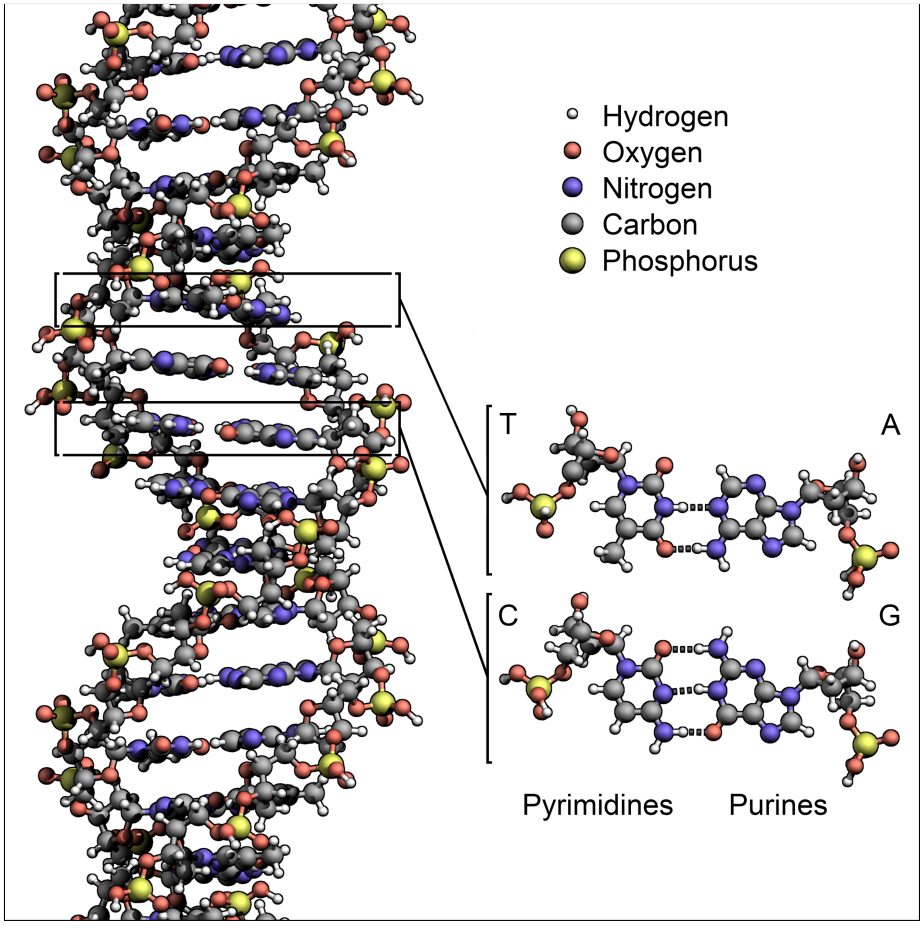


Figure 1.1 – Schematic representation of the DNA double helix and its hydrogen bonds in the base pairing of adenine (A) with thymine (T) and guanine (G) with cytosine (C). Adapted from ©Richard Wheeler (User:Zephyris) / Wikimedia Commons / CC BY-SA 3.0.

The information of a DNA strand is encoded in the sequence of its nucleotides; differences in nucleotide order determine different biological messages expressed by DNA.

The possibility of nucleotide base pairing also allows the DNA strands to be used as templates for generating completely new DNA molecules in a process called *DNA replication*. Like many others in cells, this process is performed by an enzyme, a molecule acting as a catalyst by helping complex reactions occur. The replication process is carried out by the *DNA polymerase* enzyme and starts from a defined sequence of nucleotides, the *replication origin* (Figure 1.2).

While the replication process proceeds, the DNA polymerase monitors and corrects possible errors in the base paring from the original to the new strand (*proofreading*). However, some errors can be left uncorrected, causing a so-called *mismatch*: a mispaired nucleotide. For this reason, a specific complex of molecules has the function of *mismatch repairing*. If a replication mistake escapes this additional control, the new DNA strand will present a *mutation*, a permanent change of its sequence that can alter a fundamental process called *gene expression*.

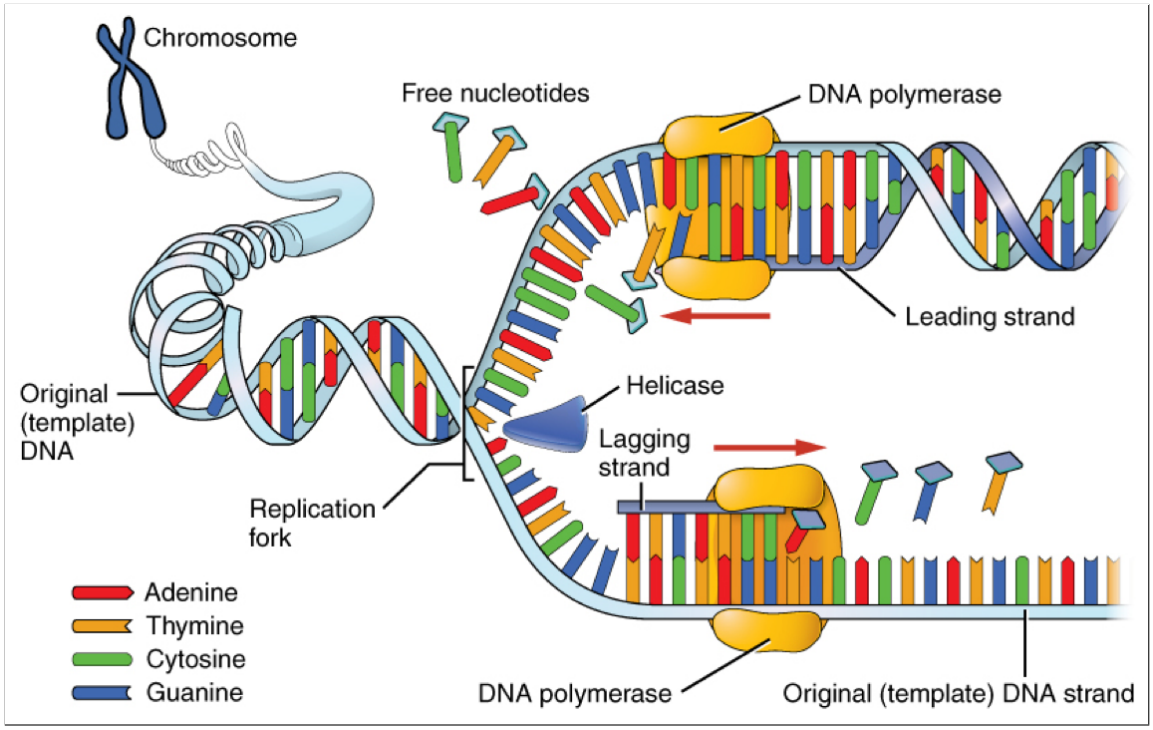


Figure 1.2 – Replication of a DNA nucleotide sequence carried out by the *DNA polymerase*. The strand taken as a template to generate the new one is accessed by unwinding the DNA double helix (a process performed by the *helicase* enzyme). DNA replication produces two new double helices, each made of one of the pre-existing strands twisted with the new strand through complementary base pairing. Image accessed free and adapted from ©OpenStax Anatomy and Physiology [14] / CC BY 4.0.

1.2.2 Gene expression

Genes are specific sequences of nucleotides that contain the instructions for producing functional molecules–*proteins* or *RNAs*–and collectively form the organism's genome. The process that converts the information encoded in the nucleotide sequence of a gene into the related functional product is defined as *gene expression*.

In this context, RNA molecules can be the intermediate or final product of the process. RNA is a linear molecule similar to DNA; however, it shows some differences. For a better understanding of the following chapters, it is important to consider that:

- RNA is composed of adenine, guanine, and cytosine bases, like DNA, but it contains uracil (U) instead of thymine (T). However, a uracil base behaves similarly to thymine and can base-pair with adenine.

- An RNA molecule is single-stranded, meaning that it can fold on itself and form three-dimensional structures. As we elaborate on in the remainder of this section, such a property allows some type of RNA molecules to carry out complex functions in cells (Figure 1.3).

RNAs are made through *transcription*, a process performed by an enzyme called *RNA polymerase* [117]; it uses one of the two strands of DNA as a template to build the RNA molecule through base pairing; the resulting product is called the *transcript*. The transcription process starts from a sequence of nucleotides defined as the *promoter* and continues until the RNA polymerase reaches another group of nucleotides, the *terminator* or *stop site*.

Most of the genes in a DNA molecule represent the instructions for synthesising *proteins*, a class of molecules that carry out fundamental activities in living cells. A protein is a sequence of amino acids; these organic compounds thus constitute the protein's primary elements (or monomers), determining its three-dimensional conformation and, consequently, its functions.

The role of RNA is often placed in the middle of the gene expression process since the genetic information is *transcribed* into the nucleotide sequence of an RNA molecule that, in turn, is *translated* into the amino acid sequence of a protein; in this case, the RNA molecule is defined as *messenger RNA* or *mRNA*). Some genes, however, encode the information to generate ultimately a molecule of RNA, which performs itself the required functions in the cell. This class of RNAs is sometimes called *functional* or *non-coding RNAs (ncRNAs)* and includes important molecules like *ribosomal RNAs*, *transfer RNAs* and *microRNAs*.

In eukaryotic cells, before being translated, mRNA must undergo three *RNA processing* steps: *capping*, *polyadenylation*, and *splicing* [92]. In particular, the last step is performed by the *spliceosome*, complex machinery partly composed of RNA (small nuclear RNA), which removes from the RNA nucleotide chain the non-coding sequences (or *introns*) while assembling the coding ones (*exons*)–see Figure 1.4.

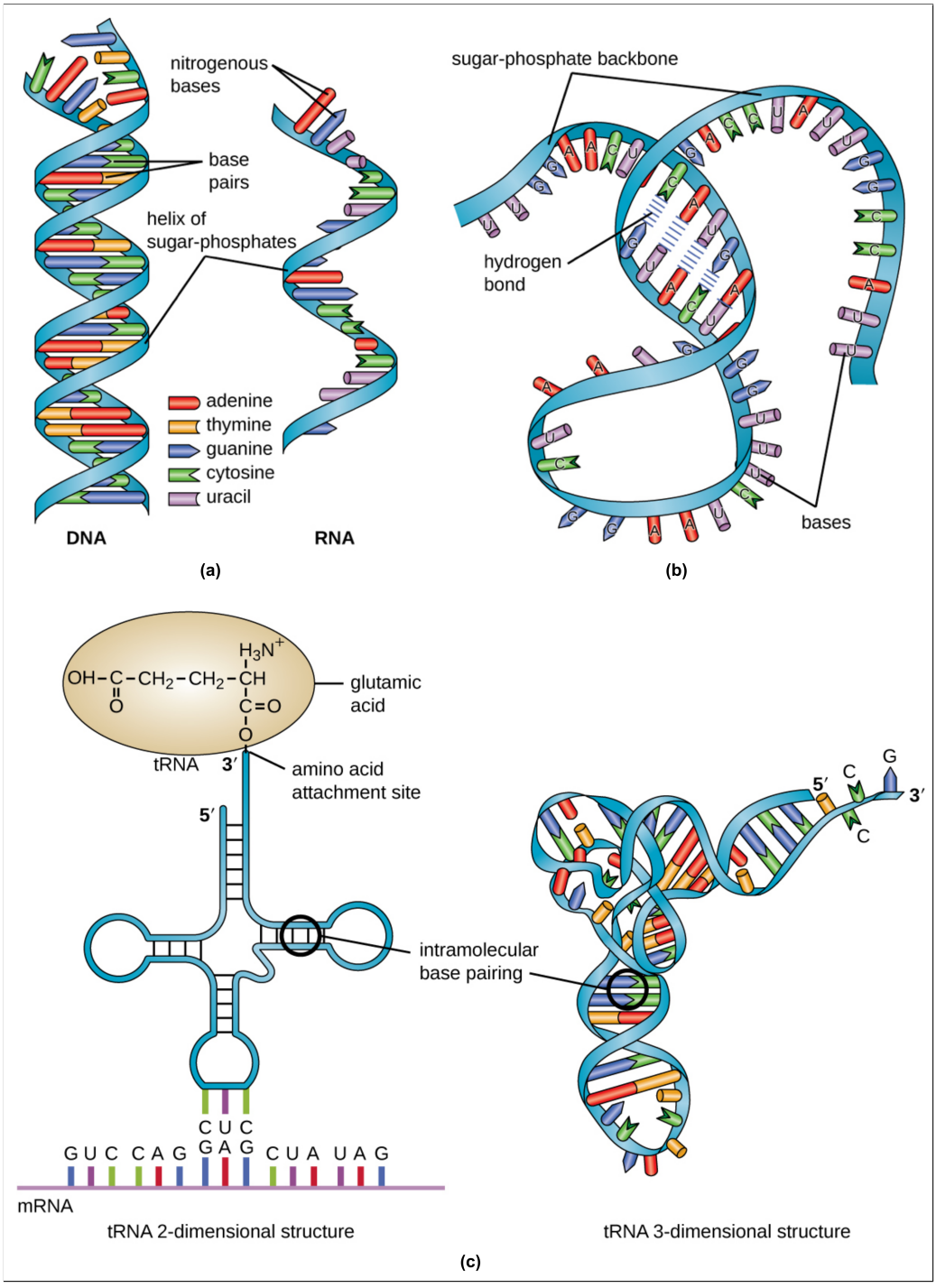


Figure 1.3 – **(a)** Differences between the double helix of DNA and the structure of RNA, which is single-stranded and contains the base uracil (U) instead of thymine (T). **(b)** Driven by a base pairing process similar to that of DNA, the linear sequence of RNA nucleotides (*primary structure*) folds into three-dimensional conformations, which determine and affect its function inside the cell. **(c)** Example of planar (*secondary*) and three-dimensional (*tertiary*) structures of RNA; specifically, we show the folding of tRNA, whose peculiar spatial arrangement makes it able to bind an amino acid and contribute to the synthesis of a protein during the translation process of gene expression. Images accessed free and adapted from ©OpenStax Microbiology [87] / CC BY 4.0.

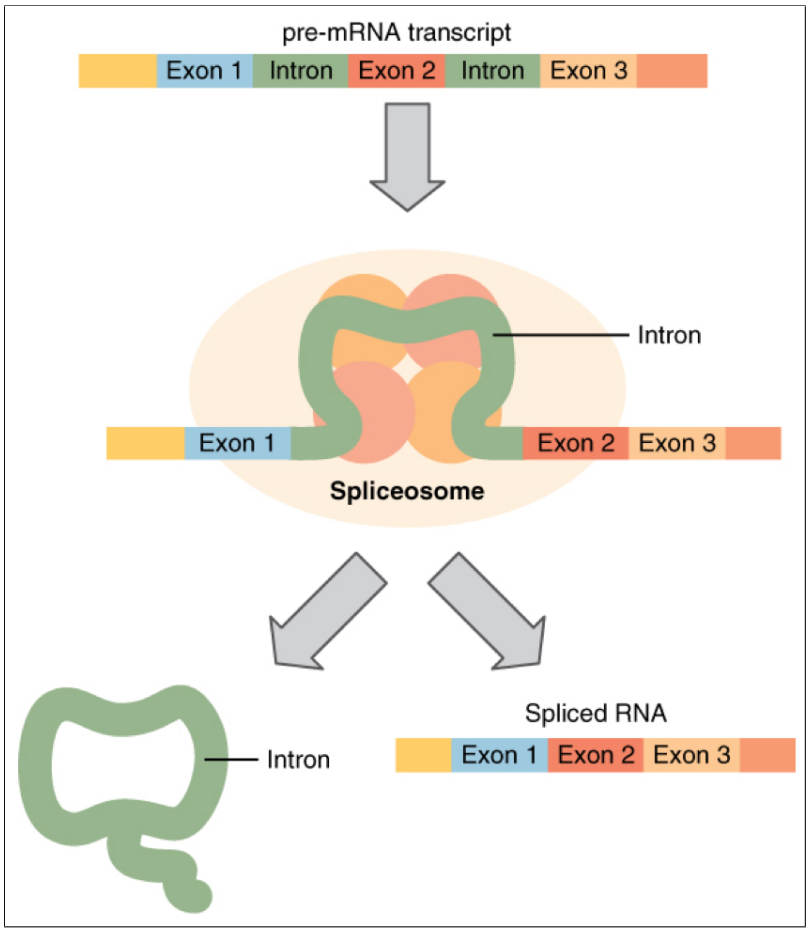


Figure 1.4 – Removal of the non-coding sequences (*introns*) from the *transcript*, performed by the *spliceosome* to generate a coding *mRNA* molecule (made of *exons*), which is ready to undergo the translation process of gene expression. Image accessed free and adapted from ©OpenStax Anatomy and Physiology [14] / CC BY 4.0.

After the transcription of a DNA sequence into an mRNA molecule, the latter undergoes the *translation* process [61], which synthesises a new protein (Figure 1.5).

As we said, a protein is made of amino acids. In the aminoacidic sequences of proteins, we can identify 20 different types of these organic compounds (see Table 1.1). However, in both DNA and their RNA transcripts, genes are composed of 4 different types of nucleotides–A, T (or U), C, and G. During the translation process, they are read three by three (in groups called *codons*); therefore, the *genetic code* associates 64 combinations of three nucleotides to 20 possible amino acids, with the obvious redundancy: the same amino acid is coded for by more than one triplet of nucleotides, as shown in Table 1.2.

Table 1.1 – List of the 20 amino acids identifiable in proteins and the abbreviations used in this manuscript.

Name	Abbreviation
Alanine	Ala
Arginine	Arg
Asparagine	Asn
Aspartic acid	Asp
Cysteine	Cys
Gycine	Gly
Glutamic acid	Glu
Glutamine	Gln
Histidine	His
Isoleucine	Ile
Leucine	Leu
Lysine	Lys
Methionine	Met
Phenylalanine	Phe
Proline	Pro
Serine	Ser
Threonine	Thr
Tryptophan	Trp
Tyrosine	Tyr
Valine	Val

Table 1.2 – Genetic code table. It represents the association of each of the 20 amino acids plus the stop signal of the translation process (all listed in the last row) with the related group of matching codons.

	aga								uua					agc					
	agg								uug					agu					
gca	cga						gga		cua				cca	uca	aca			gua	
gcc	cgc						ggc	aua	cuc				ccc	ucc	acc			guc	uaa
gcg	cgg	gac	aac	ugc	gaa	caa	ggg	cac	auc	cug	aaa	uuc	ccg	ucg	acg		uac	gug	uag
gcu	cgu	gau	aau	ugu	gag	cag	ggu	cau	auu	cuu	aag	aug	uuu	ccu	ucu	acu	ugg	uau	guu	uga
Ala	Arg	Asp	Asn	Cys	Glu	Gln	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val	STOP

The messenger RNA translation is mediated by an ncRNA molecule, the *transfer RNA* (or *tRNA*), which can bind an amino acid; in addition, it contains a triplet of nucleotides complementary to one of the codons in the mRNA chain that matches with its carried amino acid. This process does not happen spontaneously, but it is performed by large molecular complexes called *ribosomes*. They move along the mRNA and join, by forming a covalent bond (*peptide bond*), the amino acid held by a tRNA to the last one in the growing amino acid chain of the protein. Ribosomes start the translation process at a specific nucleotide triplet, AUG, and end when they reach one of the three possible *stop codons*: UAA, UAG, or UGA.

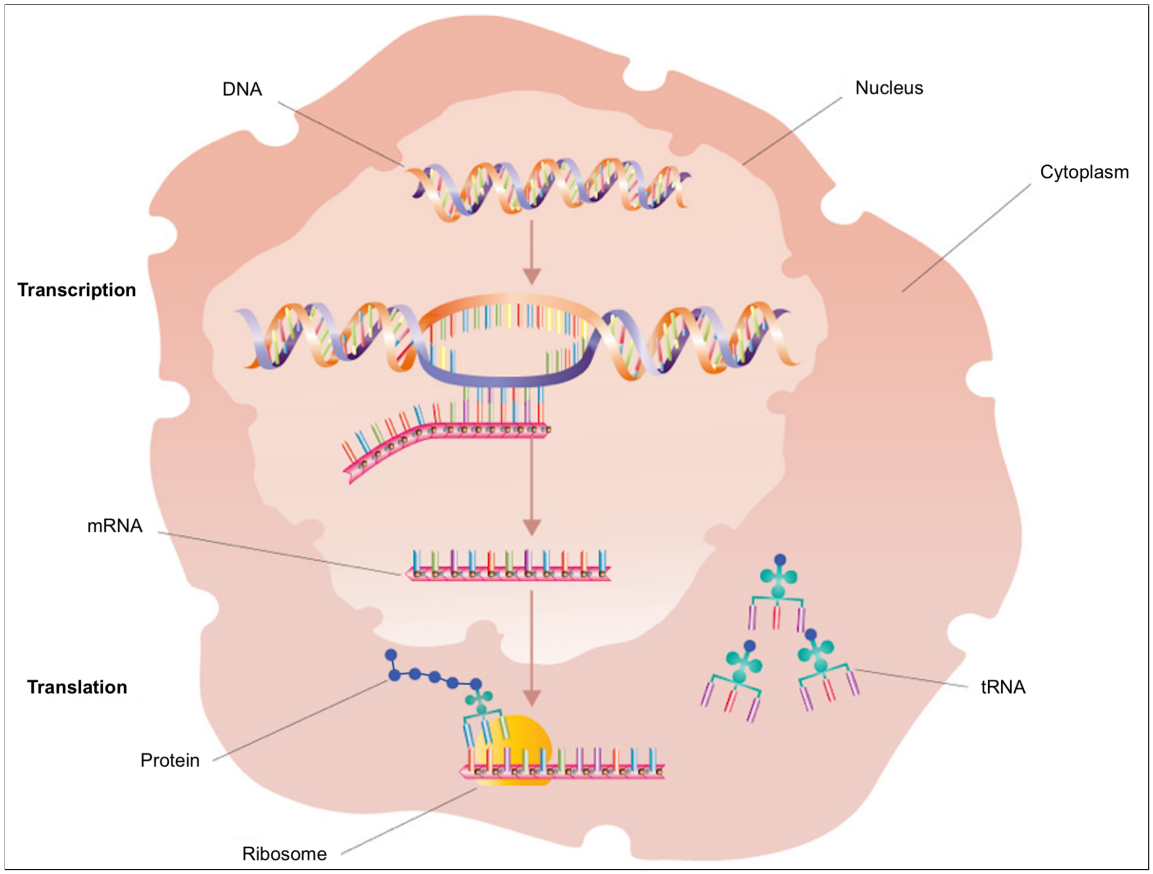


Figure 1.5 – Expression of the DNA sequence of a gene into the amino acid chain of a protein. In the *transcription* process, the RNA polymerase (not shown) reads the DNA string and generates a complementary mRNA molecule, while the subsequent *translation*, performed by ribosomes with the aid of tRNAs, takes the mRNA nucleotide sequence as a template to produce the protein. Adapted from ©NHS National Genetics and Genomics Education Centre / Wikimedia Commons / CC BY 2.0.

In Chapter 3, we describe gene expression thoroughly to analyse the differences in RNA and protein structural complexity; this study is carried out through the definition of a comprehensive gene expression model based on process algebras.

1.2.3 Protein structure and folding

An amino acid is an organic compound made of a central carbon atom bound to a *carboxyl group* (–COOH), an *amino group* (–NH_2), and a *side carbon chain* (also called *R group*). The *R group* characterises each amino acid and distinguishes it from the others. For the aim of the model proposed in the following chapters, it is important to know that amino acid side chains can be water-soluble (*hydrophilic*) or water-insoluble (*hydrophobic*). Amino acids bind to each other through peptide bonds, thus constituting the backbone of a protein (see Figure 1.6).

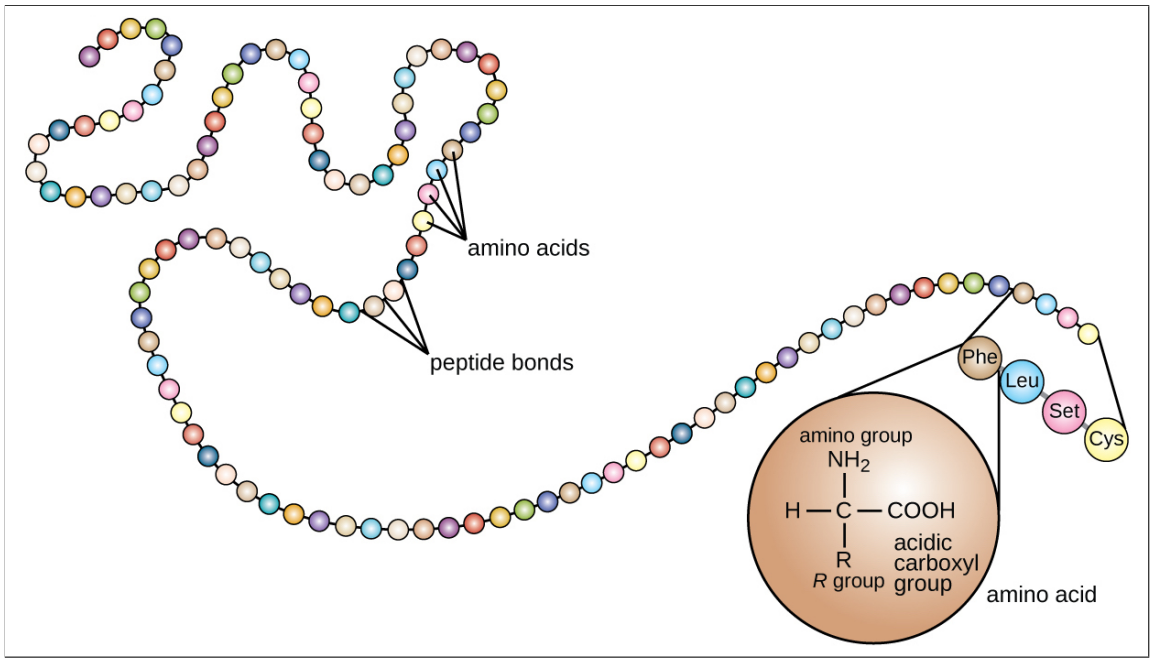


Figure 1.6 – Amino acids are linked through covalent bonds (*peptide bonds*) in the linear sequence that represents the *primary structure* of a protein. Each peptide bond forms between the *carboxyl group* (–COOH) of an amino acid and the *amino group* (–NH_2) of another. The *R group* determines the specific behaviour of an amino acid in relation to its environment (e.g., hydrophobic/hydrophilic properties). Adapted from ©CNX OpenStax / Wikimedia Commons / CC BY 4.0.

The linear sequence of amino acids that compose a protein (*primary structure*) goes through a *folding process*; it creates recurring structural patterns (*secondary structures*), such as helices or sheets of amino acids, until it forms a complex three-dimensional molecule (*tertiary structure*).

In many cases, the final conformation of a protein results from the aggregation of more than one folded polypeptide chain (*quaternary structure*).

During the folding process, the hydrophobic amino acid side chains are pushed away from water, grouping in the protein's interior. In this way, some side chains are *buried*, while others are *exposed*, generating an "inside" and an "outside" of the protein. The structure is also stabilised by hydrogen bonds between the carboxyl group of one amino acid and the amino group of another.

The resulting three-dimensional conformation has the following main properties:

- it exists under the most thermodynamically stable conditions (lowest Gibbs free energy);
- it is stabilised primarily by disulphide bonds and non-covalent interactions;
- it is associated with the functions expressible by a protein.

In other words, the linear sequence of a protein influences how it folds up into a three-dimensional structure–stabilised by non-covalent interactions–that, in turn, determines the functions of the protein (see Figure 1.7).

1.2.4 RNA folding and non-coding functions

RNA molecules tend to fold into a three-dimensional form, similarly to proteins. In this case, the hydrogen bonding between complementary bases leads the process. Moreover, base-stacking interactions push the molecule to assume a helical conformation (see Figure 1.3).

In addition to conventional Watson-Crick base pairs (see Section 1.2.1), RNA helices often contain *non-canonical* (*non-Watson-Crick*) *base pairs*. The most common are GU and GA pairs, but there are more than 20 different types identified in RNAs, including *base triples* [80].

RNAs can adopt complex three-dimensional structures to carry out non-coding functions, such as biological catalysis. In this case, they are also known as *ribozymes* and, like protein enzymes, have binding sites for a substrate and a co-factor needed for the catalytic process.

Investigating the similarities between RNAs and proteins is the "leitmotif" of Part I of this manuscript. In particular, in Chapter 2, we examine the existence of a congruence level in which these two types of molecules are able to perform functions of the same complexity. The provided results are interpreted as supporting the validity of the RNA world hypothesis. In Chapter 4, we push forward such findings by characterising, through formal models, the functions carried out by ribozymes.

1.2.5 RNA world

The majority of the molecules involved in the various stages of gene expression are proteins (e.g., RNA polymerases) or are composed in part of proteins (e.g., ribosomes). Therefore, nucleic acids are required to direct the synthesis of proteins, and proteins are required to synthesise nucleic acids; we might wonder how this system of interdependent components could have

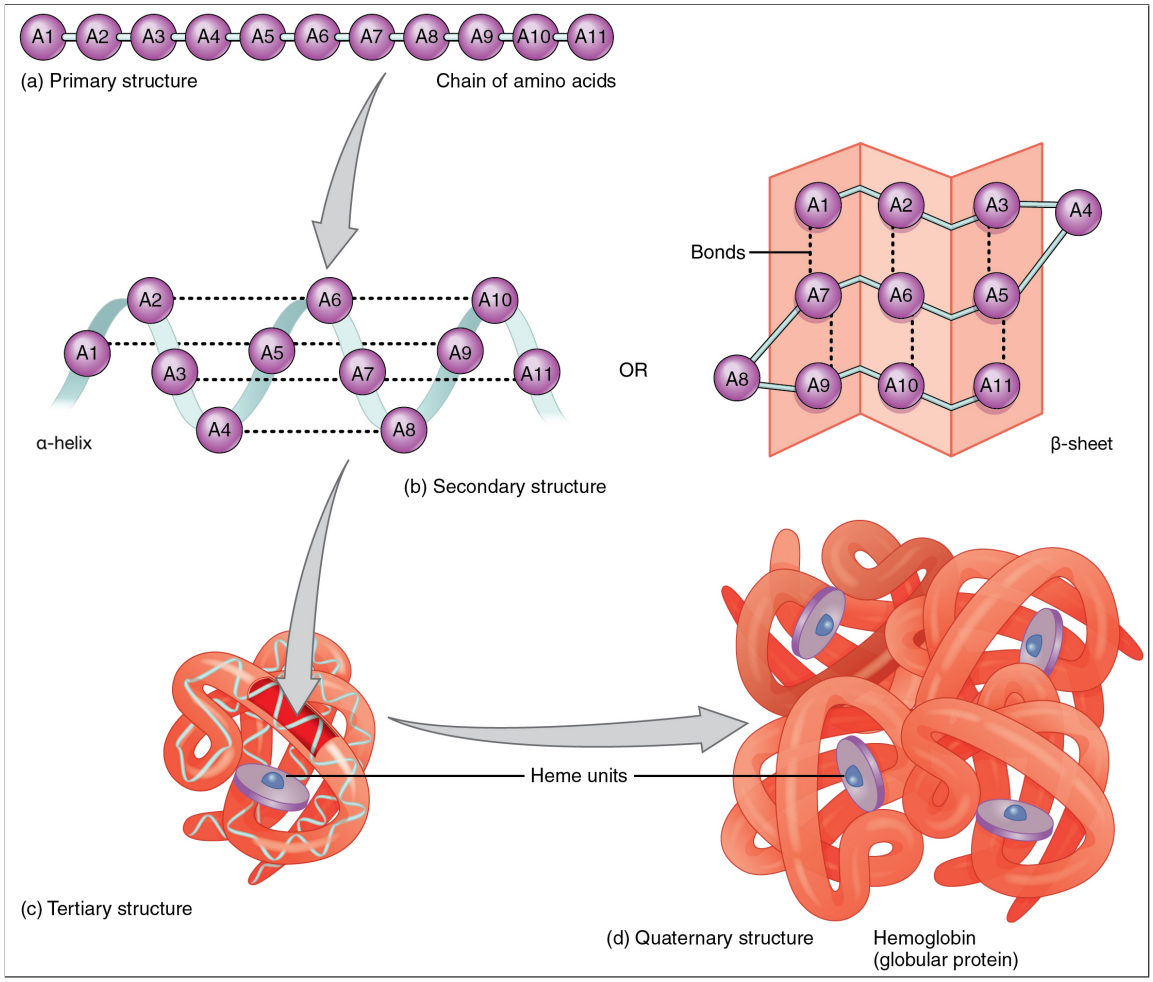


Figure 1.7 – Four levels of complexity of the structures determining the final conformation of a protein. In particular, we show the folding process of haemoglobin, a molecule discussed in Section 1.2.6 and Chapter 3. The reduction of the Gibbs free energy and the formation of non-covalent interactions (e.g., hydrogen bonds and hydrophobic interactions) push the unfolded sequence of amino acids (*primary structure*) to form α-helices and β-sheets (*secondary structures*). The same forces drive these amino acid patterns to arrange in three-dimensional polypeptide chains (*tertiary structures*), whose composition determines the *quaternary structure* that characterises some classes of molecules. Adapted from ©OpenStax College / Wikimedia Commons / CC BY 3.0.

arisen. One hypothesis is that an *RNA world* existed on Earth before modern cells appeared, and RNA, which today mostly serves as an intermediate between genes and proteins, performed both the functions of storage for the genetic information and of biological catalyst [44, 93]. Only when modern cells appeared these two functions were separated, and DNA became the carrier of the genetic information, while proteins took the role of the main catalyst in cellular processes.

This hypothesis may be supported by the existence of RNA molecules that catalyse important reactions in cells (the ribozymes, as we discussed before); moreover, the sequence of genes of DNA is copied in an mRNA molecule during the transcription process, meaning that RNA is still able to store the genetic information. Viruses are examples of organisms in which the entire genome may be exclusively in the form of RNA (RNA viruses).

However, the necessity of reducing errors in the replication process and carrying out more complex functions in cells led, at a certain point in the evolution of life, to the formation of the two specialised structures we can observe today. In Chapter 2, we support this idea with the aid of algebraic models.

1.2.6 Haemoglobin and anaemias

An important type of protein is haemoglobin, which is found in erythrocytes (red blood cells); their function is to carry oxygen from the lungs to the body's tissues and return carbon dioxide from the tissues back to the lungs.

It is composed of four protein subunits (called *globins*), each consisting of two α-chains and two β-chains, allowing haemoglobin to bind four oxygen molecules [70]. In the binding process, a fundamental role is plaid by an iron-containing compound called *heme*, which is embedded in each globin (see Figures 1.7 and 1.8).

We are interested specifically in this molecule because, in Chapter 3, we use it as a case study to observe, through the definition of formal models, the effect of a mutation on the folding process of proteins compared to that of RNAs.

Indeed, a single nucleotide change (*point mutation*) in the β-globin gene may produce valine (Val) instead of glutamic acid (Glu) in the amino acid sequence of the β-subunit (*Glu6Val* mutation, which results in the *HbS disease*). If a single point mutation of this type may not be harmful, inheriting two copies of the mutant β-globin gene will cause *sickle-cell anaemia*. The necessity of the hydrophobic amino acid to be shielded from water pushes valine to bind into the hydrophobic pocket of another haemoglobin molecule, forming, in this way, the fibrous precipitates which characterise sickle-cell disease. This process causes the red blood cells to assume a typical sickle shape, which cannot move easily into vessels and may obstruct the normal blood flow; such a behaviour has the consequence of reducing the flux of oxygen through the body [19].

In Chapter 3, we deeply analyse sickle-cell anaemia as a global behaviour resulting from the interactions occurring during the gene expression.

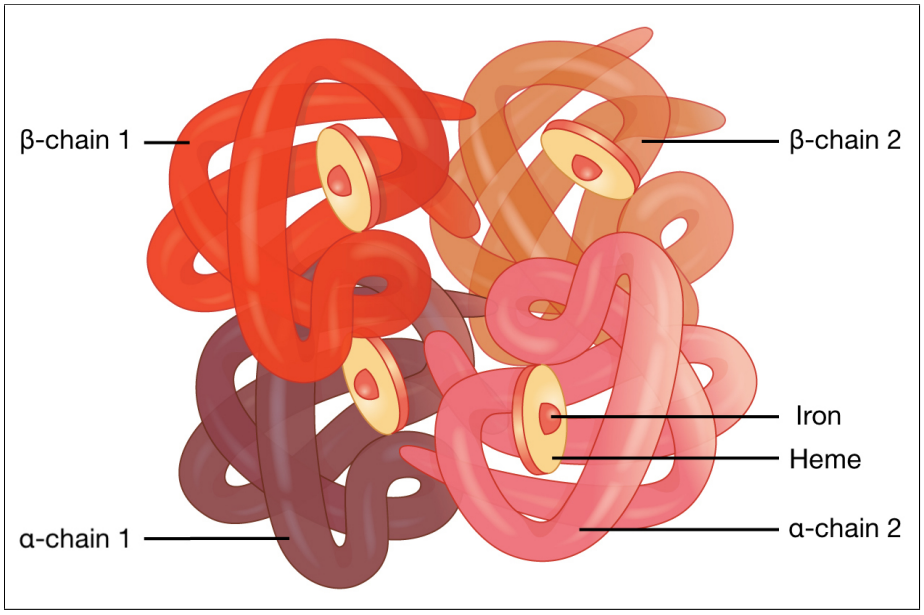


Figure 1.8 – Structure of haemoglobin with its two α and two β subunits highlighted; the corresponding four iron-containing heme groups are also shown. Adapted from ©OpenStax College / Wikimedia Commons / CC BY 3.0.

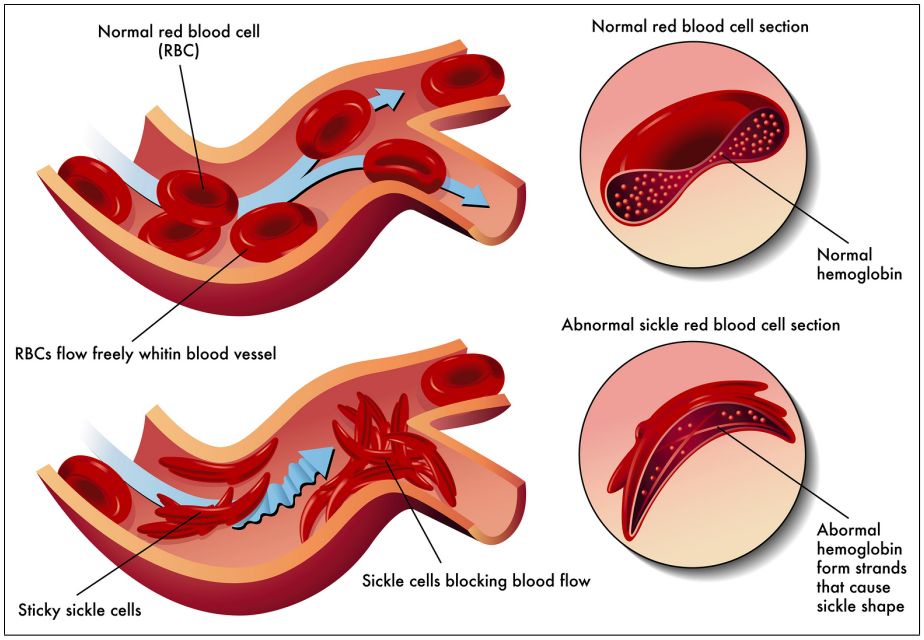


Figure 1.9 – Comparison of normal and sickle-shaped red blood cells and their different behaviours in blood vessels. Adapted from ©User:Diana grib / Wikimedia Commons / CC BY-SA 4.0.

1.3 Introduction to the Algebraic Modelling of Biological Systems

This section presents the formal methods through which we define the models proposed in this manuscript. The description is mainly based on Aceto et al. (2007) [1].

1.3.1 Calculus of Communicating Systems

Most of the models we provide in the following chapters are based on Milner's *Calculus of Communicating Systems* (CCS) [77]. It is a process algebra consisting of a collection of constructors for building a new process description from existing ones, representing them as systems that exhibit behaviour and interact via synchronised communication. A process can be viewed as a black box with a name and a set of communication channels; an input or output action on the channel $\mathtt{w}$ is indicated using the labels $\mathtt{w}$ or $\overline{\mathtt{w}}$, respectively. In this manuscript, if a label $\mathtt{w}$ is defined, the existence of $\overline{\mathtt{w}}$ is implied.

In our models, we use the following process constructors. Let P, Q be processes:

- **action prefixing:** if $\mathtt{w}$ is an action, $\mathtt{w}.\mathsf{P}$ is a process that begins by performing the action $\mathtt{w}$ and behaves like P thereafter;
- **choice operator:** $\mathsf{P} + \mathsf{Q}$ is a process that may behave like P or Q;
- **parallel composition:** $\mathsf{P}|\mathsf{Q}$ describes a system in which P and Q run in parallel, proceeding independently or communicating via complementary channels;
- **restriction:** if L is a set of labels, then $\mathsf{P} \setminus L$ is a process in which the scope of the labels in L is restricted to P; this means that those labels can only be used to indicate channels for communications within P.

CCS syntax

Given

A	the set of channel names,
$\overline{A} = \{\overline{\mathtt{w}} \mid \mathtt{w} \in A\}$	the set of complementary names,
$L = A \cup \overline{A}$	the set of labels,
$\mathbf{Act} = L \cup \{\tau\}$	the set of actions, where τ is an unobservable action,
K	the set of process names (constants),

the set E of the CCS expression is given by the following grammar:

$$\mathsf{P},\mathsf{Q} \quad ::= \quad \mathtt{K} \quad | \quad \alpha.\mathsf{P} \quad | \quad \sum_{i\in I} \mathsf{P}_i \quad | \quad \mathsf{P}|\mathsf{Q} \quad | \quad \mathsf{P}[f] \quad | \quad \mathsf{P}\backslash L' \tag{1.1}$$

where

- $\mathtt{K}$ is a process name in K;
- α is an action in **Act**;
- I is a possibly infinite index set;
- f : **Act** $\rightarrow$ **Act** is a relabelling function satisfying the following constraints:
  - $f(\tau) = \tau$
  - $f(\overline{\mathtt{w}}) = \overline{f(\mathtt{w})}$ for each label $\mathtt{w}$;
- L' is a set of labels from L.

The behaviour of each process constant $\mathtt{K} \in K$ is given by a defining equation $\mathtt{K} \stackrel{\text{def}}{=} \mathsf{P}$, where $\mathsf{P} \in E$.

CCS structural operational semantics (SOS)

$\alpha \in$ **Act** and $\mathtt{w} \in L'$,

$$\frac{}{\alpha.\mathsf{P} \xrightarrow{\alpha} \mathsf{P}} \quad \text{Action prefixing} \tag{1.2}$$

$$\frac{\mathsf{P}_j \xrightarrow{\alpha} \mathsf{P}'_j}{\sum_{i\in I} \mathsf{P}_i \xrightarrow{\alpha} \mathsf{P}'_j} \text{ where } j \in I \quad \text{Summation} \tag{1.3}$$

$$\frac{\mathsf{P} \xrightarrow{\alpha} \mathsf{P}'}{\mathsf{P}|\mathsf{Q} \xrightarrow{\alpha} \mathsf{P}'|\mathsf{Q}} \quad \text{Parallel composition (rule 1)} \tag{1.4}$$

$$\frac{\mathsf{Q} \xrightarrow{\alpha} \mathsf{Q}'}{\mathsf{P}|\mathsf{Q} \xrightarrow{\alpha} \mathsf{P}|\mathsf{Q}'} \quad \text{Parallel composition (rule 2)} \tag{1.5}$$

$$\frac{\mathsf{P} \xrightarrow{\mathtt{w}} \mathsf{P}' \quad \mathsf{Q} \xrightarrow{\overline{\mathtt{w}}} \mathsf{Q}'}{\mathsf{P}|\mathsf{Q} \xrightarrow{\tau} \mathsf{P}'|\mathsf{Q}'} \quad \text{Parallel composition (rule 3)} \tag{1.6}$$

$$\frac{\mathsf{P} \xrightarrow{\alpha} \mathsf{P}'}{\mathsf{P} \setminus L' \xrightarrow{\alpha} \mathsf{P}' \setminus L'} \text{ where } \alpha \notin L' \quad \text{Restriction} \tag{1.7}$$

$$\frac{\mathsf{P} \xrightarrow{\alpha} \mathsf{P}'}{\mathsf{P}[f] \xrightarrow{f(\alpha)} \mathsf{P}'[f]} \quad \text{Relabelling} \tag{1.8}$$

$$\frac{\mathsf{P} \xrightarrow{\alpha} \mathsf{P}'}{\mathtt{K} \xrightarrow{\alpha} \mathsf{P}'} \text{ where } \mathtt{K} \stackrel{\text{def}}{=} \mathsf{P} \quad \text{Constant definition} \tag{1.9}$$

A rule of the SOS states that, to establish that the transitions placed below the solid line of the equation can be carried out, we must first prove the possibility of performing the transitions placed above the solid line; they represent the premises of the rule. If there is no premise (as in Equation 1.2), we consider this rule to be an axiom.

1.3.2 Labelled transition systems

The biological processes described in this manuscript have been modelled as the result of subprocesses that proceed along a path made of discrete states. This aspect is often highlighted through *labelled transition systems* (LTSs) [62]; they consist of a set of processes, a set of actions and a transition relation $\rightarrow$ such that, if a process P can perform an action α and become a process P', we write $\mathsf{P} \xrightarrow{\alpha} \mathsf{P}'$ [1].

Formally, a labelled transition system (LTS) is a triple $(\mathbf{Proc}, \mathbf{Act}, \{\xrightarrow{\alpha} \mid \alpha \in \mathbf{Act}\})$, where

- **Proc** is a set of states (or processes);
- **Act** is a set of actions (or labels);
- $\xrightarrow{\alpha} \subseteq \mathbf{Proc} \times \mathbf{Proc}$ is a transition relation, for every $\alpha \in \mathbf{Act}$.

If P becomes P' after a sequence ω of actions, we write $\mathsf{P} \overset{\omega}{\Rightarrow} \mathsf{P}'$.

The LTSs in this manuscript have been generated through the automated tool CAAL - Concurrency Workbench, Alborg Edition [3]. In these cases, an output or input action on the communication channel $\mathtt{w}$ is represented with the labels $'\mathtt{w}$ or $\mathtt{w}$, respectively.

1.3.3 Strong bisimilarity

A binary relation $\mathcal{R}$ over the set of states of an LTS is a *bisimulation* iff whenever $s_1 \mathcal{R} s_2$ and $\mathtt{w} \in L$:

- if $s_1 \xrightarrow{\mathtt{w}} s_1'$, then there is a transition $s_2 \xrightarrow{\mathtt{w}} s_2'$ such that $s_1' \mathcal{R} s_2'$;
- if $s_2 \xrightarrow{\mathtt{w}} s_2'$, then there is a transition $s_1 \xrightarrow{\mathtt{w}} s_1'$ such that $s_1' \mathcal{R} s_2'$.

Two states s and s' are bisimilar, written $s \sim s'$, iff there is a bisimulation that relates them. The relation $\sim$ will be referred to as *strong bisimulation equivalence* or *strong bisimilarity*.

1.3.4 Hennessy-Milner logic

In Chapters 2 and 4, we model some biochemical properties of RNA and proteins through Hennessy-Milner formulae [54, 63]. In Chapter 3, we use a similar approach to represent nucleotide and amino acid sequences.

Hennessy-Milner logic is a multimodal logic, i.e., it involves modal operators parametrised by actions. The set M of Hennessy-Milner formulae over the set of actions **Act** is given by the following abstract syntax:

$$\mathcal{F}, \mathcal{G} ::= \mathbf{tt} \mid \mathbf{ff} \mid \mathcal{F} \wedge \mathcal{G} \mid \mathcal{F} \vee \mathcal{G} \mid \langle\alpha\rangle\mathcal{F} \mid [\alpha]\mathcal{F} \tag{1.10}$$

where $\alpha \in$ **Act**, **tt** and **ff** are used to denote respectively "true" and "false" [1]. The meaning of a formula in M is given by characterising the collection of processes that satisfy it. Intuitively, this can be described as follows:

- all processes satisfy **tt** ;
- no process satisfies **ff** ;
- a process satisfies $\mathcal{F} \wedge \mathcal{G}$ (respectively, $\mathcal{F} \vee \mathcal{G}$) iff it satisfies both $\mathcal{F}$ and $\mathcal{G}$ (respectively, either $\mathcal{F}$ or $\mathcal{G}$);
- a process satisfies $\langle\alpha\rangle\mathcal{F}$ for some $\alpha \in$ **Act** iff it affords a α-labelled transition leading to a state satisfying $\mathcal{F}$;
- a process satisfies $[\alpha]\mathcal{F}$ for some $\alpha \in$ **Act** iff all of its α-labelled transitions lead to a state satisfying $\mathcal{F}$.

As for the LTSs, in the HML formulae analysed with CAAL, an output or input action on the channel α is indicated using the labels $'\alpha$ or α, respectively.

Given $[\![\mathcal{F}]\!]$ the set of all the processes that satisfy $\mathcal{F}$, a process $\mathsf{P} \models \mathcal{F}$ iff $\mathsf{P} \in [\![\mathcal{F}]\!]$. If the formulae $\mathcal{F}$ and $\mathcal{G}$ are satisfied by exactly the same processes, that is, if $[\![\mathcal{F}]\!] = [\![\mathcal{G}]\!]$, we write $\mathcal{F} \equiv \mathcal{G}$.

In this manuscript,

$$\mathcal{F} \equiv \bigwedge_{n} \mathcal{G} \text{ iff } \mathcal{F} \equiv \underbrace{\mathcal{G} \wedge \cdots \wedge \mathcal{G}}_{\text{n}} \tag{1.11}$$

Regarding the set definitions we provide in Chapter 3, we adopt the following conventions:

$$\mathbb{N}^{+} = \mathbb{N}_0 \backslash \{0\}; \tag{1.12}$$

$$\begin{aligned} &\text{let } A \subseteq \textbf{Act}, \ \epsilon \text{ be the empty string, } A^0 = \{\epsilon\}, \ A^1 = A, \text{ and} \\ &A^{i+1} = \{\alpha\beta \mid \alpha \in A^i, \ \beta \in A, \ \forall \ i > 0\}, \text{ then:} \\ &A^* = \bigcup_{i \geq 0} A^i \text{ and } A^+ = A^* A. \end{aligned} \tag{1.13}$$

1.3.5 From algebraic to agent-based models

Agents are systems able to perceive changes in the environment and react to them. Formally, a *reactive agent* is defined by the 6-tuple $\langle E, Per, Ac, see, action, do\rangle$, where

- E is the set of all states for the environment;
- Per is a partition of E (representing the perception of the environment from the agent's point of view);
- Ac is a set of actions;
- see: $E \to Per$;
- $action$: $Per \to Ac$;
- do: $Ac \times E \to E$.

An agent observes the environment (see), selects the appropriate action ($action$), and acts (do) on the environment itself [42].

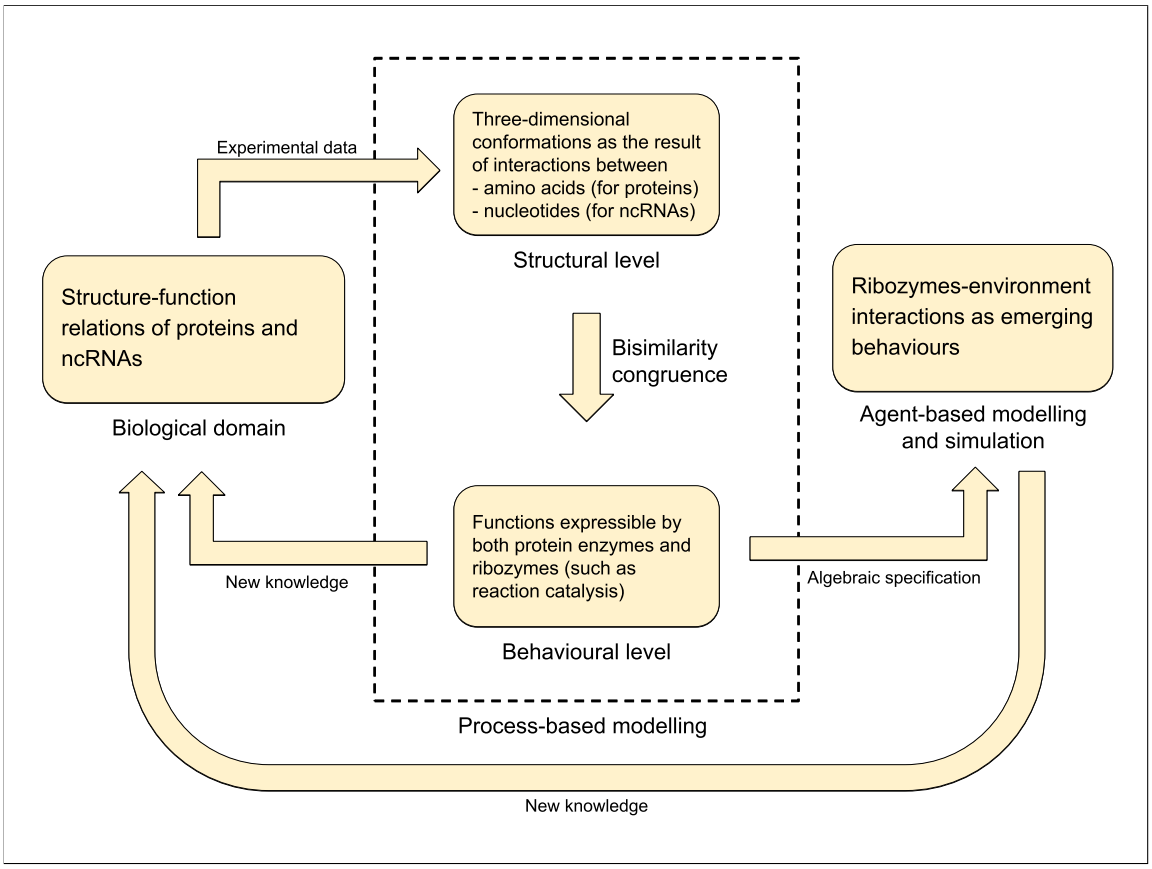


Figure 1.10 – Schematic representation of the modelling approach proposed in our work. *Experimental data* retrieved from *in vivo* and *in vitro* studies on proteins and RNAs provide the basic information and knowledge upon which we constructed the CCS models of their respective folding processes. At the *structural level*, these models correlate the interactions between the elementary units of proteins and RNAs (amino acids and nucleotides, respectively) to their three-dimensional conformations. Discovering an abstraction level in which the two kinds of folding processes are *bisimilar* gives us the perspective needed to identify a class of functions of the same complexity, which proteins and RNAs can equally perform; as clarified in Chapter 2, it also yields *new knowledge* on the *biological domain*. In Chapter 4, we outline an *algebraic specification* of this class of functions, which is intended to be the basis of an *agent-based model*, ultimately resulting in the related computer simulation. Image reproduced from a co-authored work, conducted and published as part of the PhD project [67]

In an agent-based simulation, agent interactions correspond to those performed by the components of the modelled system, quite faithfully to the actual behaviour of a biological process [75]. In process algebras, processes are concurrent, autonomous and reactive; all these properties are also shared by agents populating a multiagent environment, making process algebras suitable specification languages for agent-based systems.

A schematic representation of the transition from the biological domain (experimental data) to agent-based simulations via process-based models is provided in Figure 1.10. Agent-based models and simulations of molecular interactions are described and discussed in Part II of this manuscript.

Chapter 2

Process Calculi May Reveal the Equivalence Underlying RNA and Proteins*

2.1 Introduction

Ribonucleic acids (RNAs) and proteins are two classes of molecules that have drawn the interest of different scientific disciplines because of their fundamental roles in many biological processes. Discovering the qualitative information underlying the relationship between their structures and functions requires a thorough understanding of their folding. Indeed, from their linear sequence to their three-dimensional conformation, RNAs and proteins follow a similar path; the shapes they reach in this way allow them to perform comparable catalytic and structural tasks.

Investigating the reasons for the existence of such similar molecules led to the formulation of the RNA world hypothesis: RNA might be a "fossil" of an RNA world that existed on Earth before modern cells appeared, in which RNA fulfilled the roles of both DNA and proteins. This theory is still highly debated [44, 93]; indeed, beyond their similarities, proteins and RNAs show profound structural differences, which affect the way they perform their functions.

This chapter aims to provide a formal description of the folding process of proteins compared to that of RNAs; by highlighting their key properties, our purpose is to identify clues to the validity of the RNA world hypothesis. We focus our study on the interactions among monomers–the elementary units that compose RNA and protein linear sequences–and describe the whole folding process as the resulting behaviour of these interactions.

*This chapter is derived from a co-authored work, conducted and published as part of the PhD project: Maestri, S., Merelli, E., 2019. "Process calculi may reveal the equivalence lying at the heart of RNA and proteins". *Scientific Reports* 9, 559. CC BY 4.0. https://doi.org/10.1038/s41598-018-36965-1. S.M. implemented the method, performed the research and wrote the paper. E.M. supervised the research. Both the authors designed and reviewed the paper.

2.2 Results

The definition of the models we propose in this chapter is based on the idea that all the components involved in a system, and the communication media themselves, can be formally represented as processes. This approach has been applied to study biological systems by modelling entire molecules [13, 89]; however, it can be extended to analyse their substructures–or even their elementary units–and the interactions they perform.

The specification language that better suits our modelling of RNA and protein folding is the Calculus of Communicating Systems (CCS), proposed by Milner in 1989 [77]; thanks to this process algebra, it is possible to define the congruence of the folding processes in terms of *behavioural equivalence* and perform the model checking with the aid of automated tools (see Section 1.3.3 on page 28 for an introduction to these modelling and verification methods). Moreover, the whole folding process can be modelled as the result of sub-processes that proceed along a path made of discrete states; we capture this property by means of labelled transition systems (LTSs) [62].

We want to point out that some aspects that contribute to the folding process and can be relevant from a biological point of view are not included in our models. For example, we do not consider the role of helping molecules, such as the modulation performed by Mg^{2+} on the RNA folding or the contribution of molecular chaperones to protein folding [49, 51]. This decision is motivated by the idea of describing the folding process as behaviour resulting solely from the interactions among nucleotides and amino acids (in their respective strands) and the informational content carried by each of them. If, on the one hand, such an approach leads us to define an abstraction of the actual folding mechanisms, on the other, it allows us to formally prove the existence of distinguishing features of these processes that might be the basis of the very existence of both RNAs and proteins in cells. We want to prove that the inner potentiality of each monomer to interact with the others (in the same sequence) is the main property that determines the different structural complexity ultimately reachable by the two kinds of molecules.

To demonstrate this statement, we start by defining the models of the folding process as a sequence of folding steps, each contributing a new non-covalent interaction between two monomers. Because the folding process relies mainly upon the formation of weak and non-covalent interactions in both RNAs and proteins, the stabilisation function performed by covalent bonds (such as disulphide bridges between Cys residues) is negligible for our model definition.

We classify non-covalent interactions into three main categories:

- hydrogen bonds;
- electrostatic interactions (ionic and van der Waals);
- hydrophobic and hydrophilic interactions.

Hydrogen bonds can be considered electrostatic interactions, but due to their unique properties and central role in the folding process, they are categorised separately.

Even if the non-covalent interactions taken into account are the same for RNAs and proteins, the rules that allow two nucleotides to interact differ from those that govern the interplay of two

amino acids. Hence, we need to define two different models, one for each class of molecules. The highlighted differences affect the whole folding process and cause our models to show different traces, namely different sequences of transitions in their respective LTSs.

However, the expressiveness of process algebras allows us to identify an abstraction level in which these two processes show a congruence relation called *strong bisimilarity*; this means that they afford the same traces and that all the states they reach in such traces are equivalent [1]. At this specific level of abstraction, the two folding processes form structures with the same complexity, thus capable of expressing identical functions. If such an abstraction level corresponded to the actual folding process of RNAs and proteins, there would exist two different classes of molecules showing the same behaviour.

Our results concern the RNA world hypothesis due to the interpretation of the behavioural equivalence of RNA and protein folding under specific restrictions (as in Theorem 2.1). According to the RNA world hypothesis, in the early stages of cell evolution, RNA might have performed both structural and catalytic activities; as the complexity of cells increased, there was a need for molecules able to carry out more complex tasks. Our models show that cells cope with this necessity by forming molecules–namely proteins–whose elementary units perform interactions more complex than those of nucleotides. Towards the RNA world hypothesis, such molecules might be evolved, similarly to RNAs, as linear sequences of monomers able to fold up into three-dimensional structures, driven by free energy reduction.

2.2.1 Folding step

A *folding step* represents an iteration that allows the non-deterministic choice between one of the possible sub-processes describing the behaviour of the non-covalent interactions. It ensures that each of its sub-process complies with specific restrictions on its input and that the interaction has a negative *free-energy change*. The latter, denoted by ΔG and modelled with the `ΔG` process, can be negative (`ndg`), positive (`pdg`), or zero (`zdg`); a negative ΔG is an essential condition for an interaction to be performed. To capture the distinctive properties of RNA and protein folding beyond the common features described above, our model considers two folding step processes.

Definition 2.1 (Folding Step)**.** The *RNA folding step* and the *protein folding step* processes, denoted by $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$, respectively, are defined by the following CCS equations:

RNA folding step

$$
\begin{array}{lcl}
\mathcal{F}^s_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \mathtt{ub}.\mathcal{J}1_{\mathtt{n}} + \mathtt{ub}.\mathcal{J}2_{\mathtt{n}} + \mathtt{srsr}.\mathcal{J}1_{\mathtt{n}} + \\
 & & \mathtt{drdr}.\mathcal{J}1_{\mathtt{n}} + \mathtt{srdr}.\mathcal{J}1_{\mathtt{n}} + \mathtt{tpb}.\mathcal{J}1_{\mathtt{n}}; \\
\mathcal{J}1_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{srsr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{drdr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \\
 & & \mathtt{srdr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{tpb}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}}; \\
\mathcal{J}2_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta G_{\mathcal{P}_{\mathtt{b2}}} + \mathtt{ub}.\Delta G_{\mathcal{J}^h_{\mathtt{b}}} + \mathtt{srsr}.\Delta G_{\mathcal{P}_{\mathtt{b3}}} + \\
 & & \mathtt{drdr}.\Delta G_{\mathcal{P}_{\mathtt{b3}}} + \mathtt{srdr}.\Delta G_{\mathcal{P}_{\mathtt{b3}}}; \\
\Delta G_{\mathcal{J}^e_{\mathtt{b}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{b}}; \\
\Delta G_{\mathcal{J}^h_{\mathtt{b}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{b}}; \\
\Delta G_{\mathcal{P}_{\mathtt{b2}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b2}}; \\
\Delta G_{\mathcal{P}_{\mathtt{b3}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b3}};
\end{array}
$$

$$
\begin{array}{lcl}
\mathcal{P}_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}1_{\mathtt{b2}}; \\
\mathcal{B}1_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}2_{\mathtt{b2}}; \\
\mathcal{B}2_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}3_{\mathtt{b2}} + \overline{\mathtt{srsr}}.\mathcal{F}^s_{\mathtt{rna}} + \overline{\mathtt{drdr}}.\mathcal{F}^s_{\mathtt{rna}} + \\
 & & \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{B}3_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{P}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}1_{\mathtt{b3}}; \\
\mathcal{B}1_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}2_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{B}2_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}3_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{B}3_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}};
\end{array}
$$

$$
\begin{array}{lcl}
\mathcal{J}^e_{\mathtt{b}} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^s_{\mathtt{rna}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{J}^h_{\mathtt{b}} & \stackrel{\text{def}}{=} & hbi.\mathtt{I}_{\mathtt{rna}}; \\
\mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bb}}.\mathcal{S}; \\
\mathcal{S} & \stackrel{\text{def}}{=} & \overline{\mathtt{sb}}.\mathcal{F}^s_{\mathtt{rna}}.
\end{array}
$$

Protein folding step

group 1

$$
\begin{array}{lcl}
\mathcal{F}^s_{\mathtt{p}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\mathcal{J}1_{\mathtt{aa}} + \mathtt{aa}.\Delta G_{\mathcal{J}^h_{\mathtt{aa}}}; \\
\mathcal{J}1_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\Delta G_{\mathcal{J}^e_{\mathtt{aa}}} + \mathtt{aa}.\Delta G_{\mathcal{P}_{\mathtt{aa}}}; \\
\Delta G_{\mathcal{J}^e_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{aa}}; \\
\Delta G_{\mathcal{J}^h_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{aa}}; \\
\Delta G_{\mathcal{P}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa}};
\end{array}
$$

group 2

$$
\begin{array}{lcl}
\mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa1fnh}.\mathtt{NH}_{\mathtt{aa1}} + \\
 & & \mathtt{aa1fco}.\mathtt{CO}_{\mathtt{aa1}}; \\
\mathtt{NH}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fco}.\mathtt{CO}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fnh}.\mathtt{NH}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathtt{NH}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathcal{B}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{\mathtt{paa}}.\mathcal{F}^s_{\mathtt{p}};
\end{array}
$$

group 3

$$
\begin{array}{lcl}
\mathcal{J}^e_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^s_{\mathtt{p}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{p}}; \\
\mathcal{J}^h_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlsc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbsc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{esc}}.\mathcal{F}^s_{\mathtt{p}}; \\
\mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bsc}}.\mathcal{F}^s_{\mathtt{p}}.
\end{array}
\tag{2.1}
$$

The meanings of every subprocess and action label are provided in Tables 2.1 and 2.2, respectively.

Table 2.1 – Symbols used to denote the processes in Equation 2.1. The process name transliterations are necessary to construct the related LTS representations through the CAAL concurrency workbench [3], as shown throughout this chapter.

Process\State	Transliteration	Description
$\mathtt{NH}_{\mathtt{aa1}}$	AA1NH	first amino acid's free amino group
$\mathtt{CO}_{\mathtt{aa1}}$	AA1CO	first amino acid's free carboxyl group
$\mathtt{NH}_{\mathtt{aa2}}$	AA2NH	second amino acid's free amino group
$\mathtt{CO}_{\mathtt{aa2}}$	AA2CO	second amino acid's free carboxyl group
$\mathcal{J}^{e}_{\mathtt{aa}}$	AAEI	electrostatic interaction between amino acids
$\Delta\mathtt{G}_{\mathcal{J}^{e}_{\mathtt{aa}}}$	AAEIDG	ΔG of an AAEI
$\mathcal{J}^{h}_{\mathtt{aa}}$	AAHI	hydrophobic/hydrophilic interaction of an amino acid
$\Delta\mathtt{G}_{\mathcal{J}^{h}_{\mathtt{aa}}}$	AAHIDG	ΔG of an AAHI
$\mathcal{B}_{\mathtt{aa}}$	AAHB	hydrogen bonding between two amino acids
$\mathcal{J}1_{\mathtt{aa}}$	AAI1	non-specific amino acid interaction
$\mathcal{P}_{\mathtt{aa}}$	AAP	amino acid pairing
$\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}}$	AAPDG	ΔG of an AAP
$\mathcal{J}^{e}_{\mathtt{b}}$	BEI	electrostatic interaction between bases
$\Delta\mathtt{G}_{\mathcal{J}^{e}_{\mathtt{b}}}$	BEIDG	ΔG of a BEI
$\mathcal{B}\mathtt{X}_{\mathtt{b2}}(\mathtt{X}=1,2,3)$	BHBX (X = 1, 2, 3)	hydrogen bonding between two bases
$\mathcal{J}^{h}_{\mathtt{b}}$	BHI	hydrophobic interaction of bases
$\Delta\mathtt{G}_{\mathcal{J}^{h}_{\mathtt{b}}}$	BHIDG	ΔG of a BHI
$\mathcal{P}_{\mathtt{b2}}$	BP	base pairing
$\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b2}}}$	BPDG	ΔG of a BP
$\mathcal{J}\mathtt{X}_{\mathtt{n}}(\mathtt{X}=1,2)$	NIX (X = 1, 2)	non-specific nucleotide interaction
$\mathcal{F}^{s}_{\mathtt{p}}$	PFS	protein folding step
$\mathtt{I}_{\mathtt{p}}$	PI	protein inside
$\mathtt{O}_{\mathtt{p}}$	PO	protein outside
$\mathcal{F}^{s}_{\mathtt{rna}}$	RNAFS	RNA folding step
$\mathtt{I}_{\mathtt{rna}}$	RNAI	RNA inside
$\mathcal{S}$	S	base stacking
$\mathcal{B}\mathtt{X}_{\mathtt{b3}}(\mathtt{X}=1,2,3)$	TBHBX (X = 1, 2, 3)	hydrogen bonding between three bases
$\mathcal{P}_{\mathtt{b3}}$	TBP	triple base pairing
$\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b3}}}$	TBPDG	ΔG of a TBP

Table 2.2 – Action labels used in Equation 2.1.

Action label	Description
`aa`	amino acid
`aa1fco`	first amino acid's free carboxyl group
`aa1fnh`	first amino acid's free amino group
`aa2fco`	second amino acid's free carboxyl group
`aa2fnh`	second amino acid's free amino group
`bb`	buried bases
`bsc`	buried side chain
`dr`	double-ring base (purine)
`esc`	exposed side chain
`hb`	hydrogen bond
`hbsc`	hydrophobic side chain
hbi	hydrophobic interaction
`hlsc`	hydrophilic side chain
ii	ionic interaction
`ndg`	$\Delta G < 0$
`paa`	paired amino acids
`pdg`	$\Delta G > 0$
`sb`	stacked bases
`sr`	single-ring base (pyrimidine)
`tpb`	base triple
`ub`	unpaired base
vdwi	van der Waals interaction
`zdg`	$\Delta G = 0$

Both $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$ are structured in sub-processes that can be clustered in three main groups:

- group 1 determines the type of the elementary units involved in the current folding step, the interaction that is going to be established between them, and if its ΔG is negative;
- group 2 describes the formation of one or more hydrogen bonds between two units (unpaired or already paired);
- group 3 models the behaviour of ionic, van der Waals, and hydrophobic interactions.

In this first phase of our model definition, which aims to remain as faithful as possible to the biological folding process, the group 2 of sub-processes carries out the important task of limiting the maximum number of elementary units that can be linked by hydrogen bonds as well as the number of hydrogen bonds that can be generated between two units.

The hydrogen bond formation (in both Watson-Crick and Wobble base pair) is modelled generalising this process as an interaction between a purine (adenine or guanine) and a pyrimidine (uracil and cytosine) or between two paired bases and a third base (in this case, a generic purine or pyrimidine). Since purines are **d**ouble-**r**ing bases, they are labelled `dr`; pyrimidines, conversely, are **s**ingle-**r**ing bases and hence labelled `sr`. The base pairing is symmetric, thus `srdr` = `drsr`.

Regarding the number of hydrogen bonds in a base pair, our models allow them to be at least two and at most three. Conversely, the hydrogen bonds that link an unpaired base to a group of two already paired bases must be from one to three. We introduce these constraints because base pairs with a single hydrogen bond can be classified as variants of those linked by two, and the number of hydrogen bonds found in a base triplet is three to six [80].

In contrast with the base pairing of nucleotides, only a single hydrogen bond is allowed between two amino acids; however, there is no limitation in the length of a sequence of amino acids linked to one another via hydrogen bonds.

A complete description of the conventions adopted and the choices made to derive the two models from the biological folding processes can be found in Appendix A.

To construct the whole *RNA* and *protein folding* processes, the *RNA folding step* ($\mathcal{F}^s_{\mathtt{rna}}$) and *protein folding step* ($\mathcal{F}^s_{\mathtt{p}}$) processes are placed in parallel composition with the process $\Delta\mathtt{G}$, which represents the free energy variation during folding.

Definition 2.2 (RNA and protein folding)**.** Let $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$ be the processes defined in Equation 2.1 and `ndg`, `pdg`, and `zdg` be action labels representing the *folding step free-energy change* (negative, positive, or zero, respectively). The *RNA* and *protein folding* processes, denoted by $\mathcal{F}_{\mathtt{rna}}$ and $\mathcal{F}_{\mathtt{p}}$, respectively, are defined as follows:

$$\begin{aligned}
&\mathcal{F}_{\mathtt{rna}} \stackrel{\text{def}}{=} (\mathcal{F}^s_{\mathtt{rna}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\mathcal{F}_{\mathtt{p}} \stackrel{\text{def}}{=} (\mathcal{F}^s_{\mathtt{p}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\text{where } \Delta\mathtt{G} \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta\mathtt{G} + \overline{\mathtt{ndg}}.\Delta\mathtt{G} + \overline{\mathtt{zdg}}.\Delta\mathtt{G}.
\end{aligned} \tag{2.2}$$

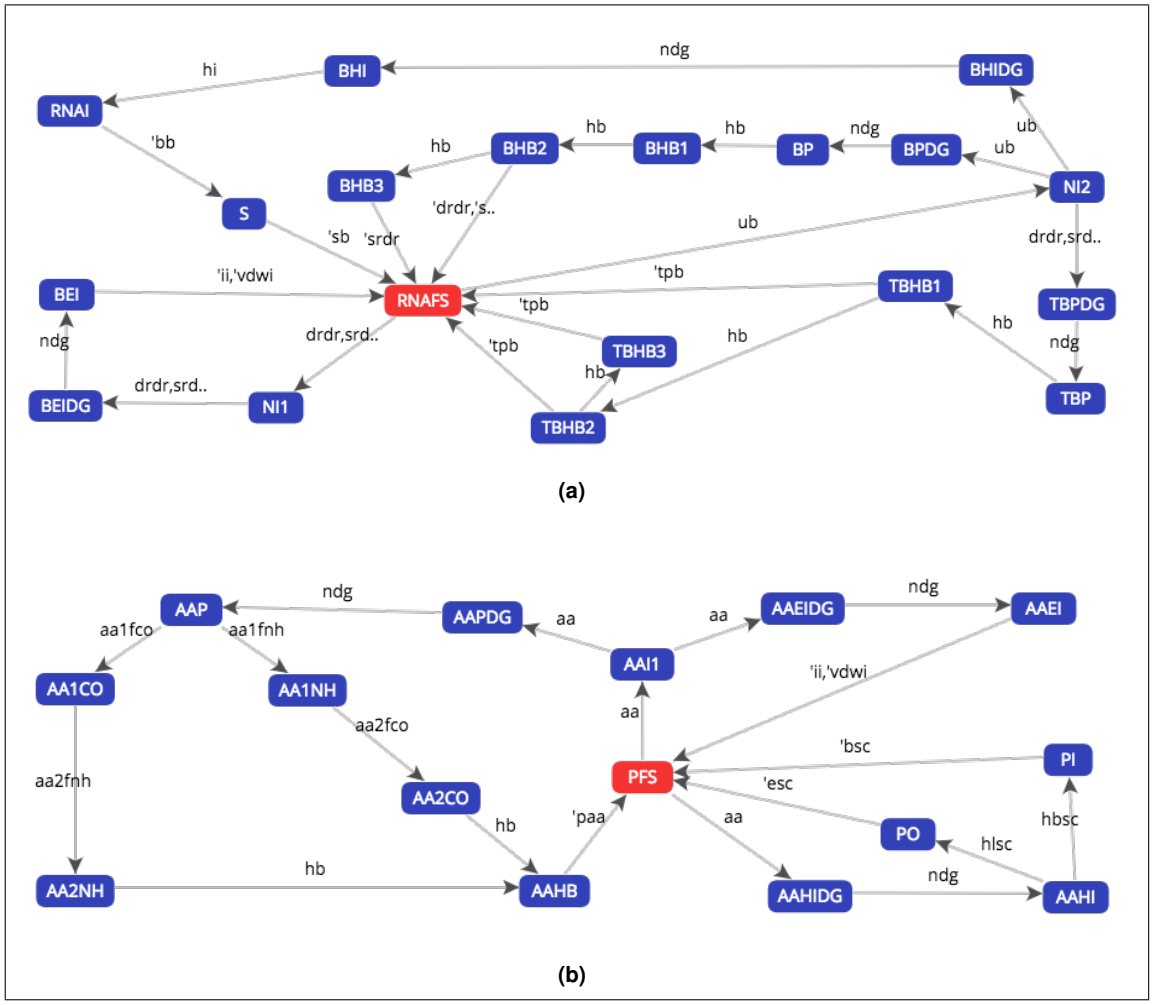


Figure 2.1 – Labelled transition systems of the $\mathcal{F}^{s}_{\mathrm{rna}}$ process **(a)**, transliterated RNAFS, and of the $\mathcal{F}^{s}_{\mathrm{p}}$ process **(b)**, transliterated PFS, generated with the CAAL web-based concurrency workbench [3]. Each process and label represented in the LTSs is described in Tables 2.1 and 2.2.

2.2.2 Bisimilarity equivalence

The verification that two processes of the proposed models are bisimilar (i.e., if they show the same behaviour) is based on *bisimulation games*, namely game characterisations of the bisimilarity. Informally, we can define a bisimulation game as a sequence of rounds in which the LTSs of two processes are compared. The game explores the LTSs by pairs of states (called configurations).

Starting from an initial configuration, two players–an attacker and a defender–try to perform in turn a transition based on one of the two LTSs; the game is begun by the attacker, who decides

which transition of the initial configuration to perform (and hence which of the two LTSs to explore). The choice made in each turn determines the configuration explored in the next one by the other player. A finite play of the game is lost by the player who cannot make a move from the current configuration. If the play is infinite (as in the case in which a cycle is detected), the defender is considered to win because the attacker cannot distinguish the behaviour of the two processes.

Two states are strongly bisimilar if and only if the defender has a *universal winning strategy* (i.e., he can always win the game, regardless of how the attacker selects his moves) in the strong bisimulation game that starts from the configuration made up of such states.

If we try to prove the behavioural equivalence of the $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$ processes, we can observe, from the LTSs in Figure 2.1, that the bisimulation game ends after only one move, regardless of the choice made by the attacker, with the defeat of the defender.

As an example, if the attacker chooses the transition $\mathsf{RNAFS} \xrightarrow{\mathtt{ub}} \mathsf{NI2}$ on the LTS of RNAFS, the defender has no available transition on the LTS of PFS to respond.

This first verification proves that a model strictly faithful to biological folding leads us to define processes whose behaviours are not equivalent.

2.2.3 High abstraction level model

We might wonder if *there is an abstraction level at which the two folding processes would show a behavioural equivalence*. As proved in the remainder of this chapter, this level of abstraction can actually be defined. Its construction, however, requires generalising the non-covalent interactions and imposing some limitations on the expressiveness of the protein folding process.

The first of the two modifications mentioned above can be achieved by:

- redefining nucleotides and amino acids as general elementary units, which can be paired or unpaired;
- abstracting from the specificity of each pairing process by no longer taking into account the number of hydrogen bonds formed between two (or three) paired units;
- generalising the hydrophobic interactions to their key feature of burying the hydrophobic molecules while exposing the hydrophilic ones (no longer considering the stacking process typical of the hydrophobic interactions of nucleotides).

These adjustments to the model do not affect the key properties of each non-covalent interaction; therefore, the model is still fairly faithful to the biological process. However, they are also not sufficient to obtain a behavioural equivalence between the folding processes of RNAs and proteins: we still need to limit the proteins' folding capability by reducing to three the number of amino acids that can interact through hydrogen bonds (because in RNA we can form at most base triples).

Based on these premises, we can define a *folding step high abstraction function* $\mathcal{H}$, which maps each folding step to its respective higher abstraction level described above.

Definition 2.3 (Folding step high abstraction function)**.** Given $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$ as in Equation 2.1 and a process P, we define the *folding step high abstraction function* $\mathcal{H}$ as follows:

$$\mathcal{H}(\mathsf{P}) = \begin{cases} \mathbb{F}^s_{\mathtt{rna}}, & \text{if } \mathsf{P} = \mathcal{F}^s_{\mathtt{rna}} \\ \mathbb{F}^s_{\mathtt{p}}, & \text{if } \mathsf{P} = \mathcal{F}^s_{\mathtt{p}} \end{cases} \tag{2.3}$$

where $\mathbb{F}^s_{\mathtt{rna}}$ and $\mathbb{F}^s_{\mathtt{p}}$ denote, respectively, the *high abstraction RNA folding step* and *high abstraction protein folding step* processes, whose behaviours are given by the following defining equations:

High abstraction RNA folding step

$$\begin{array}{lcl|l|lcl}
\mathbb{F}^s_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\mathcal{J}1_{\mathtt{n}} + \mathtt{pu}.\mathcal{J}1_{\mathtt{n}} + & & \mathbb{F}^s_{\mathtt{p}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\mathcal{J}1_{\mathtt{aa}} + \mathtt{pu}.\mathcal{J}1_{\mathtt{aa}} + \\
 & & \mathtt{uu}.\Delta G_{\mathcal{J}^h_{\mathtt{n}}} + \mathtt{uu}.\mathcal{J}2_{\mathtt{n}} + & & & & \mathtt{uu}.\Delta G_{\mathcal{J}^h_{\mathtt{aa}}} + \mathtt{uu}.\mathcal{J}2_{\mathtt{aa}} + \\
 & & \mathtt{tpu}.\mathcal{J}1_{\mathtt{n}}; & & & & \mathtt{tpu}.\mathcal{J}1_{\mathtt{aa}}; \\
\mathcal{J}1_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{pu}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + & & \mathcal{J}1_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{J}^e_{\mathtt{aa}}} + \mathtt{pu}.\Delta G_{\mathcal{J}^e_{\mathtt{aa}}} + \\
 & & \mathtt{tpu}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}}; & & & & \mathtt{tpu}.\Delta G_{\mathcal{J}^e_{\mathtt{aa}}}; \\
\mathcal{J}2_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{P}_{\mathtt{b2}}} + \mathtt{pu}.\Delta G_{\mathcal{P}_{\mathtt{b3}}}; & \text{group 1} & \mathcal{J}2_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{P}_{\mathtt{aa}}} + \mathtt{pu}.\Delta G_{\mathcal{P}_{\mathtt{aa3}}}; \\
\Delta G_{\mathcal{J}^e_{\mathtt{b}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{b}}; & & \Delta G_{\mathcal{J}^e_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{aa}}; \\
\Delta G_{\mathcal{J}^h_{\mathtt{n}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{n}}; & & \Delta G_{\mathcal{J}^h_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{aa}}; \\
\Delta G_{\mathcal{P}_{\mathtt{b2}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b2}}; & & \Delta G_{\mathcal{P}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa}}; \\
\Delta G_{\mathcal{P}_{\mathtt{b3}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b3}}; & & \Delta G_{\mathcal{P}_{\mathtt{aa3}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa3}};
\end{array}$$

High abstraction protein folding step

$$\begin{array}{lcl|l|lcl}
\mathcal{P}_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{sr}} + \mathit{hb}.\mathtt{B}_{\mathtt{dr}}\mathtt{B}_{\mathtt{dr}} + & & & & \\
 & & \mathit{hb}.\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{dr}}; & & & & \\
\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{sr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; & & \mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{NC} + \mathit{hb}.\mathtt{CN}; \\
\mathtt{B}_{\mathtt{dr}}\mathtt{B}_{\mathtt{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; & & \mathtt{NC} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{p}}; \\
\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; & & \mathtt{CN} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{p}}; \\
\mathcal{P}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{U}_{\mathtt{b3}}; & \text{group 2} & \mathcal{P}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{U}_{\mathtt{aa3}}; \\
\mathtt{U}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}^s_{\mathtt{rna}}. & & \mathtt{U}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}^s_{\mathtt{p}}; \\
 & & & & & & \\
\mathcal{J}^e_{\mathtt{b}} & \stackrel{\text{def}}{=} & \overline{\mathit{ii}}.\mathbb{F}^s_{\mathtt{rna}} + \overline{\mathit{vdwi}}.\mathbb{F}^s_{\mathtt{rna}}; & & \mathcal{J}^e_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{\mathit{ii}}.\mathbb{F}^s_{\mathtt{p}} + \overline{\mathit{vdwi}}.\mathbb{F}^s_{\mathtt{p}}; \\
\mathcal{J}^h_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{rna}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{rna}}; & & \mathcal{J}^h_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}^s_{\mathtt{rna}}; & \text{group 3} & \mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}^s_{\mathtt{p}}; \\
\mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}^s_{\mathtt{rna}}. & & \mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}^s_{\mathtt{p}}.
\end{array} \tag{2.4}$$

The process symbols and action labels that are different from those in Equation 2.1 are described in Tables 2.3 and 2.4, respectively. For a phased construction of the folding step high abstraction models, see Appendix Section A.1.7.

Table 2.3 – Processes of the folding step high abstraction models defined in Equation 2.4. The process name transliterations are used in the model LTSs to perform the bisimulation game set up to prove Theorem 2.1. The missing symbols are described in Table 2.1.

Process\State	Transliteration	Description
$\mathcal{J}\mathrm{X}_{\mathtt{aa}}(\mathrm{X}=1,2)$	AAIX (X = 1, 2)	non-specific amino acid interaction
$\mathtt{C}$	C	amino acid's carboxyl group
$\mathrm{B}_{\mathtt{dr}}$	DR	double-ring base (purine)
$\Delta\mathtt{G}$	FSDG	folding step delta G
$\mathtt{N}$	N	amino acid's amino group
$\mathcal{J}^{h}_{\mathtt{n}}$	NHI	hydrophobic/hydrophilic interaction of a nucleotide
$\Delta\mathtt{G}_{\mathcal{J}^{h}_{\mathtt{n}}}$	NHIDG	ΔG of an NHI
$\mathbb{F}^{s}_{\mathtt{p}}$	PFS	protein folding step
$\mathbb{F}^{s}_{\mathtt{rna}}$	RNAFS	RNA folding step
$\mathtt{O}_{\mathtt{rna}}$	RNAO	RNA outside
$\mathrm{B}_{\mathtt{sr}}$	SR	single-ring base (pyrimidine)
$\mathcal{P}_{\mathtt{aa3}}$	TAAP	triple amino acid pairing
$\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa3}}}$	TAAPDG	ΔG of a TAAP
$\mathrm{U}_{\mathtt{aa3}}$	TAAU	amino acid triple unit
$\mathrm{U}_{\mathtt{b3}}$	TBU	base triple unit

Table 2.4 – Action labels specific to the folding step high abstraction models defined in Equation 2.4; the remaining model labels are described in Table 2.2.

Action label	Description
`bc`	buried component
`ec`	exposed component
hb	hydrogen bonding
`hbc`	hydrophobic component
`hlc`	hydrophilic component
`pu`	paired unit
`tpu`	triple unit
`uu`	unpaired unit

The definition of these new models can be considered an important result since it is possible to prove that, *at this level of abstraction, RNA folding and protein folding show the same behaviour.*

Theorem 2.1. *Let* $\mathbb{F}^s_{\mathtt{rna}} = \mathcal{H}(\mathcal{F}^s_{\mathtt{rna}})$ *and* $\mathbb{F}^s_{\mathtt{p}} = \mathcal{H}(\mathcal{F}^s_{\mathtt{p}})$*, then* $\mathbb{F}^s_{\mathtt{rna}}$ *and* $\mathbb{F}^s_{\mathtt{p}}$ *are strongly bisimilar* ($\mathbb{F}^s_{\mathtt{rna}} \sim \mathbb{F}^s_{\mathtt{p}}$).

Proof. The proof is provided via a bisimulation game. A winning strategy of the defender starts from the pair of states ($\mathbb{F}^s_{\mathtt{rna}}$,$\mathbb{F}^s_{\mathtt{p}}$) of the model's LTSs, transliterated (RNAFS,PFS) as in Figure 2.2. The play, sown in Table 2.5, demonstrates that RNAFS ~ PFS, i.e., that the two processes are strongly bisimilar (see Section 1.3.3). □

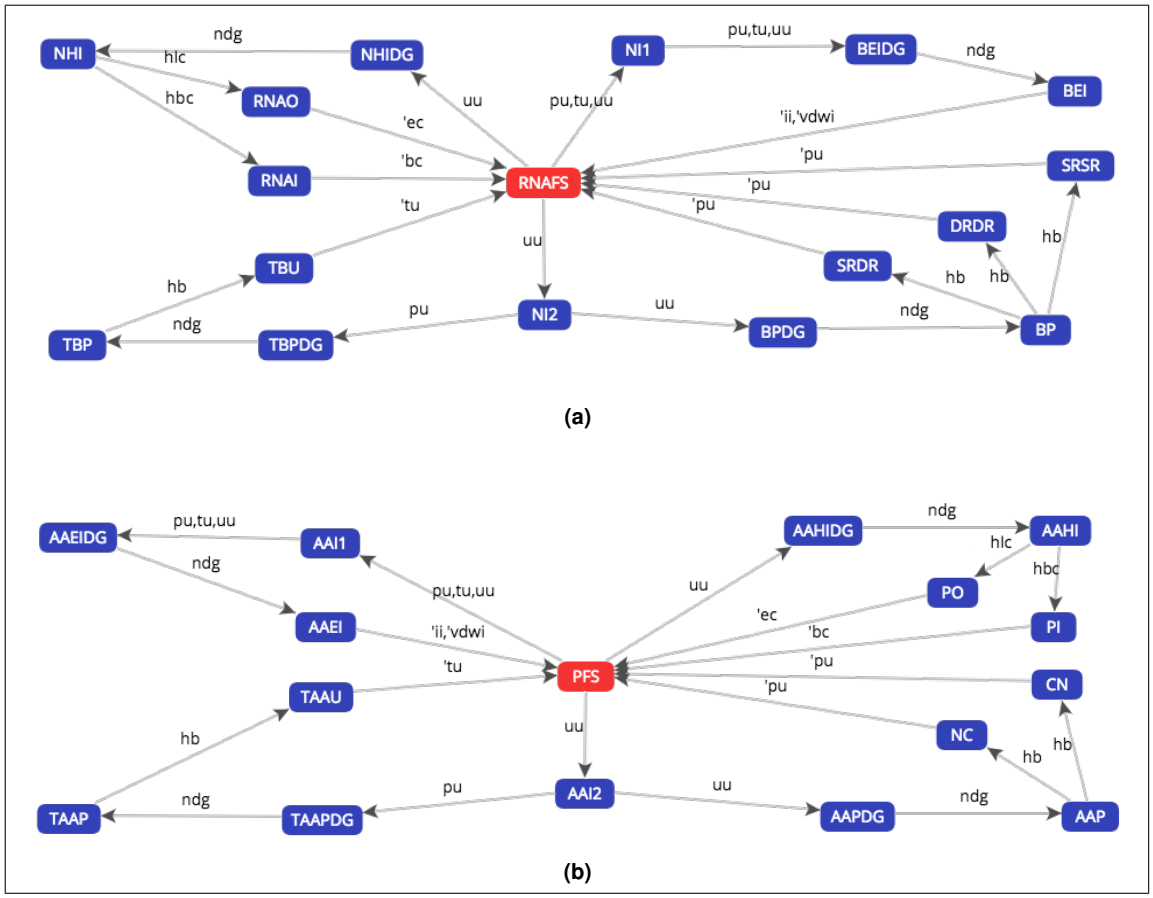


Figure 2.2 – Labelled transition systems of the $\mathbb{F}^s_{\mathtt{rna}}$ process **(a)**, transliterated RNAFS, and of the $\mathbb{F}^s_{\mathtt{p}}$ process **(b)**, transliterated PFS, generated with the CAAL web-based concurrency workbench [3]. Subprocesses and labels shown in the LTSs are described in Tables 2.1, 2.2, 2.3 and 2.4.

Round	Current configuration	Attacker	Defender
Round 1	(RNAFS,PFS)	$\mathsf{RNAFS} \xrightarrow{\mathsf{uu}} \mathsf{NI2}$	$\mathsf{PFS} \xrightarrow{\mathsf{uu}} \mathsf{AAI2}$
Round 2	(NI2,AAI2)	$\mathsf{NI2} \xrightarrow{\mathsf{uu}} \mathsf{BPDG}$	$\mathsf{AAI2} \xrightarrow{\mathsf{uu}} \mathsf{AAPDG}$
Round 3	(BPDG,AAPDG)	$\mathsf{BPDG} \xrightarrow{\mathsf{ndg}} \mathsf{BP}$	$\mathsf{AAPDG} \xrightarrow{\mathsf{ndg}} \mathsf{AAP}$
Round 4	(BP,AAP)	$\mathsf{BP} \xrightarrow{\mathsf{hb}} \mathsf{SRDR}$	$\mathsf{AAP} \xrightarrow{\mathsf{hb}} \mathsf{CN}$
Round 5	(SRDR,CN)	$\mathsf{SRDR} \xrightarrow{\overline{\mathsf{pu}}} \mathsf{RNAFS}$	$\mathsf{CN} \xrightarrow{\mathsf{uu}} \mathsf{PFS}$
Round 6	(RNAFS,PFS)	A cycle has been detected	Defender wins

Table 2.5 – Winning strategy of the defender in the strong bisimulation game that compares the $(\mathbb{F}^{s}_{\mathtt{rna}}, \mathbb{F}^{s}_{\mathtt{p}})$, pair of processes, transliterated (RNAFS,PFS). The result of this play proves that RNAFS ~ PFS, i.e. that the two processes are strongly bisimilar.

Definition 2.4 (High abstraction RNA and protein folding)**.** Let $\mathbb{F}^{s}_{\mathtt{rna}}$ and $\mathbb{F}^{s}_{\mathtt{p}}$ be the processes generated from $\mathcal{F}^{s}_{\mathtt{rna}}$ and $\mathcal{F}^{s}_{\mathtt{p}}$ through $\mathcal{H}$, and ndg, pdg, and zdg be action labels representing the *folding step free-energy change* (as in Definition 2.2). The *high abstraction RNA folding* and *high abstraction protein folding* processes, denoted by $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$, respectively, are defined as follows:

$$\begin{aligned}
&\mathbb{F}_{\mathtt{rna}} \stackrel{\text{def}}{=} (\mathbb{F}^{s}_{\mathtt{rna}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\mathbb{F}_{\mathtt{p}} \stackrel{\text{def}}{=} (\mathbb{F}^{s}_{\mathtt{p}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\text{where } \Delta\mathtt{G} \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta\mathtt{G} + \overline{\mathtt{ndg}}.\Delta\mathtt{G} + \overline{\mathtt{zdg}}.\Delta\mathtt{G}.
\end{aligned} \tag{2.5}$$

Theorem 2.2. $\mathbb{F}_{\mathtt{rna}}$ *and* $\mathbb{F}_{\mathtt{p}}$ *are strongly bisimilar* ($\mathbb{F}_{\mathtt{rna}} \sim \mathbb{F}_{\mathtt{p}}$)*.*

Proof. As proved by Milner [77], given two processes P and Q, such that P ~ Q, the following two rules hold:

P | R ~ Q | R and R | P ~ R | Q , for each process R

P \ L ~ Q \ L, for each set of labels L.

Therefore, based on Theorem 2.1 and Definition 2.4, $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$ are strongly bisimilar.

□

Proof to Theorem 2.2 can also be obtained with the aid of an automated tool; in Figure 2.3, we show the results of the bisimulation game performed with CAAL concurrency workbench on the processes $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$, transliterated RNAFOLDING and PFOLDING, respectively.

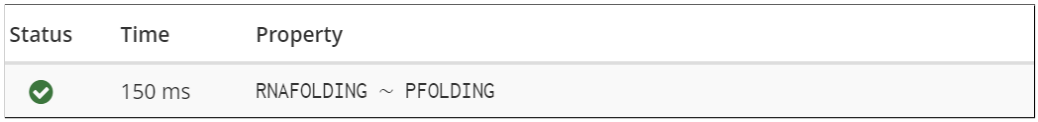


Figure 2.3 – Bisimulation game performed with CAAL concurrency workbench [3]; it shows that the $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$ processes, transliterated RNAFOLDING and PFOLDING, respectively, are strongly bisimilar, as the checkmark on the "Status" column indicates.

Through the construction of the *folding step high abstraction function* $\mathcal{H}$, we have formally demonstrated the existence of an abstraction level at which the folding processes of RNAs and proteins show the same behaviour and hence can generate three-dimensional structures of the same complexity; we refer to this abstraction level as *RNA and protein congruence level*.

2.3 Discussion

Starting from the models of RNA and protein folding, we have demonstrated how it is possible to formally define an abstraction level at which the two processes show behavioural equivalence. We have formally proved how it is possible to reach such an equivalence by reducing the complexity of the structures expressible through protein folding. This result can be interpreted as a clue that, at a point in the early evolution of life on Earth, proteins emerged to meet the need for molecules that could more effectively carry out the functions performed by RNA molecules and cope with more complex tasks. Nevertheless, we are well aware that we leave many questions unanswered regarding the RNA world theory, such as what role RNA would have played in storing genetic information; but it is not the objective of this manuscript to provide definitive proof of the theory mentioned above. However, we are equally convinced that our work lays a solid foundation for further developments in this direction.

Thanks to our models, we can infer the complexity of a biological structure, and hence of its function, based on the properties of its elementary components. In the case of RNAs and proteins, the distinguishing features of their respective folding processes have been identified and modelled only based on the known properties of the interactions that pair nucleotides (in RNAs) and amino acids (in proteins).

2.4 Conclusions

Due to its expressiveness, CCS turned out to be suitable for defining models based on the approach proposed throughout this chapter. The use of process algebras to describe molecular interactions can highlight the relationship between the complexity of the functions carried out by a biological entity and the type of interactions tying the elementary units that compose its structure.

This idea could be extended to the definition of predictive models of many other classes of biological molecules and processes by considering all the fundamental dynamics characterising

a biological system. For example, in Chapter 3, we define formal models of the entire gene expression process to study gene mutations that cause protein misfolding [32, 46].

Differently from the approach proposed in Part II of this manuscript, this chapter does not describe a simulation-based tool, but rather a theoretical way to acquire new knowledge about the studied systems. However, we have not aimed to define a new theory but a new methodology to understand biological behaviours by analysing the complexity of the interactions characterising living systems. Moreover, our work can be placed in the context of the topological analysis of the folding process [68, 73, 96].

Although the results proposed in the present chapter are based on the construction of algebraic models through process calculi, they actually provide us with factual knowledge. We believe that mathematics is not about human activity or phenomena; it is about extracting and formalising ideas and their manifold consequences [98].

Chapter 3

An Algebraic Approach to the Study of Protein Misfolding

3.1 Introduction

Formal methods have long been adopted in computer science for software specification and design; in recent years, they have also been effectively applied to model biological systems, especially with the aim of analysing the interactions occurring among their components [13, 20, 24, 27, 37].

Following this idea, in Chapter 2, we modelled the folding processes of ribonucleic acids (RNAs) and proteins in terms of the interactions performed by their monomeric units. For that purpose, we leveraged the expressivity of Milner's Calculus of Communicating Systems (CCS) [77], which allowed us to formally define a *congruence level* where such biological processes are bisimilar, namely showing equivalent behaviours. This abstraction level was obtained by reducing the complexity of the protein folding process and, thus, of the structures it can express.

We propose a type of investigation that, although strongly theoretical, has proven capable of providing new knowledge on the modelled processes; it also brought a different perspective on biological behaviours that, while well-known, have been studied mainly through experimental approaches. However, outside the congruence level, RNAs and proteins exhibit remarkable differences, both structural and functional, whose algebraic properties are yet to be explored.

This chapter goes a step further in addressing this issue by using CCS and Hennessy-Milner logic (HML) [54, 63] to model the processes that express genetic information in the form of RNA and protein structures and compare the folded conformations of the generated molecules. We focus on a class of pathologies that affect the folding process to study how they operate, in RNAs and proteins, on structural components of different complexity; specifically, we formally describe how the mutation even of a single nucleotide in a gene (point mutation) can alter the final conformation of a protein, while it is harmless for the structure of RNAs.

This analysis involves a formal description of how such pathologies, which originate as errors in the genetic code, might propagate through each step of gene expression, evading the cell's error detection and affecting both RNA and protein sequences. The adopted approach relies on a model of gene expression that is specifically defined over the transformations undergone by the informational content; it highlights the possible paths (correct or wrong) the information can follow from the DNA of a gene to the ribonucleotide sequence of an RNA molecule and, finally, to the amino acids of a protein's polypeptide chain.

We use sickle-cell anaemia, a well-known haemoglobin disease, as a case study to investigate such properties.

3.2 Results

3.2.1 Process-based models of gene expression

A *gene* is a deoxyribonucleic acid (DNA) string that codes for a functional product, which we consider to be a protein. In fact, other types of molecules can be generated, such as ribozymes, which carry out catalytic functions similar to those of proteins (see Section 1.2.4). Although this chapter aims to compare the process that generates both RNA and proteins, we will consider only the coding role of RNA.

The expression of a gene is carried out by three main processes: *transcription*, *RNA processing*, and *translation*. In what follows, we provide an algebraic model for each of them: through CCS, we represent their behaviour, whose constraints are specified with the aid of HML (see Section 1.3 for details on these two specification techniques). We remark that the models we are going to describe are strictly theoretical and aimed at capturing the core informational properties of the processes contributing to the expression of a gene. Based on these premises, we abstract DNAs and RNAs as nucleotide strings, while proteins are strings of amino acids.

As the first step in modelling gene expression, we thus introduce the set of nucleotides

$$N = \{\mathtt{a}, \mathtt{c}, \mathtt{g}, \mathtt{t}, \mathtt{u}\} \tag{3.1}$$

where each letter, in our process-based settings, is an *action label* that stands, respectively, for *adenine*, *cytosine*, *guanine*, *thymine*, and *uracil*. From the biological perspective, these are the bases that characterise and identify each nucleotide. Since a DNA sequence should not contain uracil, while RNA does not have thymine bases, it is useful to consider the following two subsets of N:

$$N_{dna} = N - \{\mathtt{u}\} = \{\mathtt{a}, \mathtt{t}, \mathtt{c}, \mathtt{g}\} \tag{3.2}$$

$$N_{rna} = N - \{\mathtt{t}\} = \{\mathtt{a}, \mathtt{u}, \mathtt{c}, \mathtt{g}\} \tag{3.3}$$

Transcription

As the first process involved in gene expression, the *gene transcription* takes the DNA sequence of a gene as a template to produce an RNA molecule (*transcript*). Each gene codes for a specific protein; therefore, the transcript must contain a copy of such definite information. The process is mainly carried out by a molecular complex called *RNA polymerase* (RNApol) [117].

In this perspective, a gene is a string of nucleotides enclosed between the labels $\mathbb{p}$ and $\mathbb{t}$, respectively denoting the transcription process's *promoter* and *terminator*. More precisely, the $\mathbb{p}$ label generalises the system of nucleotide sequences (e.g., TATA and CAAT boxes) and proteins (e.g., general transcription factors) that allows the transcription to initiate at the beginning of the gene; similarly, the $\mathbb{t}$ label indicates the end of the gene.

Although a gene could be intuitively represented as a string of N^+_{dna}, which is the non-empty set of all possible strings of the alphabet N_{dna}, a *mutation*[1] may lead to the substitution of a thymine base with uracil. Therefore, we consider a gene as a string in N^+.

Definition 3.1 (Gene)**.** Given the sets $N_g = N \cup \{\mathbb{p}, \mathbb{t}\}$ and

$$G = \{\gamma \mid \exists\, \delta = \texttt{"}\mathbb{p}\texttt{"}\ \gamma\ \texttt{"}\mathbb{t}\texttt{"},\ \delta \in N_g^+,\ \gamma \in N^+,\ g_{min} \leq |\gamma| \leq g_{max},\ \ g_{min}, g_{max} \in \mathbb{N}^+\} \tag{3.4}$$

a *gene* is a string $\gamma \in G$.

The length of the string γ, denoted by $|\gamma|$, is bounded by the two parameters g_{min} and g_{max}, which abstract, respectively, the minimum and maximum numbers of nucleotides of a gene that are experimentally retrievable or axiomatically defined. In our setting, g_{min} and g_{max} are purely theoretical, and their actual values are not considered in the models. "$\mathbb{p}$" and "$\mathbb{t}$" denote the strings containing only the label $\mathbb{p}$ and $\mathbb{t}$, respectively; the string δ is thus represented by a *concatenation of strings*, meaning that it is constructed by joining a series of strings together, end-to-end, without gaps. This approach to parameter and string definition will be taken as implicit in the rest of this chapter.

The *transcription process* $\mathcal{T}$ converts a string $\gamma \in G$ to a sequence of action labels in N^+_{rna} representing the *transcript*.

Definition 3.2 (RNA transcript)**.** Given $N_t = N_{rna} \cup \{3, 5\}$ and

$$T = \{\theta \mid \exists\, \chi = \texttt{"}5\texttt{"}\ \theta\ \texttt{"}3\texttt{"},\ \chi \in N_t^+,\ \theta \in N^+_{rna},\ g_{min} \leq |\theta| \leq g_{max}\} \tag{3.5}$$

an *RNA transcript* is a string $\theta \in T$.

The string θ is bounded in its length by g_{min} and $g_{max} \in \mathbb{N}^+$ because it has the same number of nucleotides as the source gene. 5 and 3 are not natural numbers but labels denoting, respectively, the 5' and 3' end of the transcript. They are called that way because they indicate the

[1] Mutations may occur due to hereditary diseases or during the replication process, that is, the process that duplicates a DNA strand.

extremities of the transcript; precisely, the terminus that exposes the phosphate group of the last nucleotide's fifth carbon, in the first case, and the terminus ending with the third carbon's hydroxyl group, in the second case. We made them explicit to provide entry and end points to the subsequent *processing* phase of the gene expression model.

The $\mathcal{T}$ *process* starts from the $\mathtt{p}$ label and proceeds (working on one nucleotide at a time) until it reaches the $\mathtt{t}$ label. We thus model an *RNApol* molecule as the process $\mathtt{RNA}_{\mathtt{pol}}$ that takes as input a nucleotide ($\mathtt{a}, \mathtt{t}, \mathtt{c}$, or $\mathtt{g}$) and produces as output the *base-pairing* with its complementary ribonucleotide ($\mathtt{u}, \mathtt{a}, \mathtt{g}, \mathtt{c}$, respectively), adding the latter to the sequence of the transcript. When a *mispairing* occurs, non-complementary bases are associated in a base pair. A *proofreading process* $\mathcal{R}$ takes the base pairs generated by $\mathtt{RNA}_{\mathtt{pol}}$ and, in case of mispairing, provides the correct nucleotide that has to be added to the transcript; however, $\mathcal{R}$ can make mistakes and leave a mispairing uncorrected [110]. The sets of the *canonical base pairs B* and that of the *mispaired bases M* contain the following action labels:

$$B = \{au,\ cg,\ ta,\ gc\}, \qquad M = \{ac,\ tg,\ ca,\ gu\} \tag{3.6}$$

Definition 3.3 (Transcription)**.** Let $\mathcal{G}$ be the process that initiates the expression of a gene (namely, a string of labels $\gamma \in G$). Given the set of action labels $L_{\mathcal{T}} = N \cup B \cup M \cup \{3, 5, \mathtt{p}, \mathtt{t}\}$, for each γ, the transcription process $\mathcal{T}$, such that $\mathcal{G} \overset{\overline{\gamma}}{\Rightarrow} \mathcal{T}$, is defined as follows:

$$\begin{aligned}
&\mathcal{T} \overset{\text{def}}{=} \mathtt{p}.\overline{5}.\mathtt{RNA}_{\mathtt{pol}};\\
&\mathtt{RNA}_{\mathtt{pol}} \overset{\text{def}}{=} \mathtt{a}.\mathtt{A}_t + \mathtt{t}.\mathtt{T}_t + \mathtt{c}.\mathtt{C}_t + \mathtt{g}.\mathtt{G}_t + \mathtt{t}.\overline{3}.0;\\
&\mathtt{A}_t \overset{\text{def}}{=} \overline{au}.\mathcal{R} + \overline{ac}.\mathcal{R};\\
&\mathtt{T}_t \overset{\text{def}}{=} \overline{ta}.\mathcal{R} + \overline{tg}.\mathcal{R};\\
&\mathtt{C}_t \overset{\text{def}}{=} \overline{cg}.\mathcal{R} + \overline{ca}.\mathcal{R};\\
&\mathtt{G}_t \overset{\text{def}}{=} \overline{gc}.\mathcal{R} + \overline{gu}.\mathcal{R};\\
&\mathcal{R} \overset{\text{def}}{=} au.\mathtt{U}_r + ac.\mathtt{U}_r + ta.\mathtt{A}_r + tg.\mathtt{A}_r + cg.\mathtt{G}_r + ca.\mathtt{G}_r + gc.\mathtt{C}_r + gu.\mathtt{C}_r +\\
&\qquad\qquad ac.\mathtt{C}_r + tg.\mathtt{G}_r + ca.\mathtt{A}_r + gu.\mathtt{U}_r;\\
&\mathtt{A}_r \overset{\text{def}}{=} \overline{\mathtt{a}}.\mathtt{RNA}_{\mathtt{pol}};\\
&\mathtt{U}_r \overset{\text{def}}{=} \overline{\mathtt{u}}.\mathtt{RNA}_{\mathtt{pol}};\\
&\mathtt{C}_r \overset{\text{def}}{=} \overline{\mathtt{c}}.\mathtt{RNA}_{\mathtt{pol}};\\
&\mathtt{G}_r \overset{\text{def}}{=} \overline{\mathtt{g}}.\mathtt{RNA}_{\mathtt{pol}};
\end{aligned} \tag{3.7}$$

where

- $\mathtt{A}_t$, $\mathtt{T}_t$, $\mathtt{C}_t$, $\mathtt{G}_t$, and $\mathtt{U}_t$ are states describing the behaviour of the $\mathtt{RNA}_{\mathtt{pol}}$ process when it takes the corresponding nucleotide as input: this behaviour is defined by the non-deterministic choice between the correct base-pairing and a mispairing;

- $\mathtt{A}_r$, $\mathtt{T}_r$, $\mathtt{C}_r$, $\mathtt{G}_r$, and $\mathtt{U}_r$ are states indicating which output will be provided accordingly to the choice made by the $\mathcal{R}$ (proofread) process; the labels in the first row of the $\mathcal{R}$ definition show the proper base-pairing (including error corrections), while the second row contains the cases in which the proofreading process does not recognise a mispairing.

The meaning of the process names and action labels used in this and the following CCS defining equations are provided in Tables 3.2 and 3.3.

Since considering specific transcript lengths is beyond the scope of this chapter, we generalise them as determined by the $|\gamma|$ value of each string $\gamma \in G$. Therefore, based on the algebraic definition of the $\mathcal{T}$ process, we can provide the following specification.

Given the set of labels $L'_{\mathcal{T}} = \{\overline{3}, \overline{5}, b_1, b_2, \overline{b_1 b_2}, \mathbb{p}, \mathbb{t}\}$, where $b_1 \in N$, $b_2 \in N_{rna}$, and $\overline{b_1 b_2} \in B \cup M$, for each string $\gamma \in G$,

$$\begin{aligned} &\mathcal{T} \vDash \langle \mathbb{p} \rangle \langle \overline{5} \rangle \mathcal{T} \\ &\mathcal{T} \equiv \bigwedge_{|\gamma|} \mathcal{B} \wedge \langle \mathbb{t} \rangle \langle \overline{3} \rangle \mathbf{tt} \\ &\mathcal{B} \equiv \langle b_1 \rangle \langle b_2 \rangle \langle \overline{b_1 b_2} \rangle \langle b_1 b_2 \rangle \langle \overline{b_2} \rangle \mathbf{tt} \end{aligned} \tag{3.8}$$

where b_1 represents the nucleotide read by the $\mathtt{RNA}_{\mathtt{pol}}$ subprocess, and $\overline{b_1 b_2}$ is the base pair provided as output. The $\mathcal{R}$ subprocess takes this base pair as input, and generates $\overline{b_2}$ as the (possibly) correct nucleotide that has to be added to the θ string. By definition, this means that:

$$\mathtt{RNA}_{\mathtt{pol}} \vDash \mathcal{T} \tag{3.9}$$

However, without losing generality, the length constraint can be relaxed through the use of recursion, obtaining the following specification:

$$\begin{aligned} &\mathtt{RNA}_{\mathtt{pol}} \vDash \mathcal{T}_r \\ &\mathcal{T}_r \equiv \langle b_1 \rangle \langle b_2 \rangle \langle \overline{b_1 b_2} \rangle \langle b_1 b_2 \rangle (\langle \overline{b_2} \rangle \mathbf{tt} \wedge \langle \overline{b_2} \rangle \mathcal{T}_r \wedge \langle \overline{b_2} \rangle \langle \mathbb{t} \rangle \langle \overline{3} \rangle \mathbf{tt}) \end{aligned} \tag{3.10}$$

Compared to the $\mathcal{T}$-like formulae, $\mathcal{T}_r$, as the similar formulae that will be defined for RNA processing and translation, is better suited to the case where the number of nucleotides to take into account are directly provided by a DNA sequence considered as a case study (as it will be clarified in Section 3.2.2).

Processing

The transcript can be an intermediary in the synthesis of a protein (in this case, it is called *mRNA*, standing for *messenger RNA*) or be itself the final product of the gene expression (that is, a functional RNA or non-coding RNA). However, before an RNA molecule can be considered *mature* (and hence carries out its purpose), it must undergo different *RNA processing* steps, depending on its type [92].

Two processing steps occur only on transcripts destined to become mRNA molecules:

- the *capping process*, in which an atypical guanine nucleotide (with a methyl group attached) is added to the $5'$ end of the RNA molecule;
- the *polyadenylation process*, which appends a *poly-A tail* (formed by a series of repeated adenine nucleotides) to the RNA's $3'$ end.

A third step, also common in various types of non-coding RNA, is called *RNA splicing* and removes the non-coding intervening sequence (*introns*) from the ribonucleotide chain of the transcript; as a result of this process, the transcript is converted into an uninterrupted sequence of coding portions of the gene (*exons*). In RNA molecules, an intron is identified by two starting and two ending nucleotides, *gu* and *ag*, respectively; however, not all *gu* and *ag* nucleotide motifs indicate an intron's starting and ending points. The actual splicing involves the complex molecular machinery called *spliceosome*, but, to capture how the informational content changes during the process, we can model its behaviour as the extraction of a set of label substrings from a string $\theta \in T$; each of them starts with the nucleotide string `"gu"` and ends with `"ag"`. The possibility that not all strings of this kind identify the boundaries of an intron is modelled, in the *processing* process $\mathcal{P}$, through non-deterministic choices.

Definition 3.4 (Intron and Exon)**.** Let

$$\begin{aligned} &I = \{\phi \mid \phi = \texttt{"gu"}\ \iota\ \texttt{"ag"},\ \phi \in T,\ \iota \in N^{+}_{rna},\ j_{min} \leq |\phi| \leq j_{max},\ j_{min}, j_{max} \in \mathbb{N}^{+}\} \text{ and} \\ &E = \{\zeta \mid \zeta \in T \backslash I,\ e_{min} \leq |\zeta| \leq e_{max},\ e_{min}, e_{max} \in \mathbb{N}^{+}\}, \end{aligned} \tag{3.11}$$

an *intron* is a string $\phi \in I$, while an *exon* is a string $\zeta \in E$.

Definition 3.5 (mRNA)**.** Given the set

$$R = \{\rho \mid \rho = \texttt{"c"}\ \zeta_1\ \ldots\ \zeta_k\ \texttt{"a"},\ \zeta_i \in E,\ i \in \{1,\ldots,k\}, r_{min} \leq k \leq r_{max},\ r_{min}, r_{max} \in \mathbb{N}\}, \tag{3.12}$$

an *mRNA* is a string $\rho \in R$.

r_{min}, r_{max} are, respectively, the minimum and maximum theoretical number of exons in the modelled mRNA molecule. The `c` label represents the *cap* of the mature mRNA, while the `a` label its *poly-A tail*.

Definition 3.6 (RNA Processing)**.** Given the set of labels $L_{\mathcal{P}} = N \cup \{3, 5, \mathtt{c}, \mathtt{a}\}$, for each string $\theta \in T$, the *processing* process $\mathcal{P}$, such that $\mathcal{T} \overset{\overline{\theta}}{\Rightarrow} \mathcal{P}$, is defined as:

$$
\begin{aligned}
&\mathcal{P} \overset{\text{def}}{=} 5.\mathcal{P}_{\mathtt{c}};\\
&\mathcal{P}_{\mathtt{c}} \overset{\text{def}}{=} \overline{\mathtt{c}}.\mathtt{S};\\
&\mathtt{S} \overset{\text{def}}{=} \mathtt{a}.\mathtt{A}_s + \mathtt{u}.\mathtt{U}_s + \mathtt{c}.\mathtt{C}_s + \mathtt{g}.\mathtt{G}_s + \mathtt{g}.\mathtt{I}_< + 3.\mathcal{P}_{\mathtt{a}};\\
&\mathtt{A}_s \overset{\text{def}}{=} \overline{\mathtt{a}}.\mathtt{S};\\
&\mathtt{U}_s \overset{\text{def}}{=} \overline{\mathtt{u}}.\mathtt{S};\\
&\mathtt{C}_s \overset{\text{def}}{=} \overline{\mathtt{c}}.\mathtt{S};\\
&\mathtt{G}_s \overset{\text{def}}{=} \overline{\mathtt{g}}.\mathtt{S};\\
&\mathtt{I}_< \overset{\text{def}}{=} \mathtt{u}.\mathcal{S};\\
&\mathcal{S} \overset{\text{def}}{=} \mathtt{a}.\mathcal{S} + \mathtt{a}.\mathtt{I}_> + \mathtt{u}.\mathcal{S} + \mathtt{c}.\mathcal{S} + \mathtt{g}.\mathcal{S};\\
&\mathtt{I}_> \overset{\text{def}}{=} \mathtt{g}.\mathtt{S};\\
&\mathcal{P}_{\mathtt{a}} \overset{\text{def}}{=} \overline{\mathtt{a}}.0;
\end{aligned}
\tag{3.13}
$$

where

- the process $\mathcal{P}_{\mathtt{c}}$ represents the addition of the *cap* ($\mathtt{c}$ label) to the $5'$ end (5 label) of the transcript; similarly, the process $\mathcal{P}_{\mathtt{a}}$ reproduces the polyadenylation of the RNA molecule.
- $\mathtt{S}$ is the process that models the *spliceosome* behaviour, which begins by looking for a g label that may start an intron (until it reaches the 3 label).
  - $\mathtt{A}_s$, $\mathtt{C}_s$, $\mathtt{G}_s$, and $\mathtt{U}_s$ are states through which the $\mathtt{S}$ process provides as output every nucleotide that is not the start of an intron;
  - $\mathtt{I}_<$ is a state that captures the case in which a g label followed by a u label represents the starting string of an intron ϕ; $\mathtt{I}_>$ determines that an a label followed by a g label signals the end of ϕ.
  - $\mathcal{S}$ is the process that reproduces the actual splicing by reading, one at a time, the nucleotides of an intron without producing them as output (i.e., by removing them from the sequence of the mRNA $\rho \in R$).

Given the set of labels $L'_{\mathcal{P}} = \{3, 5, b, \mathtt{c}\}$, with $b \in N_{rna}$, for each $\theta \in T$,

$$
\begin{aligned}
&\mathcal{P} \vDash \langle 5\rangle\langle\overline{\mathtt{c}}\rangle\mathcal{P}\\
&\mathcal{P} \equiv \bigwedge_{|\theta|} \mathcal{S} \wedge \langle 3\rangle\langle\overline{\mathtt{a}}\rangle\mathbf{tt}\\
&\mathcal{S} \equiv \bigwedge_{|\zeta|} \mathcal{E} \wedge \bigwedge_{|\phi|} \langle b\rangle\mathbf{tt}\\
&\mathcal{E} \equiv \langle b\rangle\langle\overline{b}\rangle\mathbf{tt}
\end{aligned}
\tag{3.14}
$$

where $\zeta \in E$ and $\phi \in I$.

In the first part of the $\mathcal{S}$ formula, the S (spliceosome) subprocess looks for an intron ϕ: the subformula $\mathcal{E}$ allows each nucleotide b that belongs to an exon ζ to be left unchanged (the label $\overline{b}$ produced as output). When the start of an intron ϕ is identified, each nucleotide read is not produced as an output; this behaviour continues for $|\phi|$ nucleotides. After that, the process looks for another exon up to the label 3. The main subprocess S, therefore, is such that:

$$\mathtt{S} \vDash \mathcal{S} \tag{3.15}$$

By relaxing the length constraints, it also satisfies the $\mathcal{S}_r$ formula, that is:

$$\begin{aligned} &\mathtt{S} \vDash \mathcal{S}_r \\ &\mathcal{S}_r \equiv \langle b \rangle \langle \overline{b} \rangle \mathcal{S}_r \wedge \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \mathcal{E} \\ &\mathcal{E} \equiv \langle b \rangle \mathcal{E} \wedge \langle \mathtt{a} \rangle (\langle \mathtt{g} \rangle \mathbf{tt} \wedge \langle \mathtt{g} \rangle \mathcal{S}_r) \end{aligned} \tag{3.16}$$

where $\{\mathtt{g}, \mathtt{u}\} \subset N_{rna}$. It is possible to note that not imposing a length on the string θ (and its introns and exons) forces the formula describing the S behaviour to explicitly take into account the recognition of the "gu" and "ag" substrings of θ.

Translation

The last step of the gene expression is the *translation* process, which converts the information contained in the nucleotide sequence of a mature mRNA into the amino acid sequence of a protein [61]. The set of the twenty *amino acid* labels is the following:

$$\begin{aligned} A = \{&\mathtt{ala}, \mathtt{asp}, \mathtt{arg}, \mathtt{asn}, \mathtt{cys}, \mathtt{gln}, \mathtt{glu}, \mathtt{gly}, \mathtt{his}, \mathtt{ile}, \\ &\mathtt{leu}, \mathtt{lys}, \mathtt{met}, \mathtt{phe}, \mathtt{pro}, \mathtt{ser}, \mathtt{thr}, \mathtt{trp}, \mathtt{tyr}, \mathtt{val}\} \end{aligned} \tag{3.17}$$

Definition 3.7 (Protein)**.** Let

$$P = \{\psi \mid \psi \in A^+,\ p_{min} \leq |\psi| \leq p_{max},\ p_{min}, p_{max} \in \mathbb{N}^+\}, \tag{3.18}$$

a *protein* is a string $\psi \in P$.

p_{min}, p_{max} are, respectively, the minimum and maximum theoretical length, in amino acids, of a protein ψ.

The translation is performed by a molecular complex called *ribosome*, which, based on the *genetic code* (see Table 3.1), associates one of the 20 amino acids with each triplet of nucleotides (called *codon*) of the mRNA sequence. Over the set N_{rna}, containing four nucleotides, we can obtain 64 triplets; hence each amino acid can be associated with more than one codon.

The *translation process* $\mathcal{L}$ begins from a *start codon* ("aug", which also codes for the *translation initiator methionine* imet) and terminates when the ribosome reaches one of the three possible *stop codons* ("uaa", "uag", and "uga" triplets).

The model represents the ribosome as the *process* R, which scans the string $\rho \in R$ one nucleotide at a time, starting from the *cap* (ℂ label) and looking for an "aug" codon. When it finds this starting substring, it begins to produce as output an amino acid for each codon it reads until it reaches a stop codon.

Definition 3.8 (Coding substring)**.** Given $\lambda, \lambda' \in N_{rna}^*$, $\rho \in R$, and

$$\begin{aligned} C = \{\xi \mid \rho = \texttt{"c"}\ \lambda\ \texttt{"aug"}\ \xi\ \sigma\ \lambda'\ \texttt{"a"},\ \sigma \in \{\texttt{"uaa"}, \texttt{"uag"}, \texttt{"uga"}\}, \\ \xi = \kappa_1\ \ldots\ \kappa_n,\ n \in \mathbb{N}^+,\ |\kappa| = 3,\ c_{min} \leq |\xi| \leq c_{max},\ c_{min}, c_{max} \in \mathbb{N}^+\}, \end{aligned} \tag{3.19}$$

a *coding substring* of ρ is a string $\xi \in C$.

c_{min}, c_{max} are, respectively, the minimum and maximum theoretical number of nucleotides of a coding string. The set definition also specifies that each coding string is made of triplets of nucleotides κ, representing its codons.

Definition 3.9 (Translation)**.** Given the set of labels $L_{\mathcal{L}} = N_{rna} \cup A \cup \{\mathtt{c}, \mathtt{imet}, \mathtt{s}\}$, for each string of labels $\rho \in R$, the *translation process* $\mathcal{L}$, such that $\mathcal{P} \overset{\overline{\rho}}{\Rightarrow} \mathcal{L}$, is defined as follows:

$$\begin{aligned}
&\mathcal{L} \overset{\text{def}}{=} \mathtt{c}.\mathrm{R};\\
&\mathrm{R} \overset{\text{def}}{=} \mathtt{u}.\mathrm{R} + \mathtt{c}.\mathrm{R} + \mathtt{a}.\mathrm{U}_{\triangleright} + \mathtt{g}.\mathrm{R};\\
&\mathrm{U}_{\triangleright} \overset{\text{def}}{=} \mathtt{u}.\mathrm{G}_{\triangleright} + \mathtt{c}.\mathrm{R} + \mathtt{a}.\mathrm{R} + \mathtt{g}.\mathrm{R};\\
&\mathrm{G}_{\triangleright} \overset{\text{def}}{=} \mathtt{u}.\mathrm{R} + \mathtt{c}.\mathrm{R} + \mathtt{a}.\mathrm{R} + \mathtt{g}.\mathcal{C}_{\triangleright};\\
&\mathcal{C}_{\triangleright} \overset{\text{def}}{=} \overline{\mathtt{imet}}.\mathcal{C};\\
&\mathcal{C} \overset{\text{def}}{=}\\
&\mathtt{u}.(\mathtt{u}.(\mathtt{u}.\mathrm{PHE} + \mathtt{c}.\mathrm{PHE} + \mathtt{a}.\mathrm{LEU} + \mathtt{g}.\mathrm{LEU}) + \mathtt{c}.(\mathtt{u}.\mathrm{SER} + \mathtt{c}.\mathrm{SER} + \mathtt{a}.\mathrm{SER} + \mathtt{g}.\mathrm{SER}) +\\
&\mathtt{a}.(\mathtt{u}.\mathrm{TYR} + \mathtt{c}.\mathrm{TYR} + \mathtt{a}.\mathcal{C}_{\square} + \mathtt{g}.\mathcal{C}_{\square}) + \mathtt{g}.(\mathtt{u}.\mathrm{CYS} + \mathtt{c}.\mathrm{CYS} + \mathtt{a}.\mathcal{C}_{\square} + \mathtt{g}.\mathrm{TRP})) +\\
&\mathtt{c}.(\mathtt{u}.(\mathtt{u}.\mathrm{LEU} + \mathtt{c}.\mathrm{LEU} + \mathtt{a}.\mathrm{LEU} + \mathtt{g}.\mathrm{LEU}) + \mathtt{c}.(\mathtt{u}.\mathrm{PRO} + \mathtt{c}.\mathrm{PRO} + \mathtt{a}.\mathrm{PRO} + \mathtt{g}.\mathrm{PRO}) +\\
&\mathtt{a}.(\mathtt{u}.\mathrm{HIS} + \mathtt{c}.\mathrm{HIS} + \mathtt{a}.\mathrm{GLN} + \mathtt{g}.\mathrm{GLN}) + \mathtt{g}.(\mathtt{u}.\mathrm{ARG} + \mathtt{c}.\mathrm{ARG} + \mathtt{a}.\mathrm{ARG} + \mathtt{g}.\mathrm{ARG})) +\\
&\mathtt{a}.(\mathtt{u}.(\mathtt{u}.\mathrm{ILE} + \mathtt{c}.\mathrm{ILE} + \mathtt{a}.\mathrm{ILE} + \mathtt{g}.\mathrm{MET}) + \mathtt{c}.(\mathtt{u}.\mathrm{THR} + \mathtt{c}.\mathrm{THR} + \mathtt{a}.\mathrm{THR} + \mathtt{g}.\mathrm{THR}) +\\
&\mathtt{a}.(\mathtt{u}.\mathrm{ASN} + \mathtt{c}.\mathrm{ASN} + \mathtt{a}.\mathrm{LYS} + \mathtt{g}.\mathrm{LYS}) + \mathtt{g}.(\mathtt{u}.\mathrm{SER} + \mathtt{c}.\mathrm{SER} + \mathtt{a}.\mathrm{ARG} + \mathtt{g}.\mathrm{ARG})) +\\
&\mathtt{g}.(\mathtt{u}.(\mathtt{u}.\mathrm{VAL} + \mathtt{c}.\mathrm{VAL} + \mathtt{a}.\mathrm{VAL} + \mathtt{g}.\mathrm{VAL}) + \mathtt{c}.(\mathtt{u}.\mathrm{ALA} + \mathtt{c}.\mathrm{ALA} + \mathtt{a}.\mathrm{ALA} + \mathtt{g}.\mathrm{ALA}) +\\
&\mathtt{a}.(\mathtt{u}.\mathrm{ASP} + \mathtt{c}.\mathrm{ASP} + \mathtt{a}.\mathrm{GLU} + \mathtt{g}.\mathrm{GLU}) + \mathtt{g}.(\mathtt{u}.\mathrm{GLY} + \mathtt{c}.\mathrm{GLY} + \mathtt{a}.\mathrm{GLY} + \mathtt{g}.\mathrm{GLY})) +\\
&\mathtt{a}.\mathcal{C}_{\square};\\
&\mathrm{ALA} \overset{\text{def}}{=} \overline{\mathtt{ala}}.\mathcal{C};\ \mathrm{ARG} \overset{\text{def}}{=} \overline{\mathtt{arg}}.\mathcal{C};\\
&\mathrm{ASN} \overset{\text{def}}{=} \overline{\mathtt{asn}}.\mathcal{C};\ \mathrm{ASP} \overset{\text{def}}{=} \overline{\mathtt{asp}}.\mathcal{C};\\
&\mathrm{CYS} \overset{\text{def}}{=} \overline{\mathtt{cys}}.\mathcal{C};\ \mathrm{GLY} \overset{\text{def}}{=} \overline{\mathtt{gly}}.\mathcal{C};\\
&\mathrm{GLU} \overset{\text{def}}{=} \overline{\mathtt{glu}}.\mathcal{C};\ \mathrm{GLN} \overset{\text{def}}{=} \overline{\mathtt{gln}}.\mathcal{C};\\
&\mathrm{HIS} \overset{\text{def}}{=} \overline{\mathtt{his}}.\mathcal{C};\ \mathrm{ILE} \overset{\text{def}}{=} \overline{\mathtt{ile}}.\mathcal{C};\\
&\mathrm{LEU} \overset{\text{def}}{=} \overline{\mathtt{leu}}.\mathcal{C};\ \mathrm{LYS} \overset{\text{def}}{=} \overline{\mathtt{lys}}.\mathcal{C};\\
&\mathrm{MET} \overset{\text{def}}{=} \overline{\mathtt{met}}.\mathcal{C};\ \mathrm{PHE} \overset{\text{def}}{=} \overline{\mathtt{phe}}.\mathcal{C};\\
&\mathrm{PRO} \overset{\text{def}}{=} \overline{\mathtt{pro}}.\mathcal{C};\ \mathrm{SER} \overset{\text{def}}{=} \overline{\mathtt{ser}}.\mathcal{C};\\
&\mathrm{THR} \overset{\text{def}}{=} \overline{\mathtt{thr}}.\mathcal{C};\ \mathrm{TRP} \overset{\text{def}}{=} \overline{\mathtt{trp}}.\mathcal{C};\\
&\mathrm{TYR} \overset{\text{def}}{=} \overline{\mathtt{tyr}}.\mathcal{C};\ \mathrm{VAL} \overset{\text{def}}{=} \overline{\mathtt{val}}.\mathcal{C};\\
&\mathcal{C}_{\square} \overset{\text{def}}{=} \overline{\mathtt{s}}.0;
\end{aligned} \tag{3.20}$$

where

- R is the process that models the behaviour of the *ribosome*; it starts by looking for the first a of a start codon and proceeds via the following subprocesses/states.
  - The states $\mathtt{U}_{\triangleright}$ and $\mathtt{G}_{\triangleright}$ allow to identify a *start codon* ("aug") and then initiate the actual translation; the first step of this process is performed by the $\mathcal{C}_{\triangleright}$ process (where $\mathcal{C}$ stands for "code"), in which the *translation initiator methionine* imet, associated with the start codon, is produced as output.
  - The $\mathcal{C}$ process performs the actual translation of each codon to the related amino acid until it reaches a *stop codon* ("uaa","uag","uga" substrings) or the *poly-A tail* (a label). Each stop codon leads to the $\mathcal{C}_{\square}$ process, which ends the translation and produces as output the *stop label* s.

Table 3.1 – Genetic code table. It represents the association of each of the 20 amino acid labels (plus the stop label s) with the related group of matching codon strings.

"aga"									"uua"					"agc"					
"agg"									"uug"					"agu"					
"gca"	"cga"						"gga"		"cua"				"cca"	"uca"	"aca"			"gua"	
"gcc"	"cgc"						"ggc"	"aua"	"cuc"				"ccc"	"ucc"	"acc"			"guc"	"uaa"
"gcg"	"cgg"	"gac"	"aac"	"ugc"	"gaa"	"caa"	"ggg"	"cac"	"auc"	"cug"	"aaa"		"uuc"	"ccg"	"ucg"	"acg"	"uac"	"gug"	"uag"
"gcu"	"cgu"	"gau"	"aau"	"ugu"	"gag"	"cag"	"ggu"	"cau"	"auu"	"cuu"	"aag"	"aug"	"uuu"	"ccu"	"ucu"	"acu"	"ugg"	"uau"	"guu"	"uga"
ala	arg	asp	asn	cys	glu	gln	gly	his	ile	leu	lys	met	phe	pro	ser	thr	trp	tyr	val	s

The model contemplates the possibility that the ribosome finds no stop codons and reaches the 3′ end of the mRNA molecule, signalled by the *poly-A tail* sequence [40]; however, handling this event is beyond this model's scope.

Although recognising the last nucleotide of a stop codon could immediately interrupt the translation process (through the *nil* process 0), we define the $\mathcal{C}_{\square}$ subprocess to make the model clearer and easier to read.

Let ξ be a coding substring of $\rho \in R$, $L'_{\mathcal{L}} = \{b, a, \mathtt{c}, \mathtt{imet}, \mathtt{a}, \mathtt{u}, \mathtt{g}\}$, where $b \in N_{rna}$, and $a \in A$; for each $\rho \in R$,

$$\begin{aligned}
&\mathcal{L} \vDash \langle \mathtt{c} \rangle \mathcal{L} \\
&\mathcal{L} \equiv \langle b \rangle \mathcal{L} \wedge \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{imet}} \rangle \mathcal{C} \\
&\mathcal{C} \equiv \bigwedge_{\frac{|\xi|}{3}} \mathcal{A} \\
&\mathcal{A} \equiv \langle b_1 \rangle \langle b_2 \rangle \langle b_3 \rangle \langle \overline{a} \rangle \mathbf{tt}
\end{aligned} \tag{3.21}$$

where b represents the base of each nucleotide read before the R subprocess reaches the "aug" codon; after that, the formula $\mathcal{C}$ describes how the process translates each codon it encounters

until it generates a protein of $\frac{|\xi|}{3}$ amino acids. The labels b_1, b_2, and b_3 represent the three bases of a codon, while a is the corresponding amino acid in the set A. By definition,

$$\mathtt{R} \vDash \mathcal{L} \tag{3.22}$$

and, if we relax the length constraint:

$$\begin{aligned} &\mathtt{R} \vDash \mathcal{L}_r \\ &\mathcal{L}_r \equiv \langle b \rangle \mathcal{L}_r \wedge \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{imet}} \rangle \mathcal{C} \\ &\mathcal{C} \equiv \langle b_1 \rangle \langle b_2 \rangle \langle b_3 \rangle (\langle \overline{a} \rangle \mathbf{tt} \wedge \langle \overline{a} \rangle \mathcal{C}) \wedge \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt} \\ &\qquad \wedge \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt} \wedge \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt} \end{aligned} \tag{3.23}$$

Similarly to the formula that specifies the S process, if we do not impose a specific length for the coding string ξ, the resulting formula must explicitly check when the process is required to terminate (in this case, by identifying one of the three stop codons).

We can summarise the model of gene expression through the process $\mathcal{G}$, defined as a cycle that synthesises multiple proteins starting from the same gene $\gamma \in G$. Due to alternative splicing, a single gene can code for several slightly different protein variants [17].

Definition 3.10 (Gene expression)**.** Given the processes $\mathcal{T}, \mathcal{P}, \mathcal{L}$, as defined above, and the label strings $\gamma \in G$, $\theta \in T$, $\rho \in R$, and $\psi \in P$, we define *gene expression* the process $\mathcal{G}$ such that

$$\mathcal{G} \overset{\overline{\gamma}}{\Rightarrow} \mathcal{T} \overset{\overline{\theta}}{\Rightarrow} \mathcal{P} \overset{\overline{\rho}}{\Rightarrow} \mathcal{L} \overset{\overline{\psi}}{\Rightarrow} \mathcal{G} \tag{3.24}$$

Maintaining the approach adopted up to this point, the model of the $\mathcal{G}$ process is strictly focused on the informational content of a gene and considers other aspects, such as the timing associated with the expression of protein variants, as negligible.

From the above models, it can be guessed that modifying a single nucleotide of a gene can change the corresponding codon and, in turn, the amino acid produced in the translation process (missense mutation); in the remainder of this chapter, we will represent how a mutated codon can code for an amino acid that affects the structure of the related protein.

3.2.2 Formal description of HBB gene expression

The process definitions provided in the previous subsection can be applied to analyse, from a theoretical perspective, the effects of a point mutation on the expression of RNAs and proteins. As a case study, we take the *HBB gene*, which codes for one of the β subunits of the haemoglobin molecule. Haemoglobin is an essential protein of erythrocytes (red blood cells), formed by four subunits called *globins*; precisely, they are two α-globins and two β-globins, allowing haemoglobin to bind four oxygen molecules [70].

Before considering the gene mutation, we define a model of its correct expression; it is based on the HBB gene's DNA sequence derived from an HBB transcript variant (1742 nucleotides) [82], which we retrieved from the National Center for Biotechnology Information (NCBI) AceView

Table 3.2 – Meaning and transliteration of the symbols representing the processes\states used in Definitions 3.3, 3.6, and 3.9.

Process\State	Transliteration	Description
$\mathtt{A}$	A	adenine states
$\mathtt{ALA}$	ALA	alanine
$\mathtt{ARG}$	ARG	arginine
$\mathtt{ASN}$	ASN	asparagine
$\mathtt{ASP}$	ASP	aspartic acid
$\mathtt{C}$	C	cytosine states
$\mathcal{P}_{\mathtt{c}}$	CAPPING	capping
$\mathtt{CYS}$	CYS	cysteine
$\mathtt{G}$	G	guanine states
$\mathtt{GLY}$	GLY	gycine
$\mathtt{GLU}$	GLU	glutamic acid
$\mathtt{GLN}$	GLN	glutamine
$\mathtt{HIS}$	HIS	histidine
$\mathtt{ILE}$	ILE	isoleucine
$\mathtt{I}_<$	INTRONSTART	intron starting state
$\mathtt{I}_>$	INTRONEND	intron ending state
$\mathtt{LEU}$	LEU	leucine
$\mathtt{LYS}$	LYS	lysine
$\mathtt{MET}$	MET	methionine
$\mathtt{PHE}$	PHE	phenylalanine
$\mathcal{P}_{\mathtt{a}}$	POLYAD	polyadenylation
$\mathtt{PRO}$	PRO	proline
$\mathcal{P}$	PROCESSING	processing
$\mathcal{R}$	PROOFREAD	proofreading
$\mathtt{R}$	RIBOSOME	ribosome
$\mathtt{R}_{\mathcal{J}^h}$	HIRIBOSOME	modified version of $\mathtt{R}$ (Eq. 3.36)
$\mathtt{RNA}_{\mathtt{pol}}$	RNAPOL	RNA polymerase
$\mathtt{SER}$	SER	serine
$\mathcal{S}$	SPLICE	intron removal
$\mathtt{S}$	SPLICEOSOME	spliceosome
$\mathtt{U}_\triangleright$	STARTCODON1	start-codon's $\mathtt{a}$ found (looking for $\mathtt{u}$)
$\mathtt{G}_\triangleright$	STARTCODON2	start-codon's $\mathtt{u}$ found (looking for $\mathtt{g}$)
$\mathtt{T}$	T	thymine states
$\mathtt{THR}$	THR	threonine
$\mathcal{T}$	TRANSCRIPTION	transcription
$\mathcal{C}$	TRANSLATE	codon - amino acid association
$\mathcal{L}$	TRANSLATION	translation
$\mathcal{C}_\triangleright$	TSTART	start-codon's $\mathtt{g}$ found
$\mathcal{C}_\square$	TSTOP	end of translation
$\mathtt{TRP}$	TRP	tryptophan
$\mathtt{TYR}$	TYR	tyrosine
$\mathtt{U}$	U	uracil states
$\mathtt{VAL}$	VAL	valine

Table 3.3 – Description of the action labels in Definitions 3.3, 3.6, and 3.9.

Action label	Description
a	adenine
ala	alanine
arg	arginine
asn	asparagine
asp	aspartic acid
c	cytosine
ȼ	transcript cap
cys	cysteine
5	RNA 5' end
g	guanine
gly	glycine
glu	glutamic acid
gln	glutamine
his	histidine
ile	isoleucine
imet	translation initiator methionine
leu	leucine
lys	lysine
met	methionine
phe	phenylalanine
pro	proline
ℙ	transcription promoter
ɑ	poy-A tail
ser	serine
ȿ	end of translation
t	tymine
ŧ	transcription terminator
thr	threonine
3	RNA 3' end
trp	tryptophan
tyr	tyrosine
u	uracil
val	valine

website [115]. Due to the theoretical setting of our approach, we choose this sequence for no other reason but to show how our model can represent the expression of an actual gene.

In what follows, we colour the exon coding regions in green, the introns in blue, while in red we highlight the codon that codes for the sixth glutamic acid (Glu 6) of the amino acid sequence produced by the HBB gene. This amino acid is the one affected by the mutation that we will model in the next subsection.

The HBB gene sequence is provided in Appendix B, along with its complete gene expression model. The adopted approach consists of applying the "relaxed" formulae introduced in the previous subsection to the HBB nucleotide sequence. As an example, we show the formula based on Equation 3.10 that is satisfied by the $\mathcal{T}$ process.

Example 3.1. (Transcription of the HBB gene) By extending Equation 3.10 to the whole transcription process $\mathcal{T}$, we obtain that:

$$\begin{aligned} &\mathcal{T} \vDash \langle \mathtt{p} \rangle \langle \overline{5} \rangle \mathcal{T}_r \\ &\mathcal{T}_r \equiv \langle b_1 \rangle \langle b_2 \rangle \langle \overline{b_1 b_2} \rangle \langle b_1 b_2 \rangle (\langle \overline{b_2} \rangle \mathbf{tt} \wedge \langle \overline{b_2} \rangle \mathcal{T}_r \wedge \langle \overline{b_2} \rangle \langle \mathtt{t} \rangle \langle \overline{3} \rangle \mathbf{tt}) \end{aligned} \tag{3.25}$$

We apply Equation 3.25 to formally describe the HBB gene transcription on the basis of its DNA sequence; the latter is represented as the γ_{hbb} string of Appendix Equation B.1.

$$\begin{aligned} &\mathcal{T} \vDash \\ &\langle \mathtt{p} \rangle \langle \overline{5} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \\ &\langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \\ &\vdots \\ &\langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \ \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \ \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \\ &\vdots \\ &\langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \\ &\langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{t} \rangle \langle \overline{3} \rangle \mathbf{tt}; \end{aligned} \tag{3.26}$$

The result of $\mathcal{T}$ on $\delta_{hbb} = \texttt{"p"}\ \gamma_{hbb}\ \texttt{"t"}$ is the string $\chi_{hbb} = \texttt{"5"}\ \theta_{hbb}\ \texttt{"3"}$, where θ_{hbb} is the *transcript* string of Appendix Equation B.4.

3.2.3 Formal description of the Glu6Val mutation

Starting from the model of the correct HBB gene expression, it is possible to formalise how a point mutation can go through each of its steps by evading error detection.

We propose here a model for the Glu6Val mutation, which causes sickle-cell anaemia; a haemoglobin molecule with this mutation is referred to as HbS. In HbS disease, a point-mutation in the β-globin gene produces a subunit in which the Glu 6 is changed to valine (Val 6). Such a mutation creates a hydrophobic patch on the surface of the haemoglobin molecule that fits into

a hydrophobic pocket of another one and forms fibrous precipitates; this process produces the characteristic sickle shape of the affected red blood cells [19].

Since this pathology is hereditary, the mutation is already present in the DNA sequence and thus is treated by the cell as correct information. However, we chose this specific mutation because the aim of our analysis is not simply to describe the behaviour of the expression of a mutation but to formally demonstrate, via the CCS models and the related HML formulae, how it differently affects the folding process of proteins and RNAs.

To maintain the readability of the formulae, we base the subsequent description on the string γ'_{hbs}, containing the first exon coding region (coloured in green) and the first intron (coloured in blue) of the mutant HBB gene γ_{hbs}; similarly to what shown in Appendix B, our approach can be effectively extended to the entire gene sequence.

$$\gamma'_{hbs} = \texttt{"taccacgtagactgaggacacctcttcagacggcaatgacgggacaccccgttccacttgcacc}\\ \texttt{tacttcaaccaccactccgggacccgtccaaccatagttccaatgttctgtccaaattcctctggttatct}\\ \texttt{ttgacccgtacacctctgtctcttctgagaacccaaagactatccgtgactgagagagacggataaccaga}\\ \texttt{taaaagggtgggaatc"} \tag{3.27}$$

The mutated nucleotide (from t to a) is underlined in the above string and in the following formulae.

As a first step, we show how the sub-formulae of the whole HbS formula, describing the Glu6Val mutation, are satisfied by the main processes of gene expression.

Transcription

$$\begin{aligned}
&\texttt{RNA}_{\texttt{pol}} \models\\
&\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\\
&\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\\
&\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\\
&\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\underline{\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle}\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\\
&\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\\
&\vdots\\
&\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\\
&\vdots\\
&\langle \texttt{g}\rangle\langle \overline{gc}\rangle\langle gc\rangle\langle \texttt{c}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{a}\rangle\langle \overline{au}\rangle\langle au\rangle\langle \texttt{u}\rangle\langle \texttt{t}\rangle\langle \overline{ta}\rangle\langle ta\rangle\langle \texttt{a}\rangle\langle \texttt{c}\rangle\langle \overline{cg}\rangle\langle cg\rangle\langle \texttt{g}\rangle\mathbf{tt}
\end{aligned} \tag{3.28}$$

The $\texttt{RNA}_{\texttt{pol}}$ process converts the incorrect adenine (a), of the mutated codon in the DNA strand, to uracil (u) in the nucleotide sequence of the transcript θ_{hbs}. As a result, the substring θ'_{hbs},

containing the first exon coding region and the first intron of θ_{hbs}, appears as follows:

$$\theta'_{hbs} = \texttt{"auggugcaucugacuccuguggagaagucugccguuacugcccuguggggcaaggugaacgugg augaaguuggugguga ggcccugggcaggugguaucaagguuacaagacagguuuaaggagaccaauaga aacugggcauguggagacagaagacucuuggguuucugauaggcacugacucucucugccuauuggucu auuuucccacccuuag"} \tag{3.29}$$

We highlight in orange the starting and ending substrings of the intron (`"gu"` and `"ag"`).

Processing

$$\mathtt{S} \vDash$$
$$\langle a\rangle\langle \overline{a}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle\langle a\rangle\langle \overline{a}\rangle\langle u\rangle\langle \overline{u}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle$$
$$\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle$$
$$\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle u\rangle\langle \overline{u}\rangle\langle a\rangle\langle \overline{a}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle$$
$$\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle\langle a\rangle\langle \overline{a}\rangle\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle a\rangle\langle \overline{a}\rangle\langle c\rangle\langle \overline{c}\rangle$$
$$\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle$$
$$\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle c\rangle\langle \overline{c}\rangle\langle u\rangle\langle \overline{u}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle \overline{g}\rangle\langle c\rangle\langle \overline{c}\rangle$$
$$\langle a\rangle\langle \overline{a}\rangle\langle g\rangle\langle \overline{g}\rangle\langle g\rangle\langle u\rangle\langle u\rangle\langle g\rangle\langle g\rangle\langle u\rangle\langle a\rangle\langle u\rangle\langle c\rangle\langle a\rangle\langle a\rangle\langle g\rangle\langle g\rangle\langle u\rangle\langle u\rangle\langle a\rangle\langle c\rangle\langle a\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle c\rangle\langle a\rangle\langle g\rangle\langle g\rangle\langle u\rangle$$
$$\langle u\rangle\langle u\rangle\langle a\rangle\langle a\rangle\langle g\rangle\langle g\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle c\rangle\langle c\rangle\langle a\rangle\langle a\rangle\langle u\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle a\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle g\rangle\langle g\rangle\langle g\rangle\langle c\rangle\langle a\rangle\langle u\rangle\langle g\rangle\langle u\rangle\langle g\rangle$$
$$\langle g\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle c\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle a\rangle\langle g\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle u\rangle\langle g\rangle\langle g\rangle\langle g\rangle\langle u\rangle\langle u\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle g\rangle\langle a\rangle\langle u\rangle\langle a\rangle$$
$$\langle g\rangle\langle g\rangle\langle c\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle g\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle g\rangle\langle c\rangle\langle c\rangle\langle u\rangle\langle a\rangle\langle u\rangle\langle u\rangle\langle g\rangle\langle g\rangle\langle u\rangle\langle c\rangle\langle u\rangle\langle a\rangle\langle u\rangle$$
$$\langle u\rangle\langle u\rangle\langle u\rangle\langle c\rangle\langle c\rangle\langle c\rangle\langle a\rangle\langle c\rangle\langle c\rangle\langle c\rangle\langle u\rangle\langle u\rangle\langle a\rangle\langle g\rangle\mathbf{tt} \tag{3.30}$$

The mutated nucleotide u is part of an exon; therefore, it is not removed from the RNA sequence by the spliceosome process S, resulting in the following ρ'_{hbs} substring of the mature mRNA ρ_{hbs}:

$$\rho'_{hbs} = \texttt{"auggugcaucugacuccuguggagaagucugccguuacugcccuguggggcaaggugaacgugg augaaguuggugguga ggcccugggcag"} \tag{3.31}$$

The start codon `"aug"` is represented in orange. To avoid dealing with excessively long formulae, in what follows, we consider the substring ρ''_{hbs} made of the first thirteen codons of ρ'_{hbs} (start codon included), that is, those substring triplets that are close to the mutated `"gug"` codon:

$$\rho''_{hbs} = \texttt{"auggugcaucugacuccuguggagaagucugccguuacu"} \tag{3.32}$$

Translation

$$\mathtt{R} \vDash$$
$$\langle a\rangle\langle u\rangle\langle g\rangle\langle \overline{\mathtt{imet}}\rangle\langle g\rangle\langle u\rangle\langle g\rangle\langle \overline{\mathtt{val}}\rangle\langle c\rangle\langle a\rangle\langle u\rangle\langle \overline{\mathtt{his}}\rangle\langle c\rangle\langle u\rangle\langle g\rangle\langle \overline{\mathtt{leu}}\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle \overline{\mathtt{thr}}\rangle\langle c\rangle\langle c\rangle\langle u\rangle$$
$$\langle \overline{\mathtt{pro}}\rangle\langle g\rangle\langle u\rangle\langle g\rangle\langle \overline{\mathtt{val}}\rangle\langle g\rangle\langle a\rangle\langle g\rangle\langle \overline{\mathtt{glu}}\rangle\langle a\rangle\langle a\rangle\langle g\rangle\langle \overline{\mathtt{lys}}\rangle\langle t\rangle\langle c\rangle\langle t\rangle\langle \overline{\mathtt{ser}}\rangle\langle g\rangle\langle c\rangle\langle c\rangle\langle \overline{\mathtt{ala}}\rangle\langle g\rangle\langle u\rangle$$
$$\langle u\rangle\langle \overline{\mathtt{val}}\rangle\langle a\rangle\langle c\rangle\langle u\rangle\langle \overline{\mathtt{thr}}\rangle\mathbf{tt} \tag{3.33}$$

The Glu6Val acts as a missense mutation, converting the "gag" codon that codes for glutamic acid (glu) to the "gug" codon, which instead codes for valine (val).

The amino acid sequence generated by the portion of the mutated HBB gene analysed in this section is represented through the substring ψ'_{hbs} of the protein ψ_{hbs}:

$$\psi'_{hbs} = \text{"imet val his leu thr pro val glu lys ser ala val thr"} \tag{3.34}$$

The translation initiator methionine is coloured in orange to indicate that it should be removed to produce the mature protein.

Although intuitively understandable, we show that the R process satisfies equally the translation of the normal and mutated genes, that is:

$$\begin{aligned}
&\mathtt{R} \models \\
&\langle a\rangle\langle u\rangle\langle g\rangle\overline{\langle imet\rangle}\langle g\rangle\langle u\rangle\langle g\rangle\overline{\langle val\rangle}\langle c\rangle\langle a\rangle\langle u\rangle\overline{\langle his\rangle}\langle c\rangle\langle u\rangle\langle g\rangle\overline{\langle leu\rangle}\langle a\rangle\langle c\rangle\langle u\rangle\overline{\langle thr\rangle}\langle c\rangle\langle c\rangle\langle u\rangle \\
&\overline{\langle pro\rangle}\langle g\rangle\langle a\rangle\langle g\rangle\overline{\langle glu\rangle}\langle g\rangle\langle a\rangle\langle g\rangle\overline{\langle glu\rangle}\langle a\rangle\langle a\rangle\langle g\rangle\overline{\langle lys\rangle}\langle u\rangle\langle c\rangle\langle u\rangle\overline{\langle ser\rangle}\langle g\rangle\langle c\rangle\langle c\rangle\overline{\langle ala\rangle}\langle g\rangle\langle u\rangle \\
&\langle u\rangle\overline{\langle val\rangle}\langle a\rangle\langle c\rangle\langle u\rangle\overline{\langle thr\rangle}\mathbf{tt} \\
&\wedge \\
&\langle a\rangle\langle u\rangle\langle g\rangle\overline{\langle imet\rangle}\langle g\rangle\langle u\rangle\langle g\rangle\overline{\langle val\rangle}\langle c\rangle\langle a\rangle\langle u\rangle\overline{\langle his\rangle}\langle c\rangle\langle u\rangle\langle g\rangle\overline{\langle leu\rangle}\langle a\rangle\langle c\rangle\langle u\rangle\overline{\langle thr\rangle}\langle c\rangle\langle c\rangle\langle u\rangle \\
&\overline{\langle pro\rangle}\langle g\rangle\langle u\rangle\langle g\rangle\overline{\langle val\rangle}\langle g\rangle\langle a\rangle\langle g\rangle\overline{\langle glu\rangle}\langle a\rangle\langle a\rangle\langle g\rangle\overline{\langle lys\rangle}\langle u\rangle\langle c\rangle\langle u\rangle\overline{\langle ser\rangle}\langle g\rangle\langle c\rangle\langle c\rangle\overline{\langle ala\rangle}\langle g\rangle\langle u\rangle \\
&\langle u\rangle\overline{\langle val\rangle}\langle a\rangle\langle c\rangle\langle u\rangle\overline{\langle thr\rangle}\mathbf{tt}
\end{aligned} \tag{3.35}$$

The verification that all the above-described formulae are satisfied by the related processes has been performed with the model checking function of the CAAL concurrency workbench [3]. The results are shown in Figure 3.1 and prove that the provided models of gene expression can satisfy both the formulae for the synthesis of the normal β-globin molecule and those of the HbS mutation.

Status	Time	Property	Verify
✔	101 ms	RNAPOL ⊨ <t><'ta><ta><'a><a><'au><au><'u><c><'cg><cg><'g><c><'cg><cg><'g><a> <'au><au><'u><c><'cg><cg><'g><g><'gc><gc><'c><t><'ta><ta><'a><a><'au><au> <'u><g><'gc><gc><'c><a><'au><au><'u><c><'cg><cg><'g><t><'ta><ta><'a><g><'gc> <gc><'c><a><'au><au><'u><g><'gc><gc><'c><g><'gc><gc><'c><a><'au><au><'u> <c> <'cg><cg><'g> <a><'au><au> <'u><c><'cg><cg><'g> <c><'cg><cg><'g><t><'ta><ta> <'a><c><'cg><cg><'g><t><'ta><ta><'a><t><'ta><ta><'a><c><'cg><cg><'g><a><'au> <au><'u><g><'gc><gc><'c><a><'au><au><'u><c><'cg><cg><'g><g><'gc><gc><'c><g> <'gc><gc><'c><c><'cg><cg><'g><a><'au><au><'u><a><'au><au><'u><t><'ta><ta> <'a><g><'gc><gc><'c><a><'au><au><'u><c><'cg><cg><'g><g><'gc><gc><'c><g><'gc> <gc><'c><g><'gc><gc><'c><a><'au><au><'u><c><'cg><cg><'g><a><'au><au><'u><c> <'cg><cg><'g><c><'cg><cg><'g><c><'cg><cg><'g><c><'cg><cg><'g><g><'gc><gc> <'c><t><'ta><ta><'a><t><'ta><ta><'a><c><'cg><cg><'g><c><'cg><cg><'g><a><'au> <au><'u><c><'cg><cg><'g><t><'ta><ta><'a><t><'ta><ta><'a><g><'gc><gc><'c><c> <'cg><cg><'g><a><'au><au><'u><c><'cg><cg><'g><c><'cg><cg><'g><t><'ta><ta> <'a><a><'au><au><'u><c><'cg><cg><'g><t><'ta><ta><'a><t><'ta><ta><'a><c><'cg> <cg><'g><a><'au><au><'u><a><'au><au><'u><c><'cg><cg><'g><c><'cg><cg><'g><a> <'au><au><'u><c><'cg><cg><'g><c><'cg><cg><'g><a><'au><au><'u><c><'cg><cg> <'g><t><'ta><ta><'a><c><'cg><cg><'g><c><'cg><cg><'g><g><'gc><gc><'c><g><'gc> <gc><'c><g><'gc><gc><'c><a><'au><au><'u><c><'cg><cg><'g><c><'cg><cg><'g><c> <'cg><cg><'g><g><'gc><gc><'c><t><'ta><ta><'a><c><'cg><cg><'g>tt	▶
✔	101 ms	SPLICEOSOME ⊨ <a><'a><u><'u><g><'g><g><'g><u><'u><g><'g><c><'c><a><'a><u> <'u><c><'c><u><'u><g><'g><a><'a><c><'c><u><'u><c><'c><c><'c><u><'u> <g><'g> <u><'u> <g><'g><g> <'g><a><'a><g><'g><a><'a><a><'a><g><'g><u><'u><c><'c><u> <'u><g><'g><c><'c><c><'c><g><'g><u><'u><u><'u><a><'a><c><'c><u><'u><g><'g> <c><'c><c><'c><c><'c><u><'u><g><'g><u><'u><g><'g><g><'g><g><'g><g><'g><c> <'c><a><'a><a><'a><g><'g><g><'g><u><'u><g><'g><a><'a><a><'a><c><'c><g><'g> <u><'u><g><'g><g><'g><a><'a><u><'u><g><'g><a><'a><a><'a><g><'g><u><'u><u> <'u><g><'g><g><'g><u><'u><g><'g><g><'g><u><'u><g><'g><a><'a><g><'g><g><'g> <c><'c><c><'c><c><'c><u><'u><g><'g><g><'g><g><'g><c><'c><a><'a><g><'g> <g> <u><u><g><g><u><a><u><c><a><a><g><g><u><u><a><c><a><a><g><a><c><a><g><g><u> <u><u><a><a><g><g><a><g><a><c><c><a><a><u><a><g><a><a><a><c><u><g><g><g><c> <a><u><g><u><g><g><a><g><a><c><a><g><a><g><a><a><g><a><c><u><c><u><u><g><g> <g><u><u><u><c><u><g><a><u><a><g><g><c><a><c><u><g><a><c><u><c><u><c><u><c> <u><g><c><c><u><a><u><u><g><g><u><c><u><a><u><u><u><u><c><c><c><a><c><c><c> <u><u><a><g>tt	▶
✔	51 ms	RIBOSOME ⊨ <a><u><g><'imet> <g><u><g><'val><c><a><u><'his><c><u><g><'leu><a> <c><u><'thr><c><c><u><'pro><g><u><g><'val><g><a><g><'glu><a><a><g><'lys><u> <c><u><'ser><g><c><c><'ala><g><u><u><'val><a><c><u><'thr>tt	▶
✔	51 ms	RIBOSOME ⊨ <a><u><g><'imet> <g><u><g><'val><c><a><u><'his><c><u><g><'leu><a> <c><u><'thr><c><c><u><'pro><g><a><g><'glu><g><a><g><'glu><a><a><g><'lys><u> <c><u><'ser><g><c><c><'ala><g><u><u><'val><a><c><u><'thr>tt and <a><u><g> <'imet> <g><u><g><'val><c><a><u><'his><c><u><g><'leu><a><c><u><'thr><c><c> <u><'pro><g><u><g><'val><g><a><g><'glu><a><a><g><'lys><u><c><u><'ser><g><c> <c><'ala><g><u><u><'val><a><c><u><'thr>tt	▶

Figure 3.1 – Verification performed via the CAAL web-based tool of the specifications provided for the Glu6Val expression. The $\mathtt{RNA_{pol}}$, S, and R processes are transliterated as RNAPOL, SPLICEOSOME and RIBOSOME, respectively; we also point out that CAAL represents the output action on a channel w using the label 'w instead of $\overline{\mathtt{w}}$. To not compromise the readability of the figure, we limited (without losing in accuracy) the formula of the $\mathtt{RNA_{pol}}$ process to the coding region of the HbS gene's first exon. The checkmarks on the "Status" column indicate that all the formulae are satisfied.

3.3 Discussion

Through the Glu6Val model, it is possible to analyse how such a point mutation affects the folding process in relation to the hydrophobic interactions.

To better understand this aspect, we can adjust the model of the translation process $\mathcal{L}$ to focus on the type of side chain that characterises each amino acid. From this perspective, we can distinguish two classes of side chains: *hydrophobic* and *hydrophilic*. Precisely, given the labels `hbsc` and `hlsc`, representing, respectively, an amino acid with a hydrophobic side chain and one with a hydrophilic side chain, we construct the process $\mathtt{R}_{\mathcal{J}h}$ by applying to the CCS specification of the `R` subprocess of $\mathcal{L}$ the following transformations:

$$
\begin{array}{l}
\mathcal{C}_{\triangleright} \stackrel{\text{def}}{=} \overline{\mathtt{imet}}.\mathcal{C}; \;\rightarrow\; \mathcal{C}_{\triangleright} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C};
\end{array}
$$

$$
\begin{array}{llll|llll}
\mathtt{ALA} \stackrel{\text{def}}{=} \overline{\mathtt{ala}}.\mathcal{C}; & \rightarrow & \mathtt{ALA} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{ARG} \stackrel{\text{def}}{=} \overline{\mathtt{arg}}.\mathcal{C}; & \rightarrow & \mathtt{ARG} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{CYS} \stackrel{\text{def}}{=} \overline{\mathtt{cys}}.\mathcal{C}; & \rightarrow & \mathtt{CYS} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{ASN} \stackrel{\text{def}}{=} \overline{\mathtt{asn}}.\mathcal{C}; & \rightarrow & \mathtt{ASN} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{GLY} \stackrel{\text{def}}{=} \overline{\mathtt{gly}}.\mathcal{C}; & \rightarrow & \mathtt{GLY} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{ASP} \stackrel{\text{def}}{=} \overline{\mathtt{asp}}.\mathcal{C}; & \rightarrow & \mathtt{ASP} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{ILE} \stackrel{\text{def}}{=} \overline{\mathtt{ile}}.\mathcal{C}; & \rightarrow & \mathtt{ILE} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{GLN} \stackrel{\text{def}}{=} \overline{\mathtt{gln}}.\mathcal{C}; & \rightarrow & \mathtt{GLN} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{LEU} \stackrel{\text{def}}{=} \overline{\mathtt{leu}}.\mathcal{C}; & \rightarrow & \mathtt{LEU} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{GLU} \stackrel{\text{def}}{=} \overline{\mathtt{glu}}.\mathcal{C}; & \rightarrow & \mathtt{GLU} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{MET} \stackrel{\text{def}}{=} \overline{\mathtt{met}}.\mathcal{C}; & \rightarrow & \mathtt{MET} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{HIS} \stackrel{\text{def}}{=} \overline{\mathtt{his}}.\mathcal{C}; & \rightarrow & \mathtt{HIS} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{PHE} \stackrel{\text{def}}{=} \overline{\mathtt{phe}}.\mathcal{C}; & \rightarrow & \mathtt{PHE} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{LYS} \stackrel{\text{def}}{=} \overline{\mathtt{lys}}.\mathcal{C}; & \rightarrow & \mathtt{LYS} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{PRO} \stackrel{\text{def}}{=} \overline{\mathtt{pro}}.\mathcal{C}; & \rightarrow & \mathtt{PRO} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{SER} \stackrel{\text{def}}{=} \overline{\mathtt{ser}}.\mathcal{C}; & \rightarrow & \mathtt{SER} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{TRP} \stackrel{\text{def}}{=} \overline{\mathtt{trp}}.\mathcal{C}; & \rightarrow & \mathtt{TRP} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{THR} \stackrel{\text{def}}{=} \overline{\mathtt{thr}}.\mathcal{C}; & \rightarrow & \mathtt{THR} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; & \\
\mathtt{VAL} \stackrel{\text{def}}{=} \overline{\mathtt{val}}.\mathcal{C}; & \rightarrow & \mathtt{VAL} \stackrel{\text{def}}{=} \overline{\mathtt{hbsc}}.\mathcal{C}; & & \mathtt{TYR} \stackrel{\text{def}}{=} \overline{\mathtt{tyr}}.\mathcal{C}; & \rightarrow & \mathtt{TYR} \stackrel{\text{def}}{=} \overline{\mathtt{hlsc}}.\mathcal{C}; &
\end{array}
\tag{3.36}
$$

As proved through model checking (see Figure 3.2), $\mathtt{R}_{\mathcal{J}h}$ satisfies the following formulae derived from Equation 3.35:

$$
\begin{array}{l}
\mathtt{R}_{\mathcal{J}h} \vDash \\
\langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \underline{\langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hlsc}} \rangle} \quad \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hlsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hbsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \mathbf{tt}
\end{array}
\tag{3.37}
$$

for the normal HBB gene;

$$
\begin{array}{l}
\mathtt{R}_{\mathcal{J}h} \vDash \\
\langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \underline{\langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hbsc}} \rangle} \quad \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hlsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{hbsc}} \rangle \quad \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hbsc}} \rangle \\
\langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{hlsc}} \rangle \mathbf{tt}
\end{array}
\tag{3.38}
$$

for the HbS gene.

Status	Time	Property	Verify
✓	75 ms	HIRIBOSOME ⊨ <a><u><g><'hbsc> <g><u><g><'hbsc><c><a><u><'hlsc><c><u><g> <'hbsc><a><c><u><'hlsc><c><c><u><'hbsc><g><a><g><'hlsc><g><a><g><'hlsc><a> <a><g><'hlsc><u><c><u><'hlsc><g><c><c><'hbsc><g><u><u><'hbsc><a><c><u> <'hlsc>tt	▶
✓	75 ms	HIRIBOSOME ⊨ <a><u><g><'hbsc> <g><u><g><'hbsc><c><a><u><'hlsc><c><u><g> <'hbsc><a><c><u><'hlsc><c><c><u><'hbsc><g><u><g><'hbsc><g><a><g><'hlsc><a> <a><g><'hlsc><u><c><u><'hlsc><g><c><c><'hbsc><g><u><u><'hbsc><a><c><u> <'hlsc>tt	▶

Figure 3.2 – Verification performed through the CAAL web-based tool of the specifications provided for expressing a portion of the correct HBB gene sequence (first row) and the Glu6Val mutation (second row), described in terms of the hydrophobic/hydrophilic property of their amino acids. The red boxes highlight the difference between the normal codon and the mutated one. The $\mathtt{R}_{\mathcal{J}^h}$ process is transliterated as HIRIBOSOME; we recall that CAAL represents the output action on a channel $\mathtt{w}$ using the label $'\mathtt{w}$ instead of $\overline{\mathtt{w}}$. The checkmarks on the "Status" column indicate that the formulae are satisfied.

Therefore, the portion of the aminoacidic sequence of the haemoglobin β subunit can be written in terms of the hydrophobic or hydrophilic property of each amino acid:

$$\psi^{hi}_{hbb} = \text{"}\mathtt{hbsc\ hbsc\ hlsc\ hbsc\ hlsc\ hbsc\ hlsc\ hlsc\ hlsc\ hlsc\ hbsc\ hbsc\ hlsc}\text{"} \tag{3.39}$$

for the normal HBB;

$$\psi^{hi}_{hbs} = \text{"}\mathtt{hbsc\ hbsc\ hlsc\ hbsc\ hlsc\ hbsc\ hbsc\ hlsc\ hlsc\ hlsc\ hbsc\ hbsc\ hlsc}\text{"} \tag{3.40}$$

in the case of the Glu6Val mutation.

Using the model of protein folding described in the previous chapter (Definition 2.1), it is possible to formally describe how the expression of a gene can affect the conformation of a protein.

Firstly, we recall how the hydrophobic interactions have been modelled in the $\mathcal{F}^s_{\mathtt{p}}$ process (protein folding step):

$$\begin{array}{lcl}
\mathcal{F}^s_{\mathtt{p}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\mathcal{J}1_{\mathtt{aa}} + \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{J}^h_{\mathtt{aa}}}; \\
\mathcal{J}1_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{aa}}} + \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}}; \\
\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{aa}}; \\
\Delta\mathtt{G}_{\mathcal{J}^h_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{aa}}; \\
\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa}}; \\
 & \vdots & \\
\mathcal{J}^e_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^s_{\mathtt{p}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{p}}; \\
\mathcal{J}^h_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlsc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbsc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{esc}}.\mathcal{F}^s_{\mathtt{p}}; \\
\mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bsc}}.\mathcal{F}^s_{\mathtt{p}}
\end{array} \tag{3.41}$$

where aa indicates an amino acid molecule, ndg represents the negative ΔG value of the process, hlsc and hbsc stand, respectively, for hydrophilic and hydrophobic side chain, while esc and bsc are the labels used to describe that a side chain is exposed to the environment or buried inside in the hydrophobic core of the protein (see Tables 2.1 on page 37 and 2.2 on page 38 for the meaning of each action label and process name). Now we can write an HML formula that specifies the behaviour of the $\mathcal{F}^s_{\mathrm{p}}$ process when applied to the amino acid sequences of ψ^{hi}_{hbb} and ψ^{hi}_{hbs}. As shown in Figure 3.3, we can demonstrate through model checking that $\mathcal{F}^s_{\mathrm{p}}$ actually satisfies this kind of formula; that is:

$$
\begin{aligned}
&\mathcal{F}^s_{\mathrm{p}} \models \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\mathbf{tt} \\
&\wedge \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hbsc}\rangle\langle \overline{\mathtt{bsc}}\rangle \\
&\langle \mathtt{aa}\rangle\langle \mathtt{ndg}\rangle\langle \mathtt{hlsc}\rangle\langle \overline{\mathtt{esc}}\rangle\mathbf{tt}
\end{aligned}
\tag{3.42}
$$

Equation 3.42 proves that the hydrophobicity of an amino acid determines its position on the inside or outside of a protein and, consequently, affects the latter three-dimensional conformation. In addition, through Equations 3.28, 3.30, and 3.33 we can trace back this property to the sequence of the originating gene; finally, Equations 3.35, 3.37, and 3.38 demonstrate that the types of amino acids of the polypeptide chain, in terms of their hydrophobic properties, can be affected by the modification of a single nucleotide of the related gene. *Therefore, we formally proved that a point mutation can modify the three-dimensional conformation of a protein.*

In contrast, the folding of the mRNA generated by the HBB shows different behaviour because each nucleotide interacts in the same way with water.

Summarising the CCS specification of the $\mathcal{F}^s_{\mathtt{rna}}$ process (RNA folding step) of the RNA folding model (see Definition 2.1), it is possible to note that each unpaired base is always buried and stacked on top of another one:

$$
\begin{aligned}
\mathcal{F}^s_{\mathtt{rna}} &\stackrel{\text{def}}{=} \mathtt{ub}.\mathcal{J}1_{\mathtt{n}} + \mathtt{ub}.\mathcal{J}2_{\mathtt{n}} + \mathtt{srsr}.\mathcal{J}1_{\mathtt{n}} + \\
&\quad\ \mathtt{drdr}.\mathcal{J}1_{\mathtt{n}} + \mathtt{srdr}.\mathcal{J}1_{\mathtt{n}} + \mathtt{tpb}.\mathcal{J}1_{\mathtt{n}}; \\
\mathcal{J}1_{\mathtt{n}} &\stackrel{\text{def}}{=} \mathtt{ub}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{srsr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{drdr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \\
&\quad\ \mathtt{srdr}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{tpb}.\Delta G_{\mathcal{J}^e_{\mathtt{b}}}; \\
\mathcal{J}2_{\mathtt{n}} &\stackrel{\text{def}}{=} \mathtt{ub}.\Delta G_{\mathcal{P}_{\mathtt{b}2}} + \mathtt{ub}.\Delta G_{\mathcal{J}^h_{\mathtt{b}}} + \mathtt{srsr}.\Delta G_{\mathcal{P}_{\mathtt{b}3}} + \\
&\quad\ \mathtt{drdr}.\Delta G_{\mathcal{P}_{\mathtt{b}3}} + \mathtt{srdr}.\Delta G_{\mathcal{P}_{\mathtt{b}3}}; \\
\Delta G_{\mathcal{J}^e_{\mathtt{b}}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{J}^e_{\mathtt{b}}; \\
\Delta G_{\mathcal{J}^h_{\mathtt{b}}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{J}^h_{\mathtt{b}}; \\
&\ \ \vdots \\
\mathcal{J}^e_{\mathtt{b}} &\stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}^s_{\mathtt{rna}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{rna}}; \\
\mathcal{J}^h_{\mathtt{b}} &\stackrel{\text{def}}{=} hbi.\mathrm{I}_{\mathtt{rna}}; \\
\mathrm{I}_{\mathtt{rna}} &\stackrel{\text{def}}{=} \overline{\mathtt{bb}}.\mathcal{S}; \\
\mathcal{S} &\stackrel{\text{def}}{=} \overline{\mathtt{sb}}.\mathcal{F}^s_{\mathtt{rna}}
\end{aligned}
\tag{3.43}
$$

where, referring in particular to the $\mathcal{I}^h_{\mathtt{b}}$ (base hydrophobic interaction) subprocess, `ub` represents an unpaired base, *hbi* stands for "hydrophobic interaction", `bb` indicates that a base is buried inside the RNA molecule, while `sb` that it is stacked parallel to another one.

As proved through model checking (see Figure 3.3), the $\mathcal{F}^s_{\mathtt{rna}}$ process satisfies, for every couple of unpaired bases, the following specification, which describes the behaviour of the hydrophobic interaction during the RNA folding:

$$\mathcal{F}^s_{\mathtt{rna}} \vDash \langle \mathtt{ub} \rangle \langle \mathtt{ub} \rangle \langle \mathtt{ndg} \rangle \langle \mathit{hbi} \rangle \langle \overline{\mathtt{bb}} \rangle \tag{3.44}$$

This means that, *in relation to the hydrophobic interactions, a point mutation cannot significantly affect the folding of RNA.*

Status	Time	Property	Verify
✓	25 ms	PFS ⊨ <aa><ndg><hbsc><'bsc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc><aa><ndg><hlsc><'esc><aa><ndg><hlsc><'esc><aa><ndg><hlsc><'esc><aa><ndg><hbsc><'bsc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc>tt and <aa><ndg><hbsc><'bsc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc><aa><ndg><hbsc><'bsc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc><aa><ndg><hlsc><'esc><aa><ndg><hlsc><'esc><aa><ndg><hbsc><'bsc><aa><ndg><hbsc><'bsc><aa><ndg><hlsc><'esc>tt	▶
✓	25 ms	RNAFS ⊨ <ub><ub><ndg><hbi><'bb>tt	▶

Figure 3.3 – Verification performed through CAAL of the specifications provided for the folding process of the haemoglobin β subunit (first row) and for the related mRNA (second row); they are described in terms of the hydrophobic interactions performed, respectively, by amino acids and nucleotides. The $\mathcal{F}^s_{\mathtt{p}}$ process is transliterated as PFS, while the $\mathcal{F}^s_{\mathtt{rna}}$ process as RNAFS. The checkmarks on the "Status" column indicate that the formulae are satisfied. The output action on each channel `w` is represented trough the label $'\mathtt{w}$.

3.4 Conclusions

In this chapter, we model gene expression through a process-based approach. A pathology that affects the folding process is then analysed, with the aid of CCS and Hennessy-Milner logic, as behaviour resulting from the interactions between the monomeric units of proteins (amino acids) and RNAs (nucleotides).

The second part of our study focuses on molecular hydrophobicity, which is critical in proteins because each amino acid is specifically hydrophobic or hydrophilic. This property is not equally valid for RNAs, in which the nucleotides interact almost in the same way with water, even if different base stacking can have dissimilar energetic values [79].

In Chapter 2, we demonstrate that the distinctive interactions among amino acids lead proteins to fold up into three-dimensional shapes more complex if compared to those characterising RNAs. Due to the tight relationship between structure and function that exists in biological

systems, these conformations allow proteins to perform more effectively most of the functions carried out by non-coding RNAs. However, as proved through the models proposed in this chapter, the greater complexity of proteins has the drawback of exposing them to some pathologies that do not affect the simpler structure of RNAs.

Further studies in this direction will involve the analysis of other pathologies associated with protein misfolding [46], particularly those responsible for ageing-related diseases (such as Alzheimer's and Parkinson's) [39]. This kind of investigation may benefit from improving and extending our models, especially by complementing the formal approaches described in this chapter with other algebraic and computational methods, such as topological data analysis and graph grammars [68, 73, 96, 98, 103].

Although the adopted approach is strictly theoretical, we propose it as an alternative standpoint to observe biological systems. A process-based view of molecular structures and functions can bring out congruence and dissimilarities difficult to detect through other computational methods or experimental techniques; this perspective can thus inspire the investigation of properties not yet considered in the current studies.

Chapter 4

Algebraic Characterisation of Non-coding RNA*

4.1 Introduction

The relationship between structures and functions is an important topic in biology, and different computational approaches, from process calculi to topological data analysis, have contributed significantly to its study [13, 27, 69, 73, 89].

In particular, formal languages and graph grammars have been successfully applied in modelling the properties that correlate the functions expressible by ribonucleic acid (RNA) molecules and specific substructures involved in their folding process [68, 96]. In Chapter 2, we pushed forward this approach and proved that the complexity of RNA functions can be traced back to the inner potentiality of each nucleotide to interact with others in the same sequence. This result has been obtained by comparing the RNA folding with that performed by proteins to identify an abstraction level at which these two classes of molecules show the same structural and functional complexity. We refer to this level as *RNA and protein congruence level* (or, simply, *congruence level*). Reaching such a goal was possible thanks to the expressiveness of process algebras [1], through which we modelled both RNA and protein folding.

In this chapter, we want to hypothesise the functions that characterise the *congruence level* and further explore the applicability of process algebras in modelling the related biological processes. The resulting models will form the basis of an agent-based simulation (see Section 1.3.5 on page 29).

*This chapter is derived from a co-authored work, conducted and published as part of the PhD project: Maestri, S., Merelli, E., 2020. "Algebraic Characterisation of Non-coding RNA", in: Cazzaniga, P., Besozzi, D., Merelli, I., Manzoni, L. (Eds.), *Computational Intelligence Methods for Bioinformatics and Biostatistics, Lecture Notes in Computer Science.* Springer International Publishing, Cham, pp. 145-158. ©2020 Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-63061-4_14. S.M. implemented the method, performed the research and wrote the paper. E.M. supervised the research. Both the authors designed and reviewed the paper.

The agent-based simulator mentioned in this chapter is designed to investigate the molecular interactions that characterise metabolic pathways and to analyse the global properties resulting from local interactions [22]. We simulated complete enzymatic reactions by modelling the molecules involved (enzymes, metabolites, and complexes) as autonomous and interactive agents. We will extensively discuss this agent-based method for simulating biomolecular interactions in Part II of this manuscript. For this reason, the present chapter can be regarded as a bridge between the two modelling approaches considered in this dissertation.

The RNA models we propose are algebraic specifications of new functionalities that will enrich the simulator. We expect that, similarly to the results we obtained regarding metabolic reactions, analysing the behaviour resulting from agents' interactions will yield additional information on the biological properties of RNAs.

4.2 Results

At the abstraction level we are exploring, the behavioural equivalence between RNAs and proteins has been reached by reducing the complexity of protein folding (limiting the number of amino acids that can interact through hydrogen bonds). This limitation also reduces the complexity of the structures–hence of the functions–the folding process can express. The functions we can represent at this level of abstraction belong to the *non-coding RNA congruence class*, that is, the class of all the functions performed by non-coding RNAs (ncRNAs). The *congruence level* introduced in Chapter 2 characterises the congruence relation that defines the ncRNA congruence class, whose complete formalisation will be provided in future work.

In this chapter, we model two functions carried out by ncRNAs in cells, *ligand binding* and *enzymatic activity*, which together specifically characterise a subclass of non-coding RNAs called *ribozymes*. They are able to catalyse biochemical reactions similarly to protein enzymes, carrying out fundamental roles in cellular processes [58, 106].

From this point of view, a ribozyme can be seen as a process capable of performing in parallel the tasks mentioned above, even if they can mutually affect one another.

Definition 4.1 (Ribozyme)**.** A *ribozyme* $\mathcal{R}$ is a process whose behaviour is given by the following defining equation:

$$\mathcal{R} \stackrel{\text{def}}{=} \mathcal{B} \mid \mathcal{E}; \tag{4.1}$$

where $\mathcal{B}$ and $\mathcal{E}$ are, respectively, the *ligand binding* and *enzymatic activity* processes.

In the remainder of this chapter, we formally define both $\mathcal{B}$ and $\mathcal{E}$.

4.2.1 Ligand binding

Through specific binding sites, ribozymes can bind small molecules necessary to carry out their enzymatic functions. As an example, the binding of GlcN6P to the glmS ribozyme is fundamental for enabling the glmS catalytic activity [35, 125]. In our models, the *ligand-binding function* consists in gaining a ligand through a binding site of the RNA molecule to

- store the ligand;
- trigger or interrupt another function of the same molecule.

A ligand can bind to a free binding site only if it shows steric and electrostatic complementarity to this site (two properties labelled sc and ec, respectively). If a steric hindrance (sn) or an electrostatic non-complementarity (en) is present, binding the ligand is not possible.

The model of this functional role is provided by the *ligand binding* process ($\mathcal{B}$), which takes a free RNA binding site (bs) and a ligand (l) as input and checks the sc and ec constraints. If both these conditions are satisfied, it produces an occupied binding site ($\mathtt{bs}_*$) as output; otherwise, the binding site is left free, and the RNA molecule is ready to check the compatibility of another ligand.

To remain as faithful as possible to the biological process and avoid the common problem of state explosion during the simulation, we abstract the parallel verification of the steric and electrostatic constraints as a non-deterministic choice.

When the binding site is occupied, three events can be triggered:

1. the binding site is maintained occupied to store the bound ligand;
2. the ligand is released;
3. a second function is activated or interrupted.

Based on the above description, we provide the following CCS specification of the process that allows checking if a ligand can be stored, producing as output an occupied binding site (bs_*).

Definition 4.2 (Ligand binding)**.** We define the *ligand binding* performed by a ribozyme $\mathcal{R}$ as the process $\mathcal{B}$ whose behaviour is specified by the following CCS equation:

$$
\begin{array}{lcl}
\mathcal{B} & \stackrel{\text{def}}{=} & \mathtt{l}.(\mathtt{SC}_v + \mathtt{EC}_v); \\
\mathtt{SC}_v & \stackrel{\text{def}}{=} & \mathtt{sc}.\mathtt{SC} + \mathtt{sn}.\mathtt{SN}; \\
\mathtt{SC} & \stackrel{\text{def}}{=} & \mathtt{ec}.\mathtt{BS}_* + \mathtt{en}.\mathtt{EN}; \\
\mathtt{EC}_v & \stackrel{\text{def}}{=} & \mathtt{ec}.\mathtt{EC} + \mathtt{en}.\mathtt{EN}; \\
\mathtt{EC} & \stackrel{\text{def}}{=} & \mathtt{sc}.\mathtt{BS}_* + \mathtt{sn}.\mathtt{SN}; \\
\mathtt{SN} & \stackrel{\text{def}}{=} & \overline{\mathtt{bs}}.\mathcal{B}; \\
\mathtt{EN} & \stackrel{\text{def}}{=} & \overline{\mathtt{bs}}.\mathcal{B}; \\
\mathtt{BS}_* & \stackrel{\text{def}}{=} & \overline{\mathtt{bs}_*}.0.
\end{array}
\tag{4.2}
$$

For a complete explanation of the symbols used in this and the following models, refer to Tables 4.1 and 4.2.

4.2.2 Enzymatic activity

Ribozymes perform a variety of enzymatic activities in cells, for which several analogies have been found to those carried out by proteins [31]. Since the present work is intended to outline a model of the functions characterising the *congruence level* that relates RNAs and proteins, we can generalise the enzymatic activity of ribozymes as the catalysis of a reaction.

Formalising this process requires first providing a basic model of a chemical reaction. A reaction, such as $S \rightleftharpoons P$, can be modelled in its key properties with two complementary *reaction directions*, represented by the following processes:

- *Forward reaction direction* ($\mathcal{R}_{\mathtt{fd}}$): starting from a substrate, generates one or more products;
- *Backward reaction direction* ($\mathcal{R}_{\mathtt{bd}}$): starting from the products, generate the original substrate.

The choice between $\mathcal{R}_{\mathtt{fd}}$ and $\mathcal{R}_{\mathtt{bd}}$ is determined by the value of the respective *free energy change* (ΔG): only the reaction direction with a negative ΔG can occur. This property has been modelled by placing both $\mathcal{R}_{\mathtt{fd}}$ and $\mathcal{R}_{\mathtt{bd}}$ in parallel composition with the $\Delta\mathtt{G}$ process; it produces the three possible outputs representing the values that the free energy variation can assume: negative, positive or zero (ndg, pdg and zdg, respectively).

Definition 4.3 (Reaction)**.** A *reaction* $\mathcal{R}$ is a process whose behaviour is given by the following defining equation:

$$
\begin{array}{lcl}
\mathcal{R} & \stackrel{\text{def}}{=} & (\mathcal{R}_{\mathtt{fd}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\} + \\
 & & (\mathcal{R}_{\mathtt{bd}} | \Delta\mathtt{G}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\}; \\
\Delta\mathtt{G} & \stackrel{\text{def}}{=} & \overline{\mathtt{ndg}}.\Delta\mathtt{G} + \overline{\mathtt{pdg}}.\Delta\mathtt{G} + \overline{\mathtt{zdg}}.\Delta\mathtt{G}; \\
\mathcal{R}_{\mathtt{fd}} & \stackrel{\text{def}}{=} & \mathtt{s}.\mathtt{S}_{\mathtt{fd}}; \\
\mathtt{S}_{\mathtt{fd}} & \stackrel{\text{def}}{=} & \mathtt{p}.\Delta\mathtt{G}_{\mathtt{fd}}; \\
\Delta\mathtt{G}_{\mathtt{fd}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathtt{P}_{\mathtt{fd}}; \\
\mathtt{P}_{\mathtt{fd}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ts}}.\mathtt{TS}_{\mathtt{fd}}; \\
\mathtt{TS}_{\mathtt{fd}} & \stackrel{\text{def}}{=} & \overline{\mathtt{p}}.\mathcal{R}; \\
\mathcal{R}_{\mathtt{bd}} & \stackrel{\text{def}}{=} & \mathtt{p}.\mathtt{P}_{\mathtt{bd}}; \\
\mathtt{P}_{\mathtt{bd}} & \stackrel{\text{def}}{=} & \mathtt{s}.\Delta\mathtt{G}_{\mathtt{bd}}; \\
\Delta\mathtt{G}_{\mathtt{bd}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathtt{S}_{\mathtt{bd}}; \\
\mathtt{S}_{\mathtt{bd}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ts}}.\mathtt{TS}_{\mathtt{bd}}; \\
\mathtt{TS}_{\mathtt{bd}} & \stackrel{\text{def}}{=} & \overline{\mathtt{s}}.\mathcal{R};
\end{array}
\tag{4.3}
$$

We want to point out that the modelled reaction is driven by the free energy reduction. The $\Delta G_{\mathtt{fd}}$ and $\Delta G_{\mathtt{bd}}$ processes check if the ΔG of the related reaction direction is negative.

Before producing its final output (p for $\mathcal{R}_{\mathtt{fd}}$ and s for $\mathcal{R}_{\mathtt{bd}}$), each reaction direction has an intermediate output, the transition state (ts). The *enzymatic activity*, modelled as the $\mathcal{E}$ process, catalyses the reaction by taking as input this transition state and an active site (as). The latter is a catalytic binding site; therefore, similarly to what we described for the ligand-binding function, it must show steric and electrostatic complementarity with the transition state for the $\mathcal{E}$ process to proceed. If these constraints are satisfied, the $\mathcal{E}$ process performs a transition to the ES state, representing the formation of the enzyme-substrate (ES) complex; otherwise, if there is steric non-complementarity (sn) or electrostatic non-complementarity (en), the active site remains free, and the ribozyme can check another transition state. As in the case of the $\mathcal{B}$ process, this verification has been modelled as a non-deterministic choice.

On the ES complex acts the binding energy of the enzyme to perform the *catalysis*, modelled with the process $\mathcal{C}$, which causes the reduction of the activation energy of the reaction (aer), to obtain the output of one of the two reaction directions.

Here we propose a simplified specification for the $\mathcal{E}$ process.

Definition 4.4 (Enzymatic activity)**.** We define *enzymatic activity* of a ribozyme $\mathcal{R}$ the process $\mathcal{E}$ whose behaviour is given by the following CCS specification:

$$
\begin{aligned}
\mathcal{E} &\stackrel{\text{def}}{=} \mathtt{ts}.(\mathtt{SC}_v + \mathtt{EC}_v);\\
\mathtt{SC}_v &\stackrel{\text{def}}{=} \mathtt{sc}.\mathtt{SC} + \mathtt{sn}.\mathtt{SN};\\
\mathtt{SC} &\stackrel{\text{def}}{=} \mathtt{ec}.\mathtt{ES} + \mathtt{en}.\mathtt{EN};\\
\mathtt{EC}_v &\stackrel{\text{def}}{=} \mathtt{ec}.\mathtt{EC} + \mathtt{en}.\mathtt{EN};\\
\mathtt{EC} &\stackrel{\text{def}}{=} \mathtt{sc}.\mathtt{ES} + \mathtt{sn}.\mathtt{SN};\\
\mathtt{SN} &\stackrel{\text{def}}{=} \overline{\mathtt{as}}.\mathcal{E};\\
\mathtt{EN} &\stackrel{\text{def}}{=} \overline{\mathtt{as}}.\mathcal{E};\\
\mathtt{ES} &\stackrel{\text{def}}{=} \mathtt{es}.\mathcal{C};\\
\mathcal{C} &\stackrel{\text{def}}{=} \overline{\mathtt{aer}}.(\mathtt{TS}_{\mathtt{fd}} + \mathtt{TS}_{\mathtt{bd}});
\end{aligned}
\tag{4.4}
$$

To further clarify how this process works, Figure 4.1 shows its labelled transition system (LTS), automatically generated with the aid of the CAAL concurrency workbench [3].

The models of *ligand binding* and *enzymatic activity* are part of the engineering life cycle for the simulation of ribozyme functions, where they outline the *process modelling*; as depicted in Figure 4.3, the subsequent step is represented by the *model verification*. In the next section, we discuss this step to cover the whole first phase of the engineering life cycle. In future works, we will provide the *modelling*, *simulation* and *validation* of the system in which ribozymes and metabolites will be represented as concurrent agents.

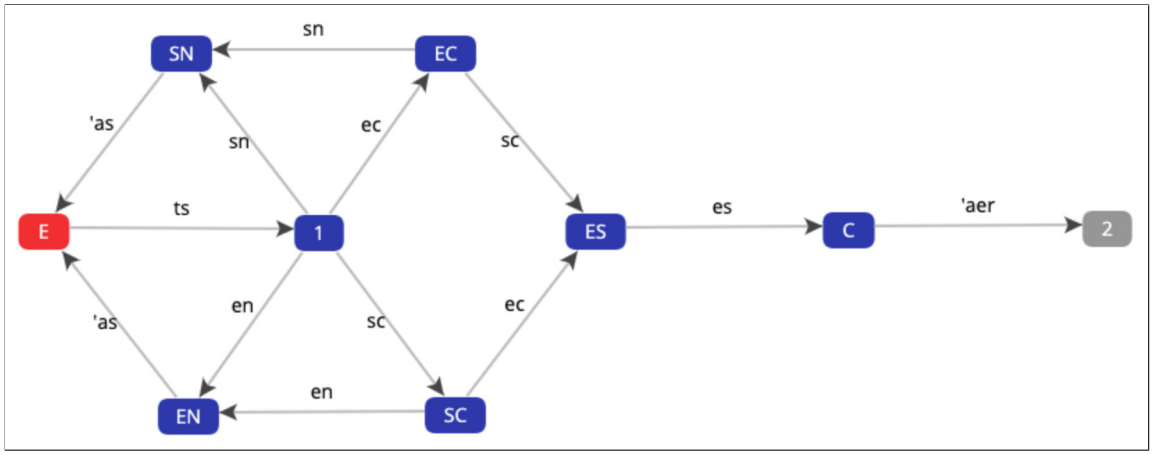


Figure 4.1 – Labelled transition system (LTS) of the $\mathcal{E}$ process. We recall that, in an LTS, each transition $\mathsf{P} \xrightarrow{\mathtt{w}} \mathsf{P}'$ means that the process P can become the process P' by performing the action $\mathtt{w}$. Each process/state has been transliterated from the CCS model, while action labels are left unchanged; output actions are indicated with a quotation mark. State 1 represents the $\mathtt{SC}_\nu + \mathtt{EC}_\nu$ choice, while state 2 corresponds to $\mathtt{TS}_{\mathtt{fd}} + \mathtt{TS}_{\mathtt{bd}}$.

4.2.3 Model checking

To show the validity of the models described in the previous section, we provide the verification of two biochemical properties of ribozyme functions; we also ensure that the free energy reduction drives all the reactions. Such biochemical properties are expressed as Hennessy-Milner logic (HML) formulas so that we can establish, via model checking, if they are satisfied [63].

- If a free binding site and a ligand exhibit steric complementarity, but they do not also show electrostatic complementarity, the binding site cannot be occupied:

$$\mathcal{R} \vDash \langle \mathtt{bs} \rangle \langle \mathtt{l} \rangle \langle \mathtt{sc} \rangle \langle \mathtt{en} \rangle [\overline{\mathtt{bs}_*}] \mathbf{ff} \tag{4.5}$$

- If the free active site of an ncRNA has electrostatic complementarity with a transition state but, at the same time, a steric hindrance is present, the active site cannot be occupied–i.e., it remains free ($\mathtt{as}$):

$$\mathcal{R} \vDash \langle \mathtt{as} \rangle \langle \mathtt{ts} \rangle \langle \mathtt{ec} \rangle \langle \mathtt{sn} \rangle \langle \overline{\mathtt{as}} \rangle \mathbf{tt} \tag{4.6}$$

- In order for a substrate and a product to form a transition state, the ΔG of the reaction must be negative:

$$\mathcal{R}_{\mathtt{fd}} \vDash \langle \mathtt{s} \rangle \langle \mathtt{p} \rangle \langle \mathtt{ndg} \rangle \langle \overline{\mathtt{ts}} \rangle \mathbf{tt} \tag{4.7}$$

The verification of these formulas has been made with the CAAL web-based model checking function [3]. The results are shown in Figure 4.2.

Table 4.1 – Symbols used to denote the processes of Equations 4.2, 4.3, and 4.4.

Process\State	Transliteration	Description
$\mathtt{BS}_*$	BSo	binding site occupied
$\mathcal{B}$	B	ligand binding
$\mathcal{C}$	C	catalysis
$\Delta\mathtt{G}$	DG	Gibbs free energy change (ΔG)
$\Delta\mathtt{G}_{\mathtt{fd}}$	DGfd	ΔG of the forward reaction direction
$\Delta\mathtt{G}_{\mathtt{bd}}$	DGbd	ΔG of the backward reaction direction
$\mathcal{E}$	E	enzymatic activity
EC	EC	electrostatic complementarity
$\mathtt{EC}_\nu$	ECv	electrostatic complementarity check
EN	EN	electrostatic non-complementarity
ES	ES	enzyme-substrate complex
$\mathtt{P}_{\mathtt{fd}}$	Pfd	product in the forward reaction direction
$\mathtt{P}_{\mathtt{bd}}$	Pbd	product in the backward reaction direction
$\mathcal{R}$	R	reaction
$\mathcal{R}_{\mathtt{fd}}$	Rfd	forward reaction direction (from substrate to product)
$\mathcal{R}_{\mathtt{bd}}$	Rbd	backward reaction direction (from product to substrate)
$\mathcal{R}$	RIBOZYME	ribozyme
$\mathtt{S}_{\mathtt{fd}}$	Sfd	substrate in the forward reaction direction
$\mathtt{S}_{\mathtt{bd}}$	Sbd	substrate in the backward reaction direction
SC	SC	steric complementarity
$\mathtt{SC}_\nu$	SCv	steric complementarity check
SN	SN	steric non-complementarity
$\mathtt{TS}_{\mathtt{fd}}$	TSfd	transition state of the forward reaction direction
$\mathtt{TS}_{\mathtt{bd}}$	TSbd	transition state of the backward reaction direction

Table 4.2 – Description of the action labels used in Equations 4.2, 4.3, and 4.4.

Action label	**Description**
`aer`	activation energy reduction
`as`	free active site
`bs`	free binding site
$\texttt{bs}_*$	occupied binding site
`ec`	electrostatic complementarity
`en`	electrostatic non-complementarity
`es`	enzyme-substrate complex
`l`	ligand
`ndg`	$\Delta G < 0$
`p`	product
`pdg`	$\Delta G > 0$
`s`	substrate
`sc`	steric complementarity
`sn`	steric non-complementarity
`ts`	transition state
`zdg`	$\Delta G = 0$

Status	Time	Property	Verify
✓	25 ms	RIBOZYME ⊨ <bs><l><sc><en>['bso]ff	▶
✓	25 ms	RIBOZYME ⊨ <as><ts><ec><sn><'as>tt	▶
✓	25 ms	Rfd ⊨ <s><p><ndg><'ts>tt	▶

Figure 4.2 – Verification of some biochemical properties of the ribozyme functions, expressed as HML formulas. It has been performed through CAAL concurrency workbench [3]; the checkmarks on the "Status" column indicate that all the formulas are satisfied. The $\mathcal{R}$ and $\mathcal{R}_{\texttt{fd}}$ processes are transliterated RIBOZYME and Rfd, respectively (see Table 4.1); the bs_* action label is transliterated as bso. We recall that CAAL represents the output action on a channel `w` using the label $'\texttt{w}$ instead of $\overline{\texttt{w}}$.

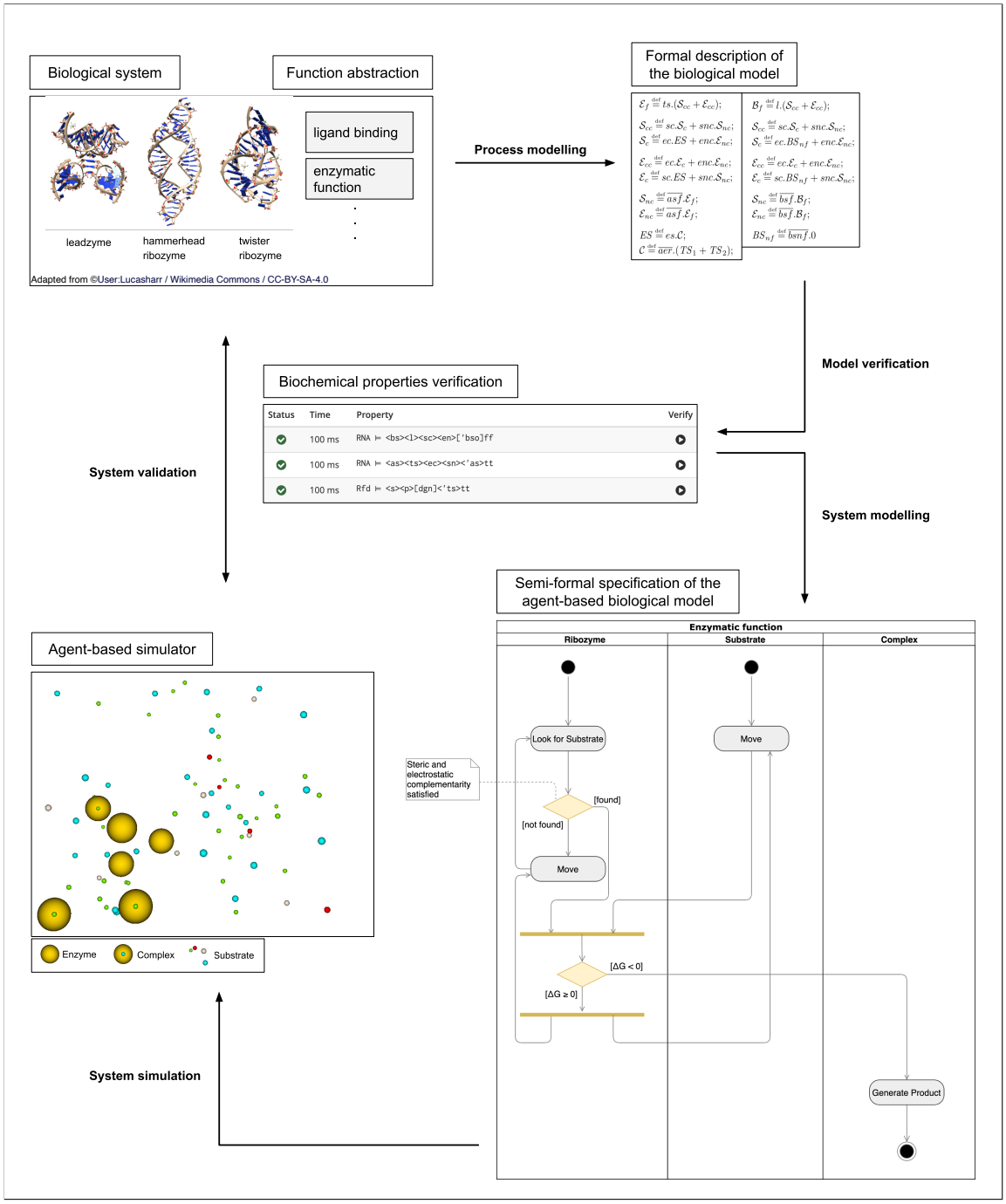


Figure 4.3 – Engineering life cycle for the simulation of ribozyme functions. We can identify five steps–represented through different formalisms–enclosed in two phases: (1) process modelling and verification, (2) system modelling, simulation, and validation. The starting point is the actual biological system, from which we derive an abstraction of the functions we aim to model and simulate. These functions are then modelled using process algebras (CCS in our case), and the properties of the models obtained are verified through the best suitable method for model checking (for our models, we chose Hennessy-Milner logic). This phase is the one explored in the present chapter; the second phase will be defined upon the agent-based simulator described in the second part of this dissertation. It involves the definition of a low-level specification, the generation of the actual agent-based simulation and the validation of the results obtained to make the agent-based model more faithful to the biological system. The UML activity diagram in this figure provides a semi-formal example of the low-level specification.

4.3 Conclusions

In this chapter, we provide a formal description of the functions that RNA molecules can perform at the abstraction level where they have the same complexity as proteins (discussed in Chapter 2). We show how CCS, thanks to its expressiveness, can handle the complexity of modelling non-coding RNA functions, specifically those performed by ribozymes. These functions characterise the congruence class defined by the RNA catalytic activity. The validity of these models has been tested using Hennessy-Milner logic to perform the model checking and confirmed through an automated tool.

These results are a solid basis upon which a multiagent simulator of molecular interactions can be enriched by implementing the functions of non-coding RNAs. The models we provide in this work should be intended as the first phase of the engineering life cycle for the simulation of ribozyme functions (see Figure 4.3).

Beyond their biological roles, ribozymes have been applied in treating respiratory viral infections; it was possible due to their ability to cleave specific RNA segments of influenza viruses, like the influenza A virus or the SARS-coronavirus [41, 86, 111]. The simulations based on the models we propose in this chapter might provide in silico support to further applications of ribozyme mediated inhibition of influenza infections.

Moreover, we are taking just the first steps towards a broader modelling and simulation approach intended to study the behaviour of the more complex class of long non-coding RNAs (lncRNAs). In recent years, it has been increasingly acknowledged the relevance of these molecules in fundamental cellular processes and their involvement in several diseases, such as in tumour progression, where they carry out either an oncogenic or a tumour-suppressive role [97, 99]. We believe that the application of formal methods to the study of non-coding RNAs can provide the perspective necessary to fully understand the behaviour of this class of molecules and thus contribute to the development of concrete strategies for addressing the pathologies in which they are involved.

Part II

Agent-based Modelling and Simulation of Metabolic Pathways

Chapter 5

Background and Methods for Part II

5.1 Introduction

This chapter describes the agent-based modelling and simulation approach that we defined to study molecular interactions in metabolic pathways. Because we chose the glycolysis of baker's yeasts (Saccharomyces cerevisiae) as a case study, Section 5.2 introduces some basic knowledge on glucose oxidation in living cells. In Section 5.3, we then go into the details of the modelling and simulation methods; they comprise a complete description of the choices we made for adapting a kinetic model of yeast's glycolysis to be used as input to a multiagent simulator. Such information is important to understand the studies we propose in the next chapters of this second part of the dissertation.

5.2 Overview of the Glycolytic Pathway

This section outlines the reactions occurring in the glycolytic pathway; the description is fairly general and based on a long-established knowledge of glucose oxidation [65]. The reader already familiar with these concepts can jump directly to Section 5.3, where we provide details on the modelling and simulation methods adopted in this second part of the manuscript.

Glycolysis is the process that degrades, through a series of enzyme-catalysed reactions, a molecule of `glucose` to yield two molecules of `pyruvate` and store some of the released free energy in the form of `ATP` and `NADH`. When glucose degradation happens in the absence of oxygen (anaerobic conditions), it is called fermentation.

The enzymes involved in the glycolysis of all eukaryotic cells are similar in their structures and functions; they only differ in the regulatory processes that determine the fate of pyruvate. The sequential reactions of the glycolytic pathway are usually schematised in *ten steps*. In what follows, we describe the most relevant of these steps and provide the name and the acronym of the related molecular species we will refer to in the remainder of this manuscript.

The initial five steps constitute the *preparatory phase*;

- *first step* : glucose (GLC) is phosphorylated to form glucose 6-phosphate (G6P)
- *second step* : G6P is converted to fructose 6-phosphate (F6P)
- *third step* : F6P is phosphorylated to fructose 1,6-bisphosphate (F16bP)

  (for both the phosphorylations, ATP is the phosphoryl group donor)
- *fourth step* : F16bP is split into dihydroxyacetone phosphate (DHAP) and glyceraldehyde 3-phosphate (GAP)
- *fifth step* : DHAP is converted to a second molecule of GAP

The energy gain occurs in the *payoff phase*, composing the remaining five steps of glycolysis. In the *sixth step*, each molecule of GAP is oxidised and phosphorylated to form 1,3-bisphosphoglycerate (BPG). Energy is then released by converting, *from the seventh to the tenth step*, two molecules of BPG into two molecules of pyruvate (PYR). Much of this energy is conserved, by the phosphorylation of four ADP molecules, into an equal number of ATPs. Since two molecules of ATP are used in the preparatory phase, the net output is two ATPs for each molecule of glucose degraded. During the payoff phase, energy is also stored by forming two molecules of NADH for each molecule of glucose.

In yeasts, pyruvate is further converted, under anaerobic conditions, into ethanol (EtOH) and CO_2, a process called *ethanol (alcohol) fermentation*.

Among the other carbohydrates involved in glycolysis, the only one we take into account in our models is glucose 1-phosphate (G1P), which is converted to G6P during the preparatory phase.

Alongside the main steps described above, the breakdown of glucose can also enter one of the glycolysis branches, which leads to the formation of end products such as trehalose (TRH), glycerol (GLY), and succinate (SUC).

A schematic representation of the steps and branches considered in our work is provided in Figure 5.1.

5.3 Modelling and Simulating the Glycolytic Pathway: an Agent-based Approach

5.3.1 Agent-based simulator for metabolic pathways

The studies proposed in Part II of this dissertation have been carried out with the aid of Orion, a spatial simulator for metabolic pathways. It has been developed in Java starting from a prototype project [5, 8, 22, 38, 71]. The version of the simulator used for this work is Orion 2.0.0, in which we fixed and refined the original software to make it capable of dealing with a large number of molecules and highlighting their interactions.

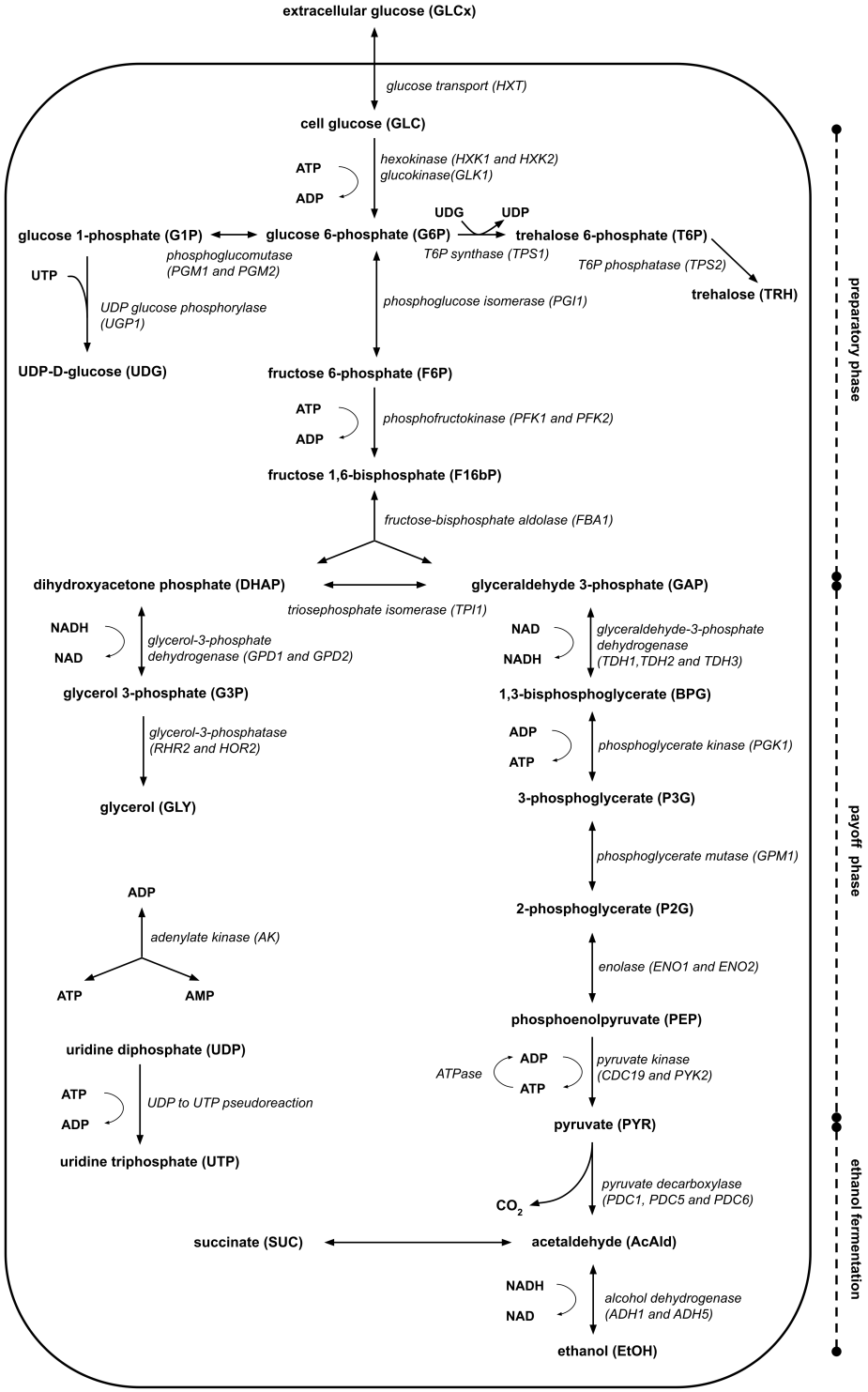


Figure 5.1 – Schematic representation of the glycolysis steps and branches taken as a starting point of our study. For each metabolite and enzyme involved, we reported both the name and the acronym adopted in this manuscript (in bold for metabolites, in italics for enzymes). On the right side of the image, we highlight the three phases identifiable in yeast glycolysis.

Orion is an agent-based simulator; this means that the molecules involved in the pathway are represented by agents: autonomous systems able to perceive changes in their environment and react to them (see Section 1.3.5).

The simulations are performed in the three-dimensional space, representing a portion of the cytoplasm, that is, the environment perceived by the agents. Each molecule is modelled as a sphere, whose radius is estimated from its molecular weight (MW) and the average value of the molar specific volume of a protein in solution (approximately 0.73 cm^3/g) by assuming the following equation [34, 50, 101, 124]:

$$V(\text{Å}^3) = \frac{0.73\ cm^3/g \times 10^{24}\ \text{Å}^3/cm^3 \times \text{MW}\ g/mole}{6.02 \times 10^{23}\ \ molecules/mole} \tag{5.1}$$

According to the data retrievable in the BioNumbers database [78], the average values of the molecular radii are about 20 Å (angstroms) for enzymes and 5 Å for metabolites. By looking at Table 6.1 (of the next chapter), the radii generated from the volumes calculated through Equation 5.1 are in agreement with those experimental results.

Our modelling choices produce a reasonably realistic molecular crowding in the simulated portion of the cytoplasm. Moreover, by making every molecule spherical, we can correlate its shape with its diffusion coefficient through the Stokes-Einstein equation for the Brownian motion of a spherical particle:

$$D = \frac{k_B T}{6\pi\eta r} \tag{5.2}$$

where k_B is the Boltzmann constant, T the temperature, η the viscosity of the environment and r the radius of the molecule. For our simulations, we set $T = 298.15$ kelvin and $\eta = 0.0011$ pascal-second.

Each molecule can freely move inside the simulation volume according to a vector applied to the centre of its sphere: its direction is generated randomly, based on polar coordinates, while its module is calculated from the ambient diffusion coefficient D, obtained via Equation 5.2, as the average value of the square of the molecule displacement x in a time t:

$$< x^2 > = 2Dt \tag{5.3}$$

A dedicated agent monitors the position of all molecules to ensure that every movement ends in an empty space of the environment, avoiding collisions and overlaps.

The simulator enables us to set the space unit and the time scale as per requirement; in this study, we consider the angstrom (Å, equivalent to 10^{-10} m) for space and 10^{-4} seconds for time (corresponding to one tick of the simulation clock). A cube of 1 attolitre (10^{-18} L, having edges of 1000 Å) represents the best option for the aim of our analysis and meets the computational demand of the simulations (Figure 5.5 shows the 3D space visualised through the interface of the simulator).

The model at the basis of the simulator classifies molecules into three types: *enzymes*, *complexes* and *metabolites*. The property that distinguishes enzymes and complexes from metabo-

lites is that the latter can only move while the first two classes of molecules can also act on the environment.

Although molecular movements are modelled based on Brownian diffusion, this study pushes forward the capabilities of the agent paradigm by not limiting molecular interactions to just those allowed by random encounters. Indeed, enzymatic reactions are simulated by exploiting the ability of agents to perceive and interact with one another: each enzyme identifies the cognate metabolites in its proximity thanks to a perception-sphere that it projects on the environment (see Figure 5.2 for a representation of this sphere in the form of the potential interactions that an enzyme can perform).

As better explained in Chapter 6, such an approach is the simulator key feature that allows studying the effects of long-distance interactions among biomolecules. Indeed, the radius of the interaction sphere can be set according to needs, so we were able to test various lengths of perception and the related molecular behaviours.

Every molecular interaction may lead to the formation of a complex, which is modelled as a new agent. If such a complex represents a saturated enzyme, it waits an amount of time corresponding to the reciprocal of its turnover number (k_{cat}) and then releases the final product (or products) of the reaction; otherwise, it moves and acts on the environment to bind the metabolite needed to reach saturation. This modelling approach is based on the construction of an *enzymatic reaction automaton*.

Enzymatic reaction automaton

According to the Michaelis-Menten model of enzyme kinetics, an enzymatic reaction can be represented as

$$E + S \leftrightharpoons ES \rightarrow E + P \tag{5.4}$$

where E is an enzyme, S is its substrate, and P is the product of the reaction catalysed by E; assuming the *steady-state approximation*, we can consider ES as constant [18, 59].

However, by taking into account local interactions in the dynamics of a biochemical reaction, we can abstract the following molecular entities:

- *Free enzyme*, seeking a substrate to interact with.
- *Dual-complex*, formed when an enzyme binds a cognate metabolite but needs another molecule (such as an energy donor) to saturate; it is unstable because the second metabolite is necessary to generate the final products of the reaction.
- *Saturated enzyme*, corresponding to the final complex of the reaction; it is formed by an enzyme linked to one or two metabolites, stably for a time interval obtained as the reciprocal of the k_{cat} of the reaction. The k_{cat} represents the number of molecules converted by an enzyme in the time unit; therefore, its reciprocal provides the interval after which the reaction products are released in the environment, and the enzyme returns free.

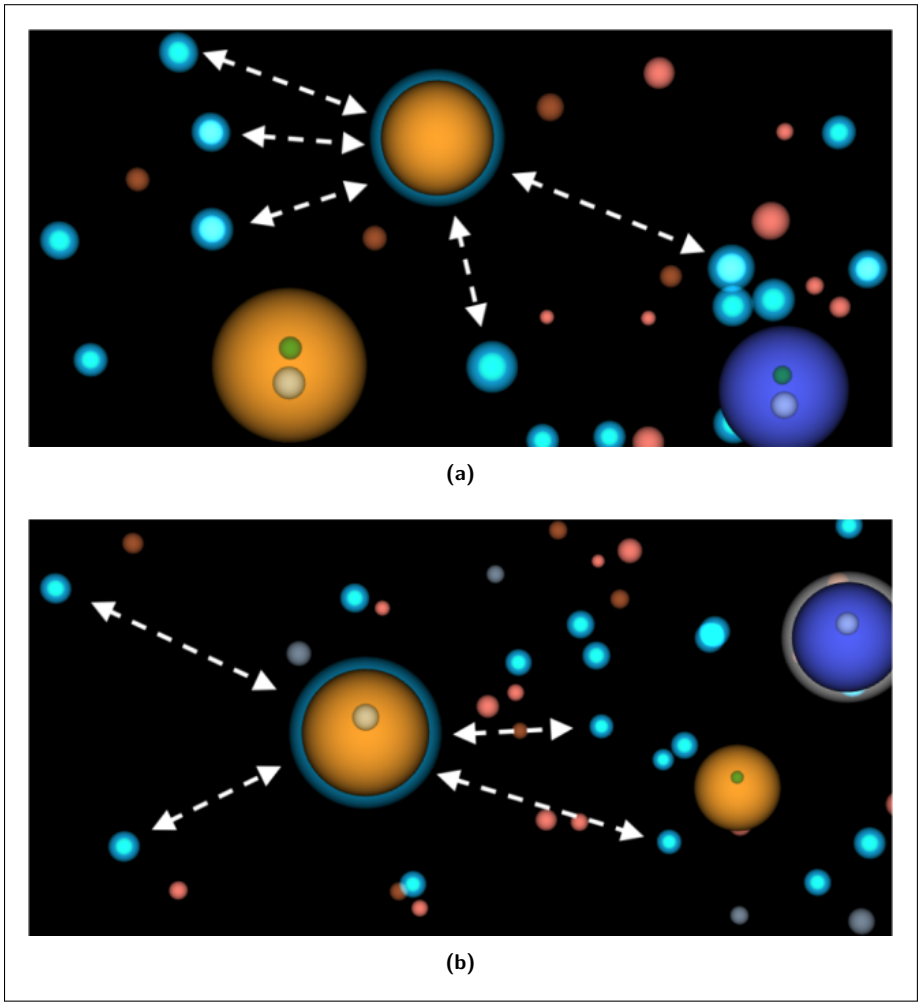


Figure 5.2 – Agents' perception capabilities underlying the molecular interactions in the simulated environment. All molecules are modelled as spheres: the larger ones correspond to enzymes while the others to metabolites. In molecular complexes, the latter are shown as attached to the sphere of the cognate enzyme. This representation has just an illustrative purpose; in the actual implementation, each complex is simulated as a single sphere whose volume is obtained from the sum of the weights of the generating molecules through Equation 5.1. The perception spheres are not explicitly represented to maintain the clarity of the illustration; instead, a perceiving enzyme and the metabolites in its perception volume are highlighted in blue. Figure **(a)** shows the possible interactions of a free enzyme; in **(b)**, a similar situation is depicted for a complex made up of an enzyme with a bound metabolite. The white arrows point out that each interaction in the system is two-body.

An enzymatic reaction cycles through these states; we model such a pattern by constructing an *automaton* based on the molecular entities described above. To reproduce the local interactions properly, we represent each molecule as an autonomous entity; as mentioned in Section 5.3.1, this entity corresponds, in our model of glycolysis, to an *agent*, a system *having the capability of perceiving and interacting with its environment*.

We provide a formal definition of the automaton through Milner's Calculus of Communicating Systems (CCS). This process algebra consists of a collection of constructors for building a new process description from existing ones by representing them as systems that exhibit behaviour and interact via synchronised communication (see Section 1.3.1). The reaction automaton represents an agent-based perspective on the CCS enzymatic reaction model proposed in Definitions 4.3 and 4.4.

Definition 5.1 (Enzymatic reaction automaton)**.** An *enzymatic reaction automaton*, denoted by $\mathcal{R}_{\mathcal{E}}$, is the process whose behaviour is given by the following defining equation:

$$\begin{aligned}
&\mathcal{R}_{\mathcal{E}} \stackrel{\text{def}}{=} \mathtt{e}.\mathtt{E}_{\mathtt{m1}} + \mathtt{e}.\mathtt{E}_{\mathtt{m2}};\\
&\mathtt{E}_{\mathtt{m1}} \stackrel{\text{def}}{=} \mathtt{m1}.\mathtt{DC1} + \mathtt{m1}.\mathtt{ES};\\
&\mathtt{E}_{\mathtt{m2}} \stackrel{\text{def}}{=} \mathtt{m2}.\mathtt{DC2};\\
&\mathtt{DC1} \stackrel{\text{def}}{=} \mathtt{m2}.\mathtt{DC1}_{\mathtt{m2}};\\
&\mathtt{DC2} \stackrel{\text{def}}{=} \mathtt{m1}.\mathtt{DC2}_{\mathtt{m1}};\\
&\mathtt{DC1}_{\mathtt{m2}} \stackrel{\text{def}}{=} \mathtt{m2}.\mathtt{ES};\\
&\mathtt{DC2}_{\mathtt{m1}} \stackrel{\text{def}}{=} \mathtt{m1}.\mathtt{ES};\\
&\mathtt{ES} \stackrel{\text{def}}{=} \overline{\mathtt{pe}}.\mathcal{R}_{\mathcal{E}};
\end{aligned} \tag{5.5}$$

where

- e is a free enzyme;
- m1 is the primary substrate of the enzyme;
- m2 is a secondary substrate of the enzyme, such as an energy donor;
- pe generalises the products of the reaction (one or more) and the enzyme that returns free;
- $\mathtt{E}_{\mathtt{m1}}$ and $\mathtt{E}_{\mathtt{m2}}$ are the states that represent the enzyme perceiving a cognate metabolite;
- DC1 and DC2 correspond to the dual-complexes of the enzyme with m1 and m2 respectively;
- $\mathtt{DC1}_{\mathtt{m2}}$ and $\mathtt{DC2}_{\mathtt{m1}}$ are the states in which the dual complexes perceive the metabolite needed to saturate the enzyme;
- ES represents the saturated enzyme.

To better highlight all the processes and actions characterising the reaction automaton, Figure 5.3 provides the labelled transition system (LTS) [62] related to its algebraic specification.

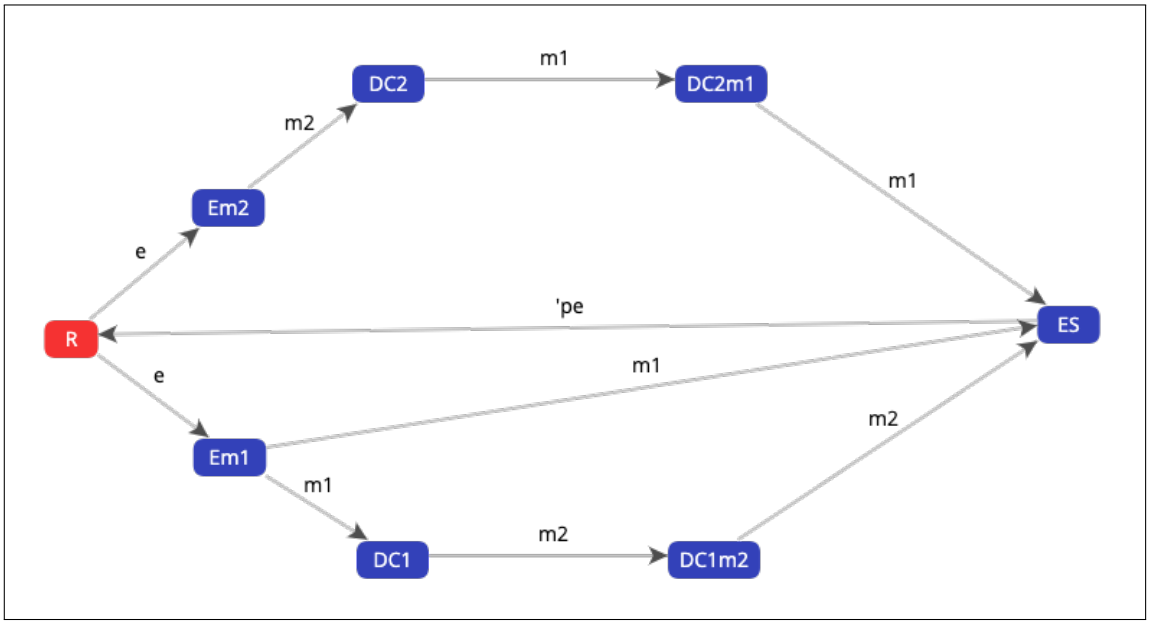


Figure 5.3 – Labelled transition system (LTS) of the automaton representing an enzymatic reaction in our agent-based model. As described in Section 1.3.2, it consists of a set of processes, a set of actions and a transition relation $\rightarrow$ such that, if a process P can perform an action $\mathtt{w}$ and become a process P', we write $\mathsf{P} \xrightarrow{\mathtt{w}} \mathsf{P}'$ [1]. The shown LTS has been generated, from the algebraic definition provided in Section 5.3.1, through the CAAL concurrency workbench [3]. The names of the states are transliterations of the names provided in the CCS specification; the output on the general communication channel $\mathtt{w}$ is denoted by the label $'\mathtt{w}$.

Since the simulator represents the molecules as spheres, we can implement this model by allowing the formation of larger spheres as a result of the interaction between two cognate molecules. The volume of the sphere corresponding to a molecular complex is calculated from the sum of the originating molecules' weights on the basis of Equation 5.1. Figure 5.4 provides a schematic representation of the automaton for the case in which the enzyme interacts with two metabolites.

5.3.2 From a kinetic to an agent-based model

The construction of an agent-based model (ABM) able to represent the molecular interactions of a metabolic pathway requires some information on the pathway itself and on the environment where it occurs. In particular, we need to know the sequence of reactions to simulate–or a subset of those relevant for our analysis–and some quantitative data, such as the initial concentrations of the species involved. In this perspective, a kinetic model can serve as a source of such data and as a reference against which to compare our results.

We cannot completely base our study on a kinetic model, since it uses experimental parameters, often assayed in vitro, to directly describe the global properties of the system through a set of differential equations. Conversely, we aim to understand if kinetic data actually underlie processes related to the ability of molecules to perceive each other, even from a long distance. An ABM of molecular interactions allows not considering a priori most of these parameters and thus provides a better baseline over which to carry out our in silico studies. ABMs describe molecular

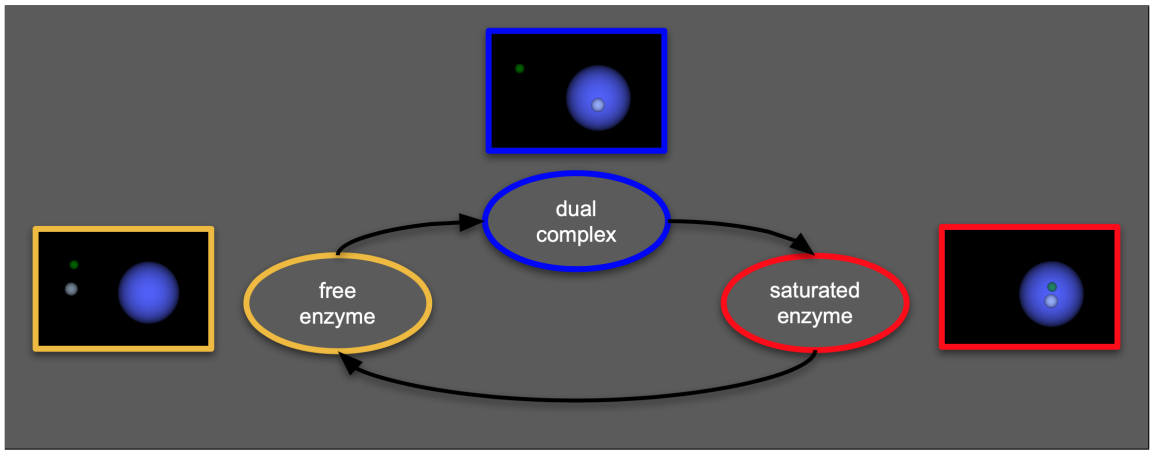


Figure 5.4 – The three states of the enzymatic reaction automaton in which the enzyme interacts with two metabolites (see also Chapter 7 for further details). Each state has been associated with a representation of the related molecular entities in the agent-based model. To better show the molecules involved in the formation of a molecular complex, we choose to draw dual-complexes and saturated enzymes as paired spheres; however, in the actual implementation, each of them is represented by a single sphere whose volume is obtained, from the sum of the weights of the generating molecules, through Equation 5.1.

interactions at a local level, but they also possess *compositionality*, which is the capability of recursively applying the rules characterising agent interactions to progressively define higher abstraction levels. In this way, we can hide the unnecessary details of a specific level and, at the same time, observe its global behaviour [16, 28]. Considering the case of a metabolic pathway, a kinetic model treats enzymatic reactions as mathematical functions that relate the concentrations of reactants to those of products, assuming that they incorporate the role carried out by each molecular interaction. In our ABM, instead, each enzyme is represented by a dedicated agent able to perceive the environment and its cognate partners; the interactions among the molecules are thus explicit in the definition of the model. The compositionality of ABMs also makes it possible to conduct the study at an abstraction level that can be represented with a small amount of empirical data without losing accuracy in reproducing macromolecular behaviours. Nonetheless, not all kinetic parameters can be disregarded. For a modelled saturated enzyme to generate the products of the reaction faithfully to its biological counterpart, it must wait for a time interval corresponding to the reciprocal of its k_{cat} (see Section 5.3.1).

Several kinetic models have already been constructed over metabolic pathways, mainly because the properties of metabolism at a steady state simplify the model definition [120]. However, by considering the enzymatic reactions as just mathematical functions from reactants to products, they mostly focus on changes in metabolite concentrations and do not provide the actual number of enzyme molecules in the simulated environment. In contrast, for the reasons explained above, this information is fundamental for constructing our ABM. Based on this requirement, we identified in the Smallbone2013 - Iteration 18 [107] a model particularly suitable to serve as a source for the ABM because it contains a complete set of experimental data

on the isoenzymes involved in a well-studied metabolic process, the glycolysis of Saccharomyces cerevisiae. The Smallbone2013 model provides a detailed description of the chain of reactions that generates energy from glucose by breaking it into two molecules of pyruvate. In addition to the main branch of glycolysis, it includes the glycerol, glycogen, and trehalose branches; it also considers the alcoholic fermentation steps, which lead to the formation of ethanol (see Figure 5.1).

5.3.3 Defining the input for the simulation

The input of our agent-based simulator (Orion) is a Systems Biology Markup Language (SBML) file filled with experimental data [5, 8, 22, 38, 56, 71]. It contains information about the molecules involved in the metabolic pathway and their initial concentrations; data related to the reactions carried out are also taken from this source. The Smallbone2013 model of glycolysis mentioned above is provided in the SBML format (`http://identifiers.org/biomodels.db/MODEL1303260018`).

A dedicated simulator component converts the SBML model to an Extensible Markup Language (XML) file specifically formatted to be interpreted by the simulator while remaining human-readable [122]. Therefore, its primary function is to translate the kinetic representation of the metabolic reactions into our agent-based model. To accomplish this task, for every reaction we want to model, it gets from the SBML file the reactants and products and generates XML code for each of its interactions, based on the algebraic definition of the automaton provided in Section 5.3.1. It also associates the k_{cat} to the related reaction and the K_m values to all its enzyme-substrate interactions. The K_m measures the affinity of an enzyme for a specific substrate; it is needed since an enzyme can form a complex with an encountered metabolite randomly or based on a priority list constructed over the k_{cat}/K_m ratio (specificity constant). This possibility can be established in the initial setup of the simulation.

SBML and XML (from which the first is derived) are markup languages that define rules for storing data in a formatted document to comply with both human and machine readability [56, 122]. They are structured as element trees: starting from a root, each element of the tree can have one or more child elements. Every element is delimited by an opening tag, in which the element name is enclosed between angle brackets (< and >), e.g. `<element_name>`, and a closing tag, similar to the opening tag but with the element name preceded by a slash symbol (/), e.g. `</element_name>`. It can also have one or more attributes, placed inside the opening tag in the form `attribute_name = "attribute_value"`.

In what follows, we provide a simplified conversion of a kinetic model, in SBML format, to the XML input of our agent-based simulator; we consider a generalised reaction that is catalysed by an enzyme `E`, with two substrate metabolites (`M1` and `M2`) and two products (`P1` and `P2`).

The conversion starts from the following SMBL source:

```
<reaction metaid="meta_E" sboTerm="SBO:0000176" id="E" name="
   reaction_name">
  <annotation>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
   xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
    <rdf:Description rdf:about="#meta_E">
     <bqbiol:is>
      <rdf:Bag>
       <rdf:li rdf:resource="resource_url"/>
      </rdf:Bag>
     </bqbiol:is>
     <bqbiol:isVersionOf>
      <rdf:Bag>
       <rdf:li rdf:resource="identifier_url"/>
       <rdf:li rdf:resource="identifier_url"/>
      </rdf:Bag>
     </bqbiol:isVersionOf>
    </rdf:Description>
   </rdf:RDF>
  </annotation>
  <listOfReactants>
   <speciesReference metaid="metaid_value" species="M1"/>
   <speciesReference metaid="metaid_value" species="M2"/>
  </listOfReactants>
  <listOfProducts>
   <speciesReference metaid="metaid_value" species="P1"/>
   <speciesReference metaid="metaid_value" species="P2"/>
  </listOfProducts>
  <listOfModifiers>
   <modifierSpeciesReference metaid="metaid_value" species="E"/>
   <modifierSpeciesReference species="P1"/>
   <modifierSpeciesReference species="P2"/>
   <modifierSpeciesReference species="E"/>
  </listOfModifiers>
  <listOfParameters>
   <parameter metaid="metaid_value" id="kcat" value="kcat_value"
   units="per_second"/>
   <parameter metaid="metaid_value" id="Km1" value="Km1_value" units
   ="mM"/>
   <parameter metaid="metaid_value" id="Km2" value="Km2_value" units
   ="mM"/>
  </listOfParameters>
```

From this SBML source, the simulator's conversion component generates the XML code:

```
<reaction>
<interaction>
 <reactants>
  <reactant id="E"/>
  <reactant id="M1"/>
 </reactants>
 <products>
  <product id="E+M1"/>
 </products>
 <Km unit="mM">Km1_value</Km>
</interaction>
<interaction>
 <reactants>
  <reactant id="E"/>
  <reactant id="M2"/>
 </reactants>
 <products>
  <product id="E+M2"/>
 </products>
 <Km unit="mM">Km2_value</Km>
</interaction>
<interaction>
 <reactants>
  <reactant id="E+M1"/>
  <reactant id="M2"/>
 </reactants>
 <products>
  <product id="E+M1+M2"/>
 </products>
 <Km unit="mM">Km2_value</Km>
</interaction>
<interaction>
 <reactants>
  <reactant id="E+M2"/>
  <reactant id="M1"/>
 </reactants>
 <products>
  <product id="E+M1+M2"/>
 </products>
 <Km unit="mM">Km1_value</Km>
</interaction>
<interaction>
 <reactants>
  <reactant id="E+M1+M2"/>
 </reactants>
```

```
    <products>
     <product id="P1"/>
     <product id="P2"/>
     <product id="E"/>
    </products>
    <Km unit="mM">0.0</Km>
   </interaction>
   <kcat unit="per_second">kcat_value</kcat>
  </reaction>
```

where `E+M1` and `E+M2` are dual complexes (the states `DC1` and `DC2` of the Equation 5.5), while `E+M1+M2` represents the saturated enzyme. The K_m of the last interaction is always 0 because this process models the release of the reaction products after the time interval given by the k_{cat} reciprocal.

The conversion component of Orion 2.0.0 also retrieves from online databases, specifically ChEBI [52] and UniProt [114], the molecular weights–needed for the simulation but missing in the SBML model. These values are required to obtain the volumes of the spheres that represent the molecules of the simulation; they are calculated through Equation 5.1.

The XML file is structured in four main parts:

1. unit definitions;
2. list of metabolites and enzymes in the modelled cytoplasm portion at the beginning of the simulation (along with the list of the complexes that may form during each enzymatic reaction);
3. list of all the reactions that may occur in the metabolic pathway;
4. ambient and interaction properties of the simulation.

The first three parts are composed of the children of the element "`pathway`" because they, indeed, set the properties of the modelled metabolic pathway.

The *unit definitions* provide a list of all the derived units adopted in the model and specify how they are obtained from the base units of the International System of Units (SI). For example, the definition of the unit *millimolar* (mM or mmol/l) is given by the following XML code:

```
<unitDefinition id="mM">
  <unit exponent="1" kind="mole" multiplier="1" scale="-3"/>
  <unit exponent="-1" kind="litre" multiplier="1" scale="0"/>
</unitDefinition>
```

The *list of molecules* part reports all the metabolites, enzymes, and complexes involved in the modelled metabolic process. It also stores the initial concentrations and the molecular weights (which the simulator retrieves from online databases, as explained before). For molecular complexes, the initial concentrations are always zero, while their molecular weights are calculated as the sum of the weights of the molecules that compose each of them. As an example of XML code for a metabolite, `glucose` (`GLC`) is defined as follows:

```
<!--metabolite - ChEBI name: D-glucopyranose; resource: http://
    identifiers.org/chebi/CHEBI:4167 -->
<molecule compartment="cell" id="GLC" name="glucose" type="
    Metabolite">
  <molecularWeight unit="dalton">180.06</molecularWeight>
  <initialConcentration unit="mM">0.6280001793382419</
    initialConcentration>
</molecule>
```

The `compartment` is reported as an attribute for allowing the possibility of modelling membrane transport (not implemented in the simulations described in this manuscript); `name` and `id` are taken from the SBML model, but the database name is also indicated in the comment (the first line, delimited by `<!--` and `-->`), along with the link to the online database record.

A similar approach is used to define enzymes and complexes. To improve the file's readability, the simulator groups each enzyme definition with those of the complexes that such an enzyme can form. For example, considering the isoenzyme `hexokinase-1` (`HXK1`), since it can interact with `glucose` and `ATP`, this enzyme and the related complexes are listed in the following way:

```
<!--Reaction: Hexokinase [HXK1]-->
<!--enzyme - UniProt name: Hexokinase-1; resource: https://www.
    uniprot.org/uniprot/P04806 -->
<molecule compartment="cell" id="HXK1" name="HXK1" type="Enzyme">
  <molecularWeight unit="dalton">53738.0</molecularWeight>
  <initialConcentration unit="mM">0.0167807457149784</
    initialConcentration>
</molecule>
<molecule compartment="cell" id="HXK1+GLC" name="HXK1+glucose" type=
    "Complex">
  <molecularWeight unit="dalton">53918.06</molecularWeight>
  <initialConcentration unit="mM">0</initialConcentration>
</molecule>
<molecule compartment="cell" id="HXK1+ATP" name="HXK1+ATP" type="
    Complex">
  <molecularWeight unit="dalton">54245.0</molecularWeight>
  <initialConcentration unit="mM">0</initialConcentration>
</molecule>
<molecule compartment="cell" id="HXK1+GLC+ATP" name="HXK1+glucose+
    ATP" type="Complex">
  <molecularWeight unit="dalton">54425.06</molecularWeight>
  <initialConcentration unit="mM">0</initialConcentration>
</molecule>
```

The *list of reactions* is in the form already described in this section; for completeness, we show, instead of just a generalisation, the reaction catalysed by `HXK1`, that is:

$$\mathtt{GLC} + \mathtt{ATP} \xrightarrow{\mathtt{HXK1}} \mathtt{G6P} + \mathtt{ADP}$$

The XML code generated for this reaction is the following:

```
<!--Hexokinase [HXK1]: irreversible reaction - forward direction-->
   <reaction>
    <interaction>
     <reactants>
      <reactant id="HXK1"/>
      <reactant id="GLC"/>
     </reactants>
     <products>
      <product id="HXK1+GLC"/>
     </products>
     <Km unit="mM">0.15</Km>
    </interaction>
    <interaction>
     <reactants>
      <reactant id="HXK1"/>
      <reactant id="ATP"/>
     </reactants>
     <products>
      <product id="HXK1+ATP"/>
     </products>
     <Km unit="mM">0.293</Km>
    </interaction>
    <interaction>
     <reactants>
      <reactant id="HXK1+GLC"/>
      <reactant id="ATP"/>
     </reactants>
     <products>
      <product id="HXK1+GLC+ATP"/>
     </products>
     <Km unit="mM">0.293</Km>
    </interaction>
    <interaction>
     <reactants>
      <reactant id="HXK1+ATP"/>
      <reactant id="GLC"/>
     </reactants>
     <products>
      <product id="HXK1+GLC+ATP"/>
     </products>
     <Km unit="mM">0.15</Km>
    </interaction>
    <interaction>
     <reactants>
      <reactant id="HXK1+GLC+ATP"/>
     </reactants>
```

```
      <products>
       <product id="G6P"/>
       <product id="ADP"/>
       <product id="HXK1"/>
      </products>
      <Km unit="mM">0.0</Km>
     </interaction>
     <kcat unit="per_second">10.2</kcat>
    </reaction>
```

The fourth part of the XML file is placed outside the "pathway" element; it specifies the physical parameters of the modelled cytoplasm portion that are needed to reproduce Brownian motion (see Section 5.3.1), as well as the properties of molecular interactions. Specifically, it lists:

- volume of the cytoplasm portion (in attoliters);
- viscosity of the environment (in pascal-seconds);
- temperature (in Kelvin degrees);
- perception distance of the active molecules (in angstroms);
- the possibility or not for enzymes to prioritise interactions based on the specificity constant (k_{cat}/K_m ratio, as previously explained in this section).

An example of configuration is the following:

```
<!--Ambient settings-->
 <ambientSettings>
  <volumeOfSimulation unit="attolitre">1</volumeOfSimulation>
  <viscosity unit="pascal_second">0.0011</viscosity>
  <temperature unit="kelvin">298.15</temperature>
 </ambientSettings>
<!--Interaction settings-->
 <interactionSettings>
  <perceptionDistance unit="angstrom">300</perceptionDistance>
  <priorityBySpecificity>false</priorityBySpecificity>
 </interactionSettings>
```

The value of the perception distance is widely discussed in Chapter 6, while the `priorityBySpecificity` value is set according to the aim of the study carried out (it is set to `false` in Chapter 6, while it is required to be `true` for the work proposed in Chapter 7). The ambient values of the example are those we set for all the simulations provided in this manuscript.

5.3.4 Simulation output and visualisation

The output of the simulator is a set of Comma-Separated Values (CSV) files reporting the type and number of molecules contained in the simulated environment, along with their position, at each instant of simulation. More precisely, the simulator generates three types of CSV files:

- A *standard output*, reporting type (metabolite, enzyme or complex) and concentration (in mmol/l) of every molecular species in the modelled cytoplasm portion at each instant of the simulation. Such files are used to generate the plots provided in Chapter 6 and related Appendix C.
- A *"verbose" output*, which, differently from the previous one, lists each molecule in the environment at every instant of the simulation, its radius (in picometres), and its position coordinates in the three-dimensional space. This file is necessary to restart an interrupted simulation and generate a 3D representation of the simulated environment; the latter is shown through a dedicated interface of the simulator.
- An *"interactors" output*, containing, for each simulation time step, the enzymes perceiving one or more metabolites and a list of these molecules; such a file is needed to highlight enzymes' perception in the 3D interface. The generation of this file is optional and can be excluded to reduce the computational demand of the simulation.

In Figure 5.5, we provide a screenshot of the simulated environment after 5.9 ms of simulation and the related plots of the concentration changes generated through the simulator interface.

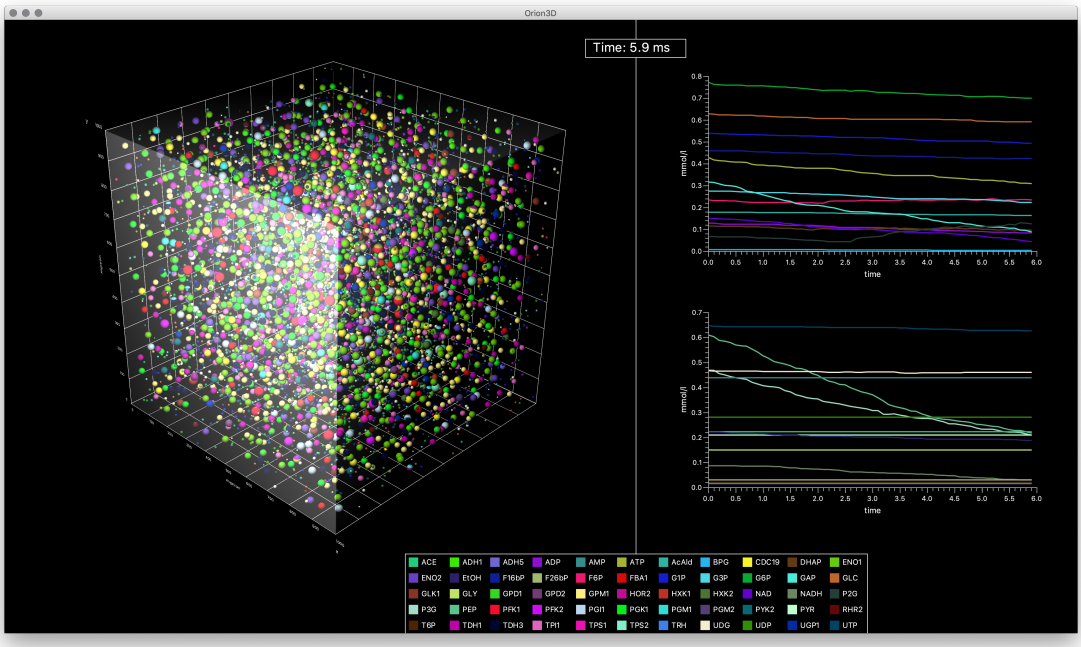


Figure 5.5 – 3D interface of the agent-based simulator. The cube representing the volume of the simulation has edges of 1000 Å. The interface shows the position of every molecule instant by instant. It is also possible to highlight the metabolites perceived by each enzyme at a specific time step of the simulation. On the right, the plots of the species concentration changes over time are generated for the first 5.9 ms of simulation. At the bottom of the interface, a legend associates each molecule with its corresponding colour.

Chapter 6

Detecting In Silico the Driving Forces of Biomolecular Interactions

6.1 Introduction

Long-distance electrodynamic interactions between two small molecules have been primarily studied within the framework of quantum electrodynamics since long-range forces can be detected among excited atoms with similar transition frequencies [72, 109]. However, interactions beyond the Debye screening length ($\simeq 10$ Å in biological systems [26]), carried out by the molecular cognate partners of a biochemical reaction, are not well investigated. Nonetheless, experimental evidence for collective excitations of biological macromolecules is available in the Raman and far-infrared spectroscopic domains [36, 85]. Electrodynamic interactions occurring between oscillating electric dipoles might have a long-range nature; deterministic selective forces can thus be activated at a distance when the molecules undergo coherent collective oscillations [95]. The existence of forces of this kind might justify the efficiency of biochemical reactions more than the sole effect of stochastic short-range interactions, which rely just on Brownian diffusion and chemical affinity. Numerical studies proved that the overall interaction potential $U(\vec{r})$ between cognate partners (with r being the intermolecular distance) is generally composed of a short-range term (r^{-6}) and a resonant long-range term (r^{-3}); this means that, when the dipole moments of two molecules oscillate at the same frequency, an attractive resonant potential $U(r) \propto -1/r^3$ should be added to the random Brownian force [94].

These phenomena have been lately analysed, theoretically and experimentally, in the interactions among lysozyme molecules and oppositely charged dyes [81]. However, detecting long-range molecular recruitments in biosystems is still held back by the current technology; even these recent results, gained through fluorescence correlation spectroscopy, are limited to systems where the long-range interactions are built-in (by setting up a solution in which the electrostatic interactions are non-screened).

Computational approaches might overcome some of these hurdles, allowing to test in silico the existing theoretical models. Indeed, numerical simulations, such as those performed through molecular dynamics, have been successfully carried out [81], considering an a priori knowledge of numerous physical parameters characterising the molecular interactions under study. A large amount of empirical information allows for generating a faithful representation of the biological system and provides a reliable in silico support for theoretical and experimental analyses; however, a lack of empirical data may limit the complexity of the system simulated.

This chapter aims to address most of these issues by exploiting an alternative way to define a computational model of molecular interactions in a metabolic pathway. Specifically, we construct an agent-based model (ABM) of a well-studied process, the glycolysis of yeasts, to simulate the effect of the long-distance electrodynamic interactions among the biomolecules involved in the pathway. The agent-based simulator we use for this study is Orion 2.0.0, introduced in Section 5.3.1, which makes use of autonomous software pieces (agents) capable of interacting with one another concurrently and asynchronously (see also Section 1.3.5); they can thus fairly faithfully replicate in silico the behaviour of the entities interacting in a real biological system. ABMs require instructing the agents representing the simulated molecules with minimal empirical information, letting the global behaviour of the process result from local interactions generated dynamically at each step of the simulation. The system evolves due to the ability of every agent to perceive and respond to the states of its environment, which is unpredictable and populated by other agents; the agent's perception results in performing an appropriate action (if any) able to modify the environment [42]. The agent-based approach allows both the environment and the molecules to be three-dimensional (as shown in Figure 5.5 on page 101); molecular shapes can thus affect the diffusion processes.

ABMs have been already successfully applied in the analysis of several biological systems and used to develop tools for in silico supporting experimental studies [11, 23, 84]. With the present work, we leverage the flexibility of the agent-based modelling to construct in silico biochemical systems; this approach is intended to simulate the glycolytic process by considering different types of forces driving molecular interactions. We aim to abstract the core features of biochemical systems characterised by purely random molecular encounters and compare them to those where cognate partners' interactions are mainly driven by deterministic long-range forces. ABMs allow us to reproduce these phenomena in a network of mutually conditioning reactions without knowing a priori all the parameters needed in a numerical simulation, which might be missing or difficult to assay experimentally. For this reason, we simulated the molecular interactions as entirely random, without predetermining any priority on the metabolites perceived by an enzyme (that is, ignoring the effects of the specificity constant in the initial setup of the simulation, as described in Section 5.3.3).

By analysing the concentration changes of the molecular species during each step of the agent-based simulation, we can hypothesise how long-distance interactions may quantitatively and qualitatively affect the glycolysis process. This way, we can also hint at what might be the physical phenomena underlying the related kinetic parameters if they were assayed in vivo and highlight possible discrepancies with the values obtained in vitro. These results would provide the basis for setting up further experimental studies.

6.2 Additional Methods

6.2.1 Designing the model of glycolysis

As explained in Section 5.3.2, we chose the Smallbone2013 - Iteration 18 model [107], provided in the SBML format (http://identifiers.org/biomodels.db/MODEL1303260018), as the data source for the simulations of this dissertation.

By importing the reactions of the SMBL file as input for our agent-based simulations, we excluded all those for which the Smallbone2013 model does not provide enzymatic concentrations. Our simulator can handle these kinds of reactions since we can model them in terms of their bulk effects. However, introducing any bulk reaction would perturb the environment and hide the absence of actual interactions among the molecules; this would make the ABM close to a standard kinetic model and compromise the possibility of observing the global behaviour of glycolysis as resulting from the local molecular interactions. Based on this idea, we do not consider the adenylate kinase reaction, the ATPase reactions, the UDP to UTP reaction, and the glucose transport (between the extracellular environment and the cytosol). The most significant of these reactions is the adenylate kinase since it controls the ratio of ATP, ADP, and AMP (also called energy charge), which in turn affects the allosteric regulation of important enzymes, such as phosphofructokinase and hexokinase [48]. However, the length of the simulated process (1 second, as discussed in Section 6.3) makes the allosteric regulation and the whole energy charge effects negligible [64, 116]. Suppressing glucose transport and enzyme regulation also prevents, de facto, the achievement of a steady state, helping us to emphasise the effects of the various types of interactions on the concentration changes in the simulation interval.

The initial concentrations of the molecular species are gained from the SBML file as millimoles per litre (mmol/l). A dedicated simulator component converts these values into the initial particle numbers needed to instantiate the agents at the beginning of the simulation. In this regard, we point out that, although agent-based simulations have a fairly light computational load, reproducing a metabolic pathway involves thousands of molecules, therefore as many agents running concurrently. The resulting resources demand conditioned the molecular concentrations we were able to simulate. More precisely, we scaled the concentrations provided by the Smallbone2013 model to values less than 1 mmol/l. In Table 6.1, we report the initial concentrations of all the simulated species. The total number of molecules (enzymes and metabolites) in the environment at the beginning of the simulation is 6955.

Our agent-based model is intended to study the glycolytic pathway from the general perspective of the oxidation of one molecule of glucose to two molecules of pyruvate; for this reason, we consider the pyruvate as the end product of the process and excluded the fermentation-related reactions, catalysed by the pyruvate decarboxylase isoenzymes (PDC1, PDC5, PDC6) and by the two alcohol dehydrogenase isoenzymes (ADH1 and ADH5). Therefore, the branches acting on pyruvate, that is, the succinate and acetate branches of glycolysis, are not taken into account in our model.

Table 6.1 – Initial concentrations and sphere radii of the molecular species simulated in our study. The original amounts provided by the Smallbone2013 - Iteration 18 model have been scaled to values less than 1 mmol/l to fit the computational demand of the simulations. Each radius is obtained from the volume calculated through Equation 5.1.

Metabolites

Name	Initial Conc. (mmol/l)	Sphere Radius (Å)
ADP	0.129	4.98
AMP	0.44	4.65
ATP	0.429	5.28
BPG	0.007	4.26
DHAP	0.116	3.67
F16bP	0.458	4.62
F26bP	0.030	4.62
F6P	0.235	4.22
G1P	0.539	4.22
G3P	0.274	3.68
G6P	0.772	4.22
GAP	0.316	3.67
GLC	0.628	3.74
GLY	0.150	2.99
NAD	0.150	5.77
NADH	0.086	5.78
P2G	0.068	3.78
P3G	0.470	3.78
PEP	0.610	3.65
PYR	0.211	2.93
T6P	0.020	4.96
TRH	0.015	4.63
UDG	0.467	5.47
UDP	0.282	4.89
UTP	0.649	5.18

Enzymes

Name	Initial Conc. (mmol/l)	Sphere Radius (Å)
CDC19	0.205	25.07
ENO1	0.686	23.83
ENO2	0.197	23.85
FBA1	0.134	22.54
GLK1	0.045	25.2
GPD1	0.068	23.14
GPD2	0.008	24.26
GPM1	0.730	19.98
HOR2	0.055	20.03
HXK1	0.017	24.95
HXK2	0.061	24.98
PFK1	0.047	31.48
PFK2	0.039	31.15
PGI1	0.138	26.07
PGK1	0.258	23.47
PGM1	0.033	26.32
PGM2	0.013	26.32
PYK2	0.061	25.17
RHR2	0.051	20.07
TDH1	0.351	21.78
TDH3	0.420	21.78
TPI1	0.294	19.79
TPS1	0.034	25.32
TPS2	0.027	30.99
UGP1	0.062	25.29

To complete the list of changes we made to the original kinetic model, we report that, according to most of the literature, we modelled the reactions catalysed by `hexokinase` (and `glucokinase`), `phosphofructokinase`, and `pyruvate kinase` as irreversible [12, 30, 60], since they function as control points of the whole glycolysis process, despite the Smallbone2013 model considers irreversible only the reaction performed by `phosphofructokinase`.

The subset of reactions characterising the model at the basis of our simulations, as resulting from the above-described adaptations, can be found in Figure 6.1 and Table 6.2.

6.2.2 Modelling short- and long-range forces among biomolecules

To simulate the effects of the molecular interactions operating at different distances, we endowed agents with specifically designed perception capabilities. Their core property lies in the definition of a *perception sphere* that surrounds each active molecule (enzymes and complexes, as explained in Section 5.3.1). By setting the *perception radius*, that is, the radius of the perception sphere, we can model various lengths at which enzymes and complexes can interact with their cognate partners. Therefore, *the capability of agents to perceive and interact with one another allows us to abstract the effects of the electrostatic and electrodynamic potentials among the molecules of the simulated environment*; this can be achieved without taking into account all the physical parameters usually required in molecular dynamics simulations (such as the potential values or the forces generated by molecular collisions) [81].

Each perception radius is obtained by summing the radius of the enzyme to the *perception distance* at which we want the enzyme to be able to find a cognate metabolite; the perception distance extends beyond the surface of the sphere representing the enzyme. As the distance of the metabolite from the enzyme increases, the intensity of the forces acting on it diminishes; for this reason, each perception sphere is characterised by different interaction probabilities, depending on its size.

We simulated three different systems in which the interactions characterising the glycolytic process are driven by the specific kinds of forces whose effects on the pathway we aim to compare. Going into detail, the agent-based modelling approach makes us able to define:

- A system in which molecular encounters are driven only by Brownian motion and dynamic complementarities (e.g., lock-and-key or induced-fit phenomena). We modelled this system allowing enzymes and complexes to identify a cognate metabolite within a *perception distance of 5 Å*; this sets the space on which electrostatic forces, such as those resulting from van der Waals-like potentials, operate. When a metabolite enters the related sphere (of a cognate enzyme), there is a probability $p = 1$ that the interaction will happen.
- A system where a *10 Å perception distance* models the effects of electromagnetic potentials limited by the Debye screening [26]; it restricts the interactions to just those allowed by stochastic short-range forces. In this case, the probability of the interaction is still $p = 1$ when the metabolite is, at most, 5 Å far from the enzyme sphere; it reduces to $p = 1/2$ when the metabolite is detected at a distance d such that $5 < d \leq 10$ angstroms.
- A system characterised by *perception distances of 300 Å*, chosen as the average length to simulate the existence of long-range electrodynamic forces among biomolecules (considering that the size of the simulation volume of our study is 1000 cubic angstroms). As mentioned in the Introduction, these are deterministic attractive forces activated by a long-range potential between two dipolar molecules, A and B, if they vibrate at frequencies $\omega_A \simeq \omega_B$ (that is, if they are at resonance). In real cells, this phenomenon might be observed because a macromolecule oscillating at a high frequency (in the range of $10^{10} - 10^{11}$ Hz) does not suffer the Debye screening effect by the ions of the medium [94, 95].

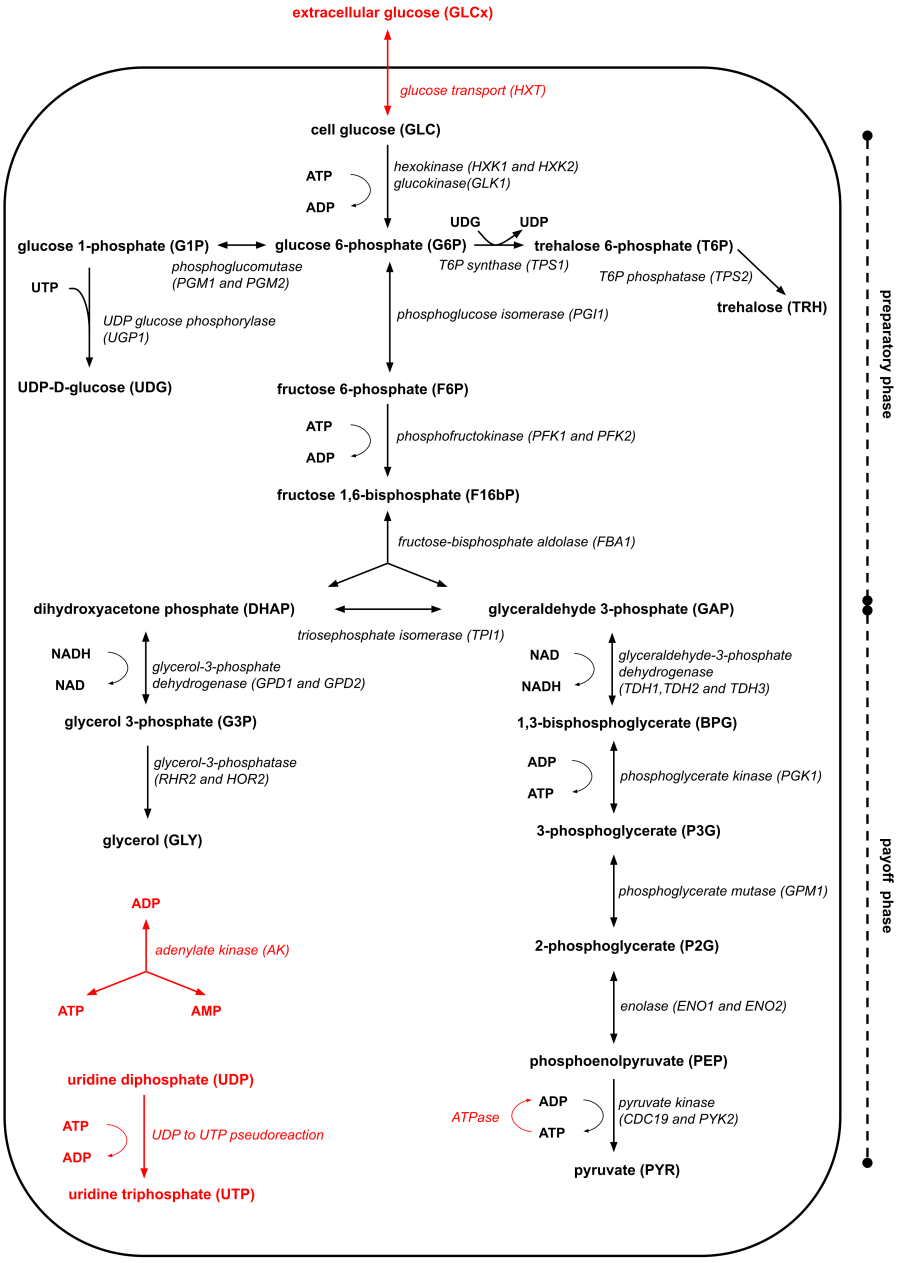


Figure 6.1 – Schematic representation of the glycolysis steps and branches taken into account in the ABMs of this chapter. They are extracted and adapted, through a dedicated component of Orion, from the SBML of the Smallbone2013 kinetic model [107]. The reactions in red are those excluded during the conversion (see the Section 6.2.1 for details). For each metabolite involved, we report both the name and the acronym (in bold), while, for every reaction, we indicate the abbreviation of each isoenzyme carrying it out (in italics). On the right side of the image, we highlight the two main phases of the process; since the ethanol fermentation has not been simulated, we prefer not to show this phase to preserve the readability of the figure.

Table 6.2 – Table of the reactions gained from the Smallbone2103 - Iteration 18 model to define the ABM underlying our simulations. They represent a subset of all the Smallbone2013 reactions, specifically those for which the enzymatic concentration is provided and those not involved in the transformation of pyruvate to ethanol. The reactions are shown in alphabetic order; the related k_{cat} values are also reported.

Reaction name	**Chemical equations**	$\mathbf{k_{cat}}$ $(\mathbf{s^{-1}})$
3-phosphoglycerate kinase	ADP + BPG $\overset{\text{PGK1}}{\rightleftharpoons}$ ATP + P3G	58.6
enolase	P2G $\overset{\text{ENO1}}{\rightleftharpoons}$ PEP	7.6
	P2G $\overset{\text{ENO2}}{\rightleftharpoons}$ PEP	19.87
fructosebisphosphate aldolase	F16bP $\overset{\text{FBA1}}{\rightleftharpoons}$ DHAP + GAP	4.14
glyceraldehyde phosphate dehydrogenase	GAP + NAD $\overset{\text{TDH1}}{\rightleftharpoons}$ BPG + NADH	19.12
	GAP + NAD $\overset{\text{TDH2}}{\rightleftharpoons}$ BPG + NADH	8.63
	GAP + NAD $\overset{\text{TDH3}}{\rightleftharpoons}$ BPG + NADH	18.16
glycerol 3-phosphatase	G3P $\overset{\text{HOR2}}{\longrightarrow}$ GLY	161.38
	G3P $\overset{\text{RHR2}}{\longrightarrow}$ GLY	17.26
glycerol 3-phosphate dehydrogenase	DHAP + NADH $\overset{\text{GPD1}}{\rightleftharpoons}$ G3P + NAD	114.6
	DHAP + NADH $\overset{\text{GPD2}}{\rightleftharpoons}$ G3P + NAD	987.3
hexokinase	GLC + ATP $\overset{\text{HXK1}}{\longrightarrow}$ G6P + ADP	10.2
	GLC + ATP $\overset{\text{HXK2}}{\longrightarrow}$ G6P + ADP	63.1
	GLC + ATP $\overset{\text{GLK1}}{\longrightarrow}$ G6P + ADP	0.07
phosphofructokinase	ATP + F6P $\overset{\text{PFK1}}{\longrightarrow}$ ADP + F16bP	209.6
	ATP + F6P $\overset{\text{PFK2}}{\longrightarrow}$ ADP + F16bP	209.6
phosphoglucomutase	G6P $\overset{\text{PGM1}}{\rightleftharpoons}$ G1P	39.12
	G6P $\overset{\text{PGM2}}{\rightleftharpoons}$ G1P	101.39
phosphoglucose isomerase	G6P $\overset{\text{PGI1}}{\rightleftharpoons}$ F6P	487.36
phosphoglyceromutase	P3G $\overset{\text{GPM1}}{\rightleftharpoons}$ P2G	400
pyruvate kinase	ADP + PEP $\overset{\text{CDC19}}{\longrightarrow}$ ATP + PYR	20.15
	ADP + PEP $\overset{\text{PYK2}}{\longrightarrow}$ ATP + PYR	0
T6P synthase	G6P + UDG $\overset{\text{TPS1}}{\longrightarrow}$ T6P + UDP	145.49
T6P phosphatase	T6P $\overset{\text{TPS2}}{\longrightarrow}$ TRH	879.75
triosephosphate isomerase	DHAP $\overset{\text{TPI1}}{\rightleftharpoons}$ GAP	564.38
UDP glucose phosphorylase	G1P + UTP $\overset{\text{UGP1}}{\longrightarrow}$ UDG	2137.21

A perception sphere of this size is modelled with four different interaction probability intervals. Specifically, let p be the probability of interaction, d_{per} the perception distance, and d_m the distance of the metabolite from the centre of the sphere representing the perceiving enzyme (all the lengths expressed in angstroms):

- if $d_m \leq \frac{1}{4}\, d_{per}$, then $p = 1$
- if $\frac{1}{4}\, d_{per} < d_m \leq \frac{3}{4}\, d_{per}$, then $p = \frac{3}{4}$
- if $\frac{3}{4}\, d_{per} < d_m \leq d_{per}$, then $p = \frac{1}{2}$

This modelling approach turned out to be a reasonable abstraction to represent the progressive reduction of the attraction strength exerted by the enzyme on a cognate metabolite as the distance between the two molecules increases. In Figure 6.2, we provide a graphical representation of how the perception radii project on the environment.

6.3 Results

By setting the local rules that determine the movements and interactions of the molecules involved in our model of yeast glycolysis (as detailed in Section 5.3.1), the global behaviour of the pathway can be observed during the simulation in the form of molecular concentration changes (mmol/l) over time (s).

To balance the computational demand of dealing with thousands of molecules and the need to produce worthwhile outputs, we ran each type of simulation for an interval of 1 second (about ten days of actual running time); it turned out to be sufficiently long for us to observe and compare the specific features of each of the three modelled systems.

In Figure 6.3, we report some of the concentration changes that characterise each type of system. For generating these plots, we selected the metabolites whose amount variations during the simulation were most meaningful for our analysis (a complete set of plots covering all the metabolite species considered in our models is provided in Appendix Section C.1.1).

The simulation performed by setting *perception distances of 300 Å*, which represents a system where we hypothesise the existence of selective long-range molecular recruitments, has the highest reactivity and efficiency (Figure 6.3a); already after 0.9 s, all the `glucose` in the environment is consumed, and the `pyruvate` (one of the main products of the pathway) increases from an initial concentration of 0.2 mmol/l to about 1 mmol/l.

In the system where we limited the electromagnetic forces to those below the Debye screening (*perception distances of 10 Å*–Figure 6.3b), we do not observe utterly different concentration changes in comparison to the previous one; however, they clearly show a lower efficiency in the production of `pyruvate` from glucose, which a system of that type is unable to deplete completely in the chosen simulation interval.

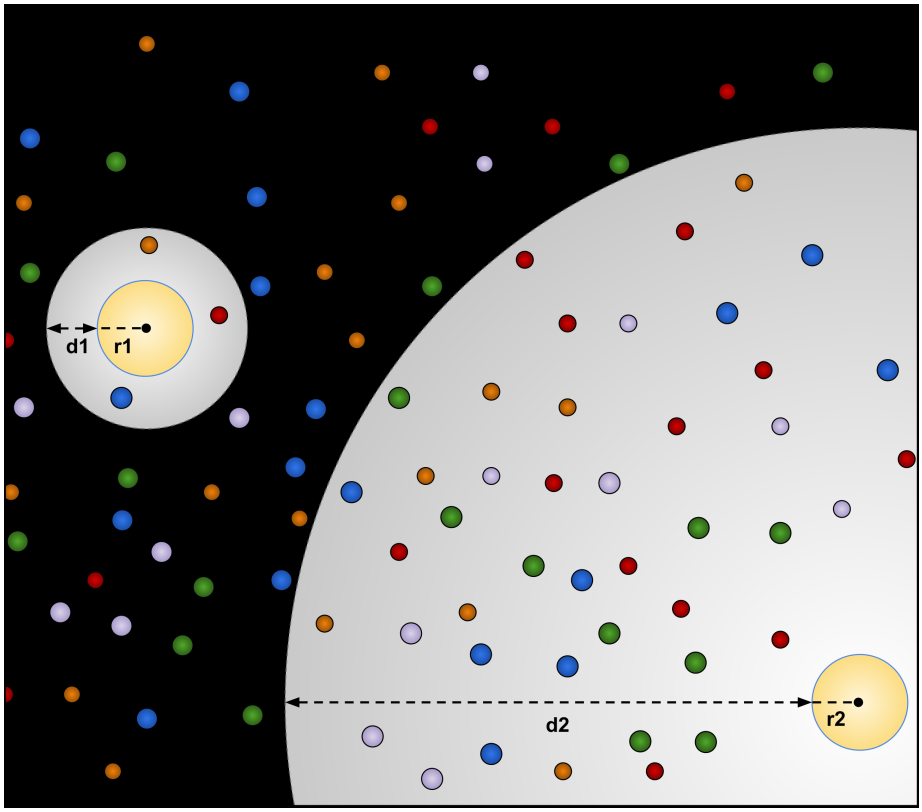


Figure 6.2 – Graphical representation of the agent's perception, by which every modelled enzyme detects the cognate metabolites in its surrounding environment. Each enzyme, depicted as a sphere of radius `r`, is able to perceive its neighbouring metabolites at different distances `d`. This process is fundamental for reproducing in silico the effects of the long-range forces on biochemical reactions, as discussed throughout this chapter. Each *perception radius* is given as the sum of the enzyme radius (`r`) and the *perception distance* (`d`) at which the molecule can detect its cognate metabolites. The perception radius of the `r1+d1` type schematises the constraint that limits the enzyme interactions to those allowed by short-range forces (both *5 and 10 Å perception distances*), while a `r2+d2` type radius models the effects of the long-distance electrodynamic interactions. We point out that the figure arranges side by side two different types of radii just for comparative purposes; in the ABMs defined for glycolysis, *only one type of radius is allowed per modelled system.*

We can also compare these two types of simulations in terms of variations of the other main products of glycolysis and related branches. Indeed, they both report a clear, yet similar, increase of the `glycerol` amount. Conversely, if we take into account the effects of long-distance interactions, the `trehalose` branch shows a change in its end product from 0.015 to 0.76 mmol/l, a higher concentration compared to the about 0.60 mmol/l resulting from a system limited by a 10 Å perception distance. Regarding `ATP` and `NADH`, their concentrations reach a value close to

zero almost immediately and then fluctuate slightly for the remainder of the simulation. This behaviour is observable due to the short interval of glycolysis we are analysing: at this stage of the process, the reactions that use ATP as an energy donor, as well as the redox conversion of dihydroxyacetone phosphate to glycerol 3-phosphate (which is coupled with the oxidation of NADH), still have an abundance of the substrate to consume. As a consequence, the related enzymes continuously bind the ATP and NADH in the environment to perform their catalytic activity.

We obtained remarkably opposed results in the case of simulations based on *perception distances of 5 Å*, which model a system affected only by short-range van der Waals-like potentials (Figure 6.3c). Despite the certainty that a metabolite will be bound by its cognate enzyme when it enters such a small perception sphere (as detailed in Section 6.2.2), at the end of the simulation, we can observe negligible increases in the concentration of the pathway end products as well as in the consumption of glucose. In particular, the curve representing this last metabolite reaches a plateau after a small depletion in its concentration, a behaviour we would observe at steady state; however, mostly because we did not implement enzyme regulation and glucose transport, such a condition is unlikely in our simulated systems. We can also observe similar concentration changes for glycerol (in this case, it increases before reaching a plateau). In both the situations, these anomalous behaviours are explainable if we observe the curves of ATP and NADH: the amount of ATP never decreases because, during the preparation phase of glycolysis, neither the hexokinases nor the phosphofructokinases are able to bind this molecule and complete the catalysis of their respective reactions. Indeed, glucose molecules are bound at the beginning of the simulation, but then the environment maintains the same concentration of hexokinase-glucose and glucokinase-glucose complexes for the entire simulated interval (plots reporting the concentrations changes of complexes are provided in Appendix Section C.1.2). This phenomenon also justifies why fructose 1,6-bisphosphate (product of the phosphofructokinase) can only decrease, consumed by fructose-bisphosphate aldolase. NADH, instead, remains stable at its initial concentration of 0.086 mmol/l because the glycerol-3-phosphate dehydrogenase is not able to bind it; when all the glycerol 3-phosphate in the environment has already been converted in glycerol, the latter can no longer be produced (causing the observed plateau of its curve).

We compared the results described thus far with the output obtained through a numerical time-course simulation; this has been performed via the Copasi software [55] over the Smallbone2013 model [107]. We modified the original SBML with the same adjustments to the considered reactions and initial molar concentrations of our ABM (see Section 6.2.1). However, we left the functions associated with enzyme regulation unchanged because a system of differential equations resulted in being less flexible than an ABM, and removing this feature would have compromised its consistency, making the numerical simulation impossible. As shown in Figure 6.3d, the kinetic model thus generates results closer to a steady-state condition, a property that, at first glance, may mislead the observer to find analogies with the simulations reproducing 5 Å perception distances.

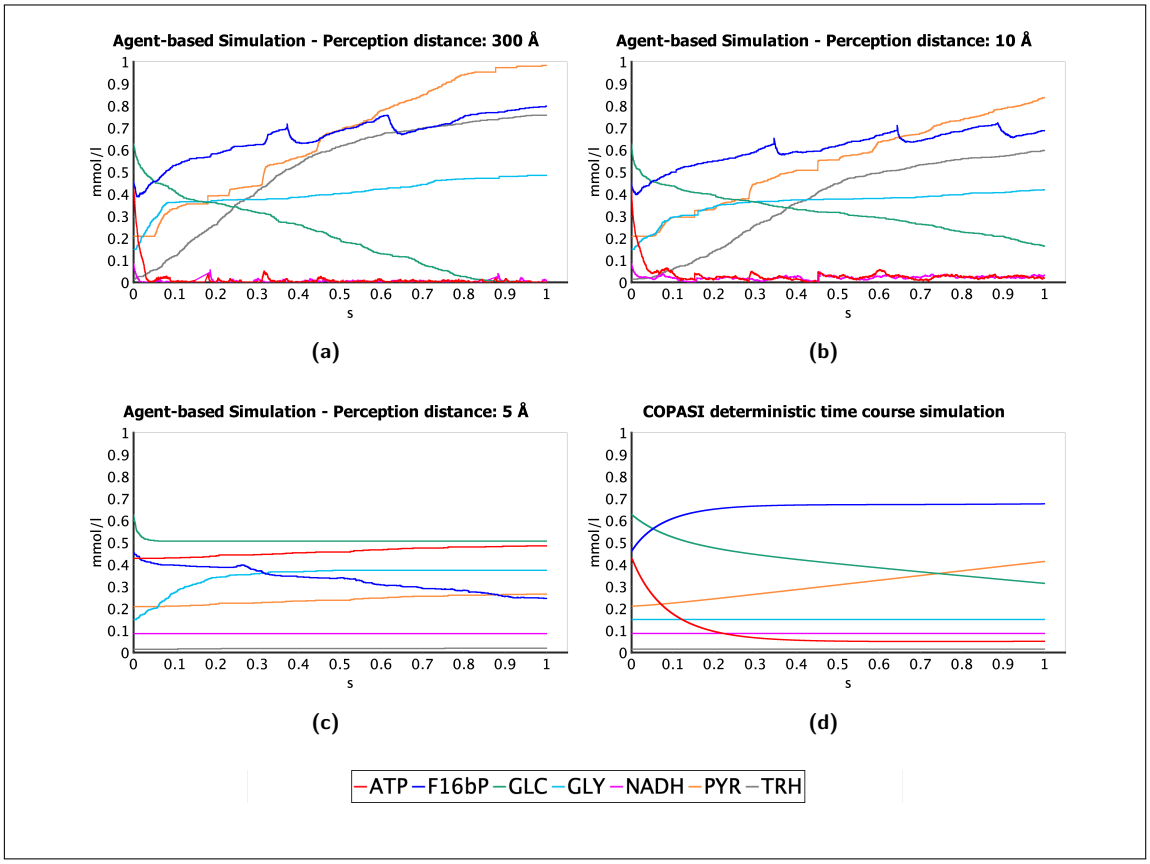


Figure 6.3 – Concentration changes over time, in simulations of 1 second, of a selection of metabolites particularly relevant for our study (for the complete set of plots, representing all the metabolites simulated, see Appendix Section C.1.1). This figure provides a comparison of the plots generated by three agent-based simulations–with perception distances set to 300 Å **(a)**, 10 Å **(b)**, and 5 Å **(c)**, respectively–and by a deterministic time course simulation based on the Smallbone2013 kinetic model **(d)**. We selected the following metabolite species: `glucose` (`GLC`), the source of the glycolytic pathway; `pyruvate` (`PYR`), `NADH`, and `ATP`, that is, the end products of glycolysis; `trehalose` (`TRH`) and `glycerol` (`GLY`), the products, respectively, of the two main glycolysis branches; `fructose 1,6-bisphosphate` (`F16bP`), the product of the most important control-point reaction of the glycolytic pathway, namely the one catalysed by the `phosphofructokinase`. In plot **(a)**, it is possible to notice how the simulation that takes into account long-range electrodynamic forces (300 Å perception distance) also shows a higher reactivity and an evident increase in the amounts of the pathway end products. In comparison, the simulation that limits the electromagnetic forces to those affected by Debye screening (10 Å perception distance), shown in **(b)**, is not able to consume the whole `glucose` in the environment and generates significantly smaller amounts of `pyruvate` and `thralose`. Simulating a system driven by van der Waals-like potentials (5 Å perception distance), whose plot is represented in **(c)**, causes negligible changes in metabolite concentrations, and the `glucose` consumption reaches a plateau; the agent-based approach allows us to attribute this behaviour to the inability of the reactions that use `ATP` or `NADH` as energy donor to bound these types of metabolites (see Appendix Section C.1.2 for further details). The plot **(d)** is generated through the deterministic time-course simulation of the Smallbone2013 model using the software Copasi [55].

However, excluding the fluctuations of metabolites concentrations, which are better captured by the ABM and more evident in the related plots, most of the shown concentration changes are loosely similar to those identified when we simulated 10 Å and 300 Å perception distances. This can be verified at least for the consumption of `glucose` and `ATP`, and for the increase of `pyruvate` and `fructose 1,6-bisphosphate`; nonetheless, they show significantly smaller variations from their initial molar concentrations. Although we consider identifying such properties in some of the most relevant species of the pathway noteworthy, we also point out that the last observations do not apply to all the simulated metabolites (as explained in Appendix Section C.1).

6.4 Discussion

The outcomes of the agent-based simulations detailed above suggest that the two systems that reproduce an off-resonance situation–where molecular interactions rely only on van der Waals-like potentials or, at least, on electromagnetic forces shorter than the Debye length–are not able to oxidise glucose at a high rate. This property is particularly true when we limit the *perception distance* to 5 Å, resulting in negligible changes in metabolite concentrations. By analysing the complexes formed by specific enzymes, such as `hexokinases` and `phosphofructokinases`, we attributed this behaviour to the inability of the electrostatic forces to guarantee the interaction of these enzymes with the needed energy donors. In this regard, the agent-based approach shows one of its major capabilities: it reproduces the dynamics of local interactions among the molecules (modelled as autonomous agents) and "captures" the formation of complexes, even when they are partly-saturated enzymes.

Such a possibility allowed us to observe, in the system limited by short-range electrostatic interactions, a condition that might be detrimental to the cell anaerobic metabolism, which commits the production of energy (in the form of `ATP`) only to a fast-paced glycolytic process. At the current stage of our work, this consideration represents just a hypothesis: in real cells, glycolysis processes occur in times ranging from a few seconds to hours [64, 83, 100, 116], making our one-second interval of simulation just a testbed to validate the capability of ABMs to support the study of the above-described forces in biological systems. However, if confirmed by further analyses, this result might suggest the non-feasibility of the *lock-and-key* model for enzymes in metabolic processes.

Interestingly, even though enzyme regulation has not been modelled, the systems driven by electromagnetic forces (including those below the Debye screening length) produce oscillatory-like fluctuations in the concentrations of `fructose 1,6-bisphosphate`, the main product of `phosphofructokinase`. Moreover, as shown in Figure 6.4, these fluctuations are synchronised with the concentration changes of `DHAP` and `GAP`, the products of the subsequent reaction in the glycolytic pathway, especially due to its reversibility. Conversely, such behaviour is almost unnoticeable in the output of the simulation that allows only short-range van der Waals-like potentials (5 Å perception distance). `Phosphofructokinase` has a central role in the regulation of glycolysis and, pivoting around this enzyme, an oscillatory behaviour has been experimentally

observed during the oxidation of glucose (even if at much lower frequencies) [100, 121]. Considering the high level of abstraction of the current glycolysis ABM, this result might be viewed as another clue that, by not limiting molecular interactions to just shape complementarities and chemical affinities, we generated processes more faithful to those occurring in cellular glycolysis.

We could not reach such a conclusion if we based our analysis on a standard kinetic model, which derives the changes over time of the concentrations (often of metabolites alone) through rate and balance equations. As it lacks the ability to represent the granularity of a molecular system, this approach hardly grasps the fluctuations in the species amounts, generating several discrepancies with the results we gained through our agent-based approach. Although differential equations best suit modelling the continuum or macroscale level [23], such divergences might also be attributed to the possible inaccuracy through which kinetic parameters are essayed in vitro. Indeed, already in the early 2000s, Teusink et al. questioned that in vitro kinetics could be able to faithfully describe an in vivo behaviour [112].

Molecular dynamics simulations, which may not be affected by the limitations of the standard kinetic approach, require a high number of physical parameters to be performed. Numerical simulations of this kind have been carried out to detect long-range interactions among biomolecules through the molecular diffusion behaviour [81]. In this case, simulating just one type of protein (the white egg Lysozyme) and one oppositely charged dye (the Alexa Fluor 488) required an a priori knowledge of several data; applying the same approach to a complex pathway of many reactions would be significantly more difficult than performing in silico studies through agent-based simulations.

6.5 Conclusions

We think that the results provided in this chapter support the reliability of ABMs in capturing the essential features of a complex biological process and faithfully reproducing different aspects of its behaviour, even on the basis of few empirical data. This approach identified in the long-range electrodynamic forces some of the fundamental "ingredients" necessary for glycolysis to operate efficiently.

However, we just laid the groundwork for further in silico and experimental studies that would explore those aspects of metabolism dynamics overlooked at the current stage of our analysis. An optimised implementation of Orion would allow longer simulations that, complemented by experimental validation of the present results, might highlight if some of our outcomes could be biased by the abstraction level of the agent-based models. Once we reinforce the robustness of our agent-based approach, it might pave the way for a better comprehension of those phenomena associated with cellular metabolism that are still not well understood. For example, it can be applied in the study of the Warburg effect, which describes the preference of cancer cells for the anaerobic (and energetically inefficient) consumption of glucose through glycolysis, even in the presence of a high oxygen concentration [118]. Recent studies have linked such a process to the effect of glycolytic oscillations [102] and to the rate of glycolysis, increased to provide a selective advantage over the metabolic competition in the tumour environment [66]. In this chapter, we have shown how long-range electrodynamic forces may affect the rate and efficiency of glucose oxidation and the oscillations in glycolysis intermediates; therefore, additional studies might enlighten us on their potential involvement in such an anomalous behaviour of tumour cells.

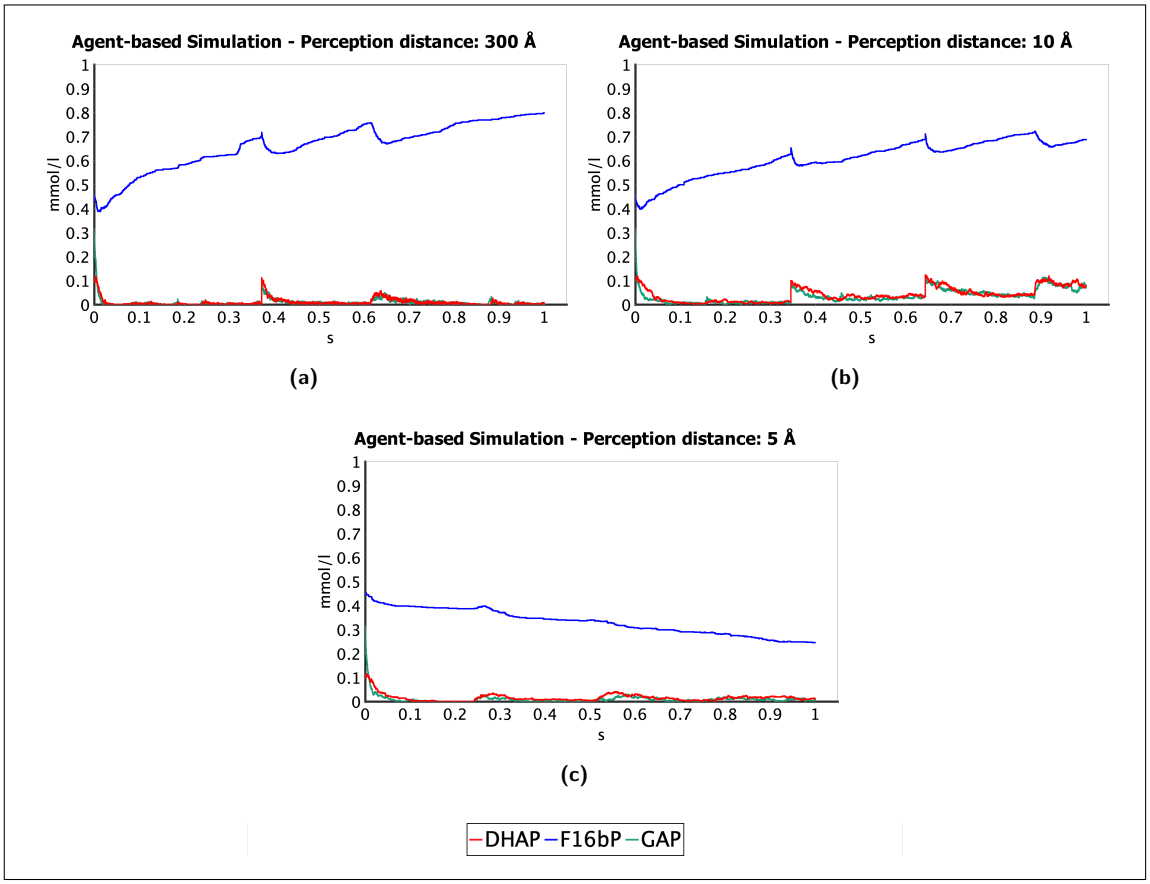


Figure 6.4 – Synchronised oscillation-like fluctuations observed in `fructose 1,6-bisphosphate` (`F16bP`), `dihydroxyacetone phosphate` (`DHAP`) and `glyceraldehyde 3-phosphate` (`GAP`). The first metabolite is the product of the phosphorylation of `fructose-6-phosphate`, catalysed by `phosphofructokinase`, while the other two are generated by the subsequent reaction in the glycolytic pathway, carried out by `fructose-bisphosphate aldolase`. `DHAP` and `GAP` are also interconverted by the `triosephosphate isomerase`. In **(a)** and **(b)**, that is, the plots of the simulations that take into account the electromagnetic forces (limited or not by the Debye screening), we can observe an oscillatory trend with a frequency of about 2.8 s^{-1}, synchronised in all the three curves. Conversely, in the simulation that considers just short-range electrostatic interactions, shown in plot **(c)**, these oscillations are almost unnoticeable. The higher frequency measured experimentally in yeast's glycolysis is 0.03 s^{-1} [100]; therefore, at the time scale of our simulations, these results give us just a clue of the higher faithfulness to the actual glycolytic process of the models whose interactions are not limited to just random encounters and chemical affinities.

Similar results might also be reached by empowering the capabilities of the agent-based approach with methods from other disciplines. Among them, the topological data analysis, already

used to better understand enzymatic reactions through ABMs (see Chapter 7), may provide the current model with a many-body perspective. Shape calculus can be applied to represent molecular conformations and increase the accuracy of the interactions between cognate partners; these approaches would allow modelling the geometry of molecule shapes and collisions with a higher precision [20]. Moreover, we may better capture the collective synchronisation properties of a population of molecules behaving as coupled oscillators by using BOSL, the biological oscillators synchronisation logic [10]. Putting efforts in these directions might provide a new standpoint in our comprehension of molecular interactions and disclose aspects of biological systems that are still unexplored.

Chapter 7

Modelling Interactions as Perceptions in Metabolic Reactions*

7.1 Introduction

This chapter analyses the space of potential reactions in a simulated metabolic process with the topological data analysis, one of the most effective methods to extract information patterns from a data collection [43, 76, 90, 91, 123]. This technique consists in building simplicial complexes–i.e., finite collections of objects, each of which could be seen as an n-body relation–and selecting the most meaningful one. Weight rank clique filtration is used to *map* simulation data into simplicial complexes and *visualise* the significant simplicial structures in the specific domain of metabolic reactions [25, 33, 88, 126].

This approach allows us to define a new visualisation paradigm based on the concept of *interaction-as-perception*: whenever a molecule perceives another one to interact with, a potential link between the two is established; the latter consolidates if the interaction ends up in a complex formation. In this way, we can derive the graph of perceptions at a given step; over this graph, we apply the topological data analysis to capture the 3-body interactions by interpreting 2-simplices–which are convex hulls of three points–as observable structures. We use the 2-simplex formation as a semantic to represent the global dynamics of the system and as a possible validation tool for the agent-based models of glycolysis introduced in the previous chapters.

*This chapter is derived from a co-authored work, conducted and published as part of the PhD project: Piangerelli, M., Maestri, S., Merelli, E., 2020. "Visualising 2-simplex formation in metabolic reactions". *Journal of Molecular Graphics and Modelling* 97, 107576. ©2020 Elsevier Inc. https://doi.org/10.1016/j.jmgm.2020.107576. M.P. and S.M. contributed equally to the work; they both curated the data and wrote the paper. M.P dealt with the topological data analysis; S.M. dealt with the agent-based modelling and simulation. E.M. supervised the research. All the authors conceptualised the study and reviewed the paper.

7.2 Additional Methods

7.2.1 Simulating glucose phosphorylation

The investigation we present is performed with the aid of Orion 2.0.0, the spatial simulator for metabolic pathways we discussed in Chapters 5 and 6, taking the Smallbone2013 model of glycolysis [108] as a source for the species concentrations and kinetic values (see Section 5.3.2 on page 92). The only reaction simulated for the aim of this study is the phosphorylation of glucose–catalysed by `hexokinase` and `glucokinase`–which produces `glucose 6-phosphate` and `ADP`; the Smallbone2013 model takes into account the contribution of isoenzymes; therefore, we considered the following three reactions:

$$\texttt{GLC} + \texttt{ATP} \xrightarrow{\texttt{HXK1}} \texttt{G6P} + \texttt{ADP}$$
$$\texttt{GLC} + \texttt{ATP} \xrightarrow{\texttt{HXK2}} \texttt{G6P} + \texttt{ADP}$$
$$\texttt{GLC} + \texttt{ATP} \xrightarrow{\texttt{GLK1}} \texttt{G6P} + \texttt{ADP}$$

For such reactions, the Smallbone2013 model provides the experimental data in Table 7.1. As explained in the following sections, the specificity constant is used in this study as a weight to characterise each molecular perception and interaction; thus, we ran the simulations by enabling the Orion option that assigns a priority to enzymes for their substrate based on the k_{cat}/K_m ratio (see Section 5.3.3 on page 94).

Table 7.1 – Initial concentrations and kinetic parameters from the Smallbone2013 model [108].

ID	$Conc.$ (mM/l)	k_{cat} (s^{-1})	K_{GLC} (mM)	K_{ATP} (mM)
enzymes				
HXK1	0.017	10.2	0.15	0.293
HXK2	0.061	63.1	0.2	0.195
GLK1	0.045	0.0721	0.0106	0.865
metabolites				
GLC	6.28	/	/	/
ATP	4.29	/	/	/
ADP	1.29	/	/	/
G6P	0.77	/	/	/

7.2.2 Simplicial data analysis

Topological data analysis is a promising technique for finding hidden patterns in (big) data. It is based on topology, a branch of mathematics that studies the shapes of spaces. According to

topology, a space can be characterised by quantities called *topological invariants*, which can be thought of as $n-$dimesional holes.

A topological space is constructed over a set of data points endowed with the notion of proximity, which characterises a coordinate-free metric. Because we are working in a discrete domain, we focus on topological spaces known as *simplicial complexes*. They are made up of building blocks called *simplices*: points are 0-simplices, line segments are 1-simplices, filled triangles are 2-simplices, filled tetrahedra are 3-simplices, and so on.

A *filtration* is a collection of nested simplicial complexes. Filtering can be compared to looking at a dataset through different lenses that allow extracting different types of information from the topological space; different filtrations result in different conversions of data points into simplicial complexes. In this chapter, we use the *weight rank clique filtration*, a graph-specific filtration that allows constructing a simplicial complex from a weighted undirected graph. *Graphs* are mathematical objects that lie in two dimensions: using simplicial data analysis, we derive from a graph the relative simplicial complex, which can be in any dimension. To perform the weight rank clique filtration and the related visualisation, we use a tool that is currently under development at the Bioshape and Data Science Lab of the University of Camerino. This tool exploits the GraphSharp library for visualisation.

7.2.3 Interaction-as-perception paradigm

The output of the simulator has been adapted to carry out a topological interpretation of the modelled molecular interactions. To achieve this result, we defined an *interaction-as-perception* paradigm applied to the agent dynamics of our metabolic simulator. The idea at the basis of this approach is that the perception between cognate partners could be interpreted as an abstraction for a complex formation.

Going into detail, we generated, along with the standard output of the simulator (as described in Section 5.3.4), additional information about every interaction performed at each time step. In particular, we gained the identifier of all the molecules involved in such an interaction and the value of the related k_{cat}/K_m ratio. Based on these data, we can define the following classes of perception:

- **Direct unstable perception**, of an enzyme for one of the possible cognate metabolites identified in its surroundings.
- **Direct fixed perception**, of an enzyme for an already bound metabolite (to form a dual-complex).
- **Indirect unstable perception**, of the metabolite forming the dual-complex for an external one perceived by the cognate enzyme; the enzyme mediates this kind of perception, which, by convention, has the fixed value of 0.001.
- **Indirect fixed perception**, of a metabolite for another metabolite bound to the same enzyme.

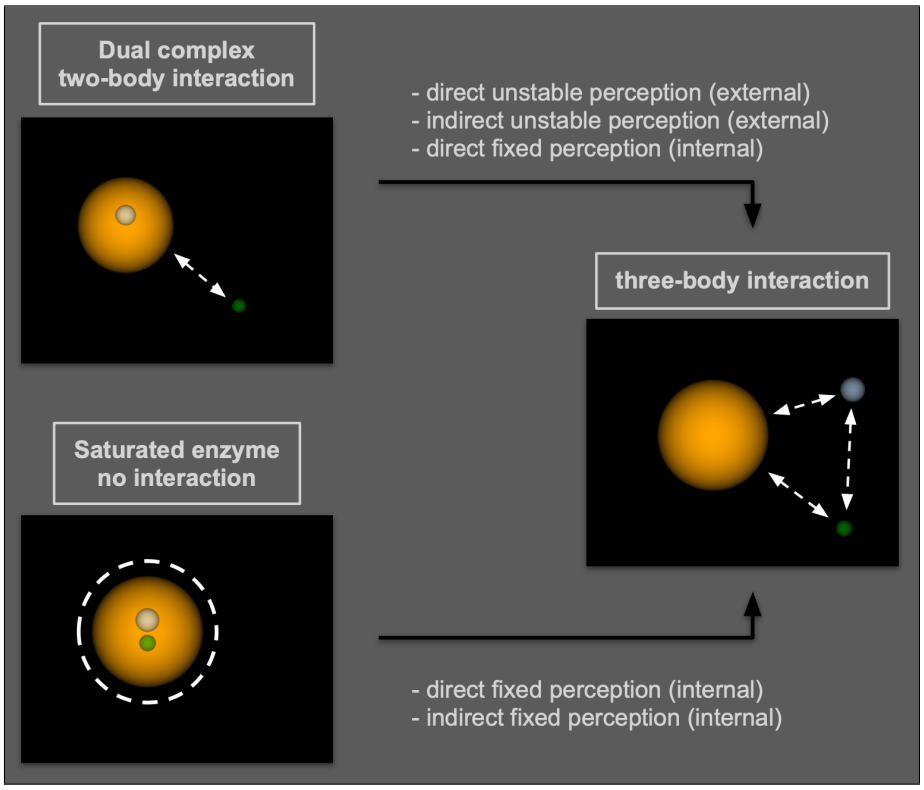


Figure 7.1 – Representation of the interaction-as-perception paradigm. In the classical agent-based model, the interaction between a dual-complex and a complementary metabolite is 2-body; a saturated enzyme has no interactions at all. Conversely, through the interaction-as-perception paradigm, they can be interpreted as 3-body since we consider the potential interactions. However, to illustrate this paradigm based on the entities of an agent-based simulator, we need to force the original model and disrupt the structures represented by the agents. This limitation is overcome by the topological representation of intermolecular perceptions as simplicial structures.

By analysing the dynamics of the agent-based simulations from the above-defined perspective, we can observe the following behaviours:

- A *free enzyme* can have direct unstable perceptions or no perception at all (if there is no other compatible molecule in its surroundings).
- A *dual-complex*, since it forms when an enzyme binds one of the perceived metabolites, always carries out an inner fixed perception–of the enzyme for the bound metabolite. Two additional kinds of perceptions are generated for every external compatible metabolite it identifies, i.e., the direct and the indirect unstable perceptions performed, respectively, by the enzyme and by the metabolite composing the dual-complex.

- A *saturated enzyme* can show just the direct fixed perceptions of the enzyme for the bound metabolites and an indirect fixed perception between the two metabolites (if more than one is present, as in the case of the reaction we analysed). This condition is maintained for the duration of the delay given by the reciprocal of the reaction's k_{cat} (after which the enzyme returns free, and two new metabolites, corresponding to the products of the reaction, are released in the simulation environment).

These three different behaviours identify the states of the automaton that describes the cyclical pattern of an enzymatic reaction (see Section 5.3.1). The iteration of this cycle drives the evolution of the reaction through phases of higher/lower stability, a property that we highlight through quantitative analysis of the topological representation (2-simplex) of intermolecular perceptions (see Figure 7.3). The 2-simplex structures provide a higher-order global representation of interactions than a classical agent-based model. In the latter, each molecular interaction is 2-body, defined according to the biochemical reactions (such as those shown in Section 7.2.1), and generates a new agent (a new complex or a final product); conversely, in the topological setting, the potential interactions between molecules can be 3-body and represented as a whole on the basis of the interaction-as-perception paradigm (see Figure 7.1).

7.3 Results

By applying our agent-based simulation to study the metabolic reactions catalysed by hexokinase isoenzymes, we can observe how the molecules in the simulated environment move and interact at each time step (see Chapter 5 for details).

To analyse the dynamic evolution of each reaction from a topological point of view, we need to abstract, based on an interaction-as-perception paradigm, from the standard spatial simulation output. According to such an approach, an enzyme perceives a cognate metabolite whether a metabolite enters its perception sphere (see Section 6.2.2 on page 107) or the two molecules actually bind. The resulting network of intermolecular perceptions can be interpreted in terms of simplicial complexes formation, where, every time an enzyme perceives a cognate metabolite, an *edge* is drawn between the two molecules.

Changes in simplicial structures go along the evolution of the simulated reaction, according to the following general observations:

- At the beginning of the simulation, every molecule in the simulated volume does not perceive nor interact; therefore, the topological environment is filled with sparse nodes (0-simplices–see Figure 7.2a);

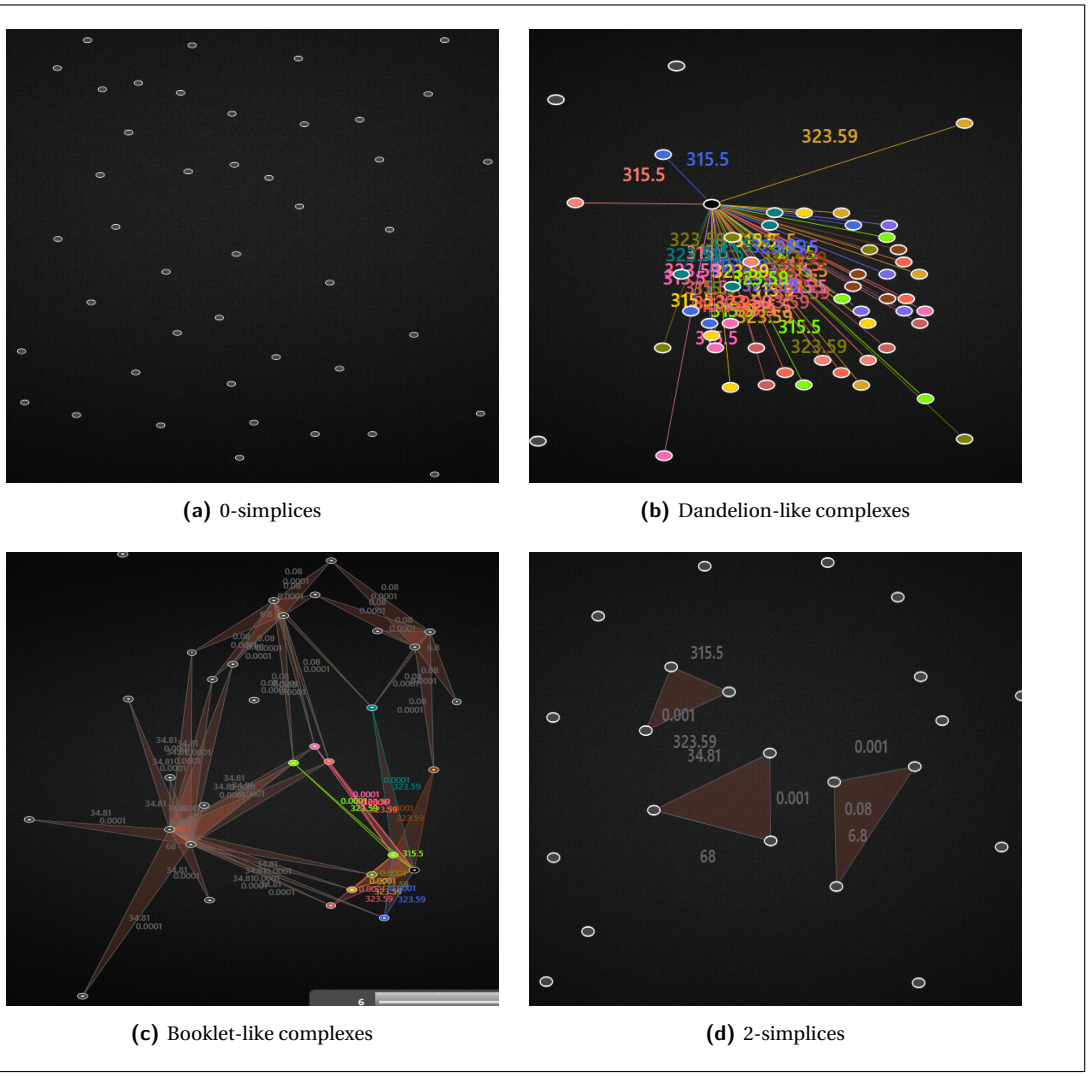
- In the first simulation instants, since enzymes start to perceive the related substrate, we can observe the formation of isolated enzyme-metabolite edges (1-simplices) as well as of "dandelion-like" structures (Figure 7.2b), made of a central hub (the enzyme) connected to multiple nodes (metabolites).

- The binding between an enzyme and a single metabolite is caught in our representation by the formation of stable isolated 1-simplices composed of the two nodes.
- Each metabolic complex may perceive the presence of the metabolite needed to saturate the enzyme; in this case, we can both observe the presence in the environment of isolated triangles (2-simplices) and "booklet-like" complexes, each made of an edge placed at the centre of a star of 2-simplices and linking the half-saturated enzyme to its bound metabolite (as shown in Figure 7.2c). Every triangle of this type is a potential stable link connecting the central complex and the opposite vertex.
- The potential conditions described above are resolved when fully saturated enzymes form; they are identified by stable 2-simplices (Figure 7.2d). Each final complex remains in the simulation volume for a time given by the reciprocal of the related k_{cat}; therefore, after such a delay, three new isolated nodes appear in place of a 2-simplex, i.e., those representing the enzyme and the products of the catalysed reaction.

All the simplicial complexes we can observe during the time evolution of the simulation have a direct correlation with the perception-based structures described in Section 7.2.3. Table 7.2 summarises such relations by associating each simplicial structure identified in the previous description with the corresponding perception class.

Table 7.2 – Correlation between interaction-as-perception paradigm and simplicial structures.

Interaction as perception (Multiagent Simulation)		**Simplicial Data Analysis**
Molecule	**Perception**	**Structure**
free enzyme	no perception	0-simplex (isolated node)
	direct unstable perception	1-simplex\ dandelion-like structure
dual-complex	no perception	1-simplex
	direct unstable perception (external) indirect unstable perception (external) direct fixed perception (internal)	2-simplex\ booklet-like structure
saturated enzyme	direct fixed perception (internal) indirect fixed perception (internal)	stable 2-simplex



(a) 0-simplices

(b) Dandelion-like complexes

(c) Booklet-like complexes

(d) 2-simplices

Figure 7.2 – This figure shows the most significant structures we can identify through our topological analysis of the simulation output. **(a)** 0-simplices, representing all the molecules at the beginning of the simulation; **(b)** a "dandelion-like" structure made of a central node (enzyme) linked to the nodes corresponding to the compatible substrate inside its perception sphere; **(c)** "booklet-like" structures composed of a central hub made of two linked nodes (enzyme-metabolite dual-complex), each forming an edge with an external node–i.e., a metabolite that can complete the enzyme saturation; **(d)** isolated 2-simplices correlated to the saturated enzymes that we can identify in this portion of the environment. In figures (b), (c) and (d), the value above each edge–i.e., its weight–represents the specificity (k_{cat}/K_m ratio) of the enzymes for the cognate metabolite connected by that edge.

Representing the dynamics of the agent-based simulation using the simplicial approach described above allows us to highlight some fundamental properties of the progression of metabolic reactions over time. Specifically, we can observe that changes in the system's reactivity are affected by the fluctuation of 2-simplices concentration. A simulated reaction alternates states of high reactivity and states of semi-stability that can be correlated to the number of 2-simplices identifiable in the environment. Stars of 2-simplices determine the system's instability; therefore, we observe high concentrations of these "booklet-like" structures during the reactive phases. As shown in Figure 7.3, considering a long temporal horizon, blocks of reactive phases are clearly distinguishable from those almost saturated with stable 2-simplices (representing final molecular complexes).

Inside these higher reactive blocks, the formation of stable 2-simplices causes the transition from one reactivity phase to another. Indeed, a new stable 2-simplex forms when a star of 2-simplices resolves its instability (by choosing one of the possible associated peripheral nodes); such an event determines the immediate drop of the system's 2-simplices amount, correlated to just one unit increase of stable 2-simplices. As we can observe in Figure 7.3, such a behaviour determines a progressive decrease in 2-simplex stars amount and, therefore, in the block's reactivity over time.

We also highlight that a transition from a stable to a reactive block is related to the k_{cat} value of the reaction since it determines the time interval through which a stable 2-simplex maintains its conformation. After such a lapse of time, the product is released, and the enzyme starts to look for a new substrate, pushing the system towards a new reactive block. In Section 7.2.3, we mentioned a three-state automaton as a formal representation of the studied enzymatic reaction. The progression through phases of the simulation as described above is directly related to the cyclical iteration of the three states of a reaction, identified by the molecular structures that cause them, i.e., free enzymes, dual complexes, and saturated enzymes (see Figure 7.3).

7.4 Discussion

In the present work, we use an agent-based simulation to generate the dynamics of a complex system and the weight rank clique filtration to try to visualise and understand the global behaviour of that system.

Thanks to the interaction-as-perception paradigm, the visualisation clearly shows the formation of the simplicial structures characterising the system. Such structures are directly correlated to the dynamical evolution of molecular complex formation and allow us to identify specific patterns that underline the in silico behaviour of a metabolic reaction. Moreover, those instruments gave us insights into what happens in the simulated systems in terms of topological invariants.

Even if we do not claim to infer any direct biological meaning from these results, we hypothesise that the patterns mentioned above reveal the reactivity trend of the modelled reaction, turning out to be an effective validation tool for a biochemical reaction simulation. Indeed, we can compare the highlighted trends with those obtained by applying our visualisation method to

other well-proven modelling approaches (e.g., based on differential equations) or even directly to experimental data. This approach might allow us to identify how the simulated process differs from the one chosen as a benchmark and, consequently, make the necessary adjustments to make them fit.

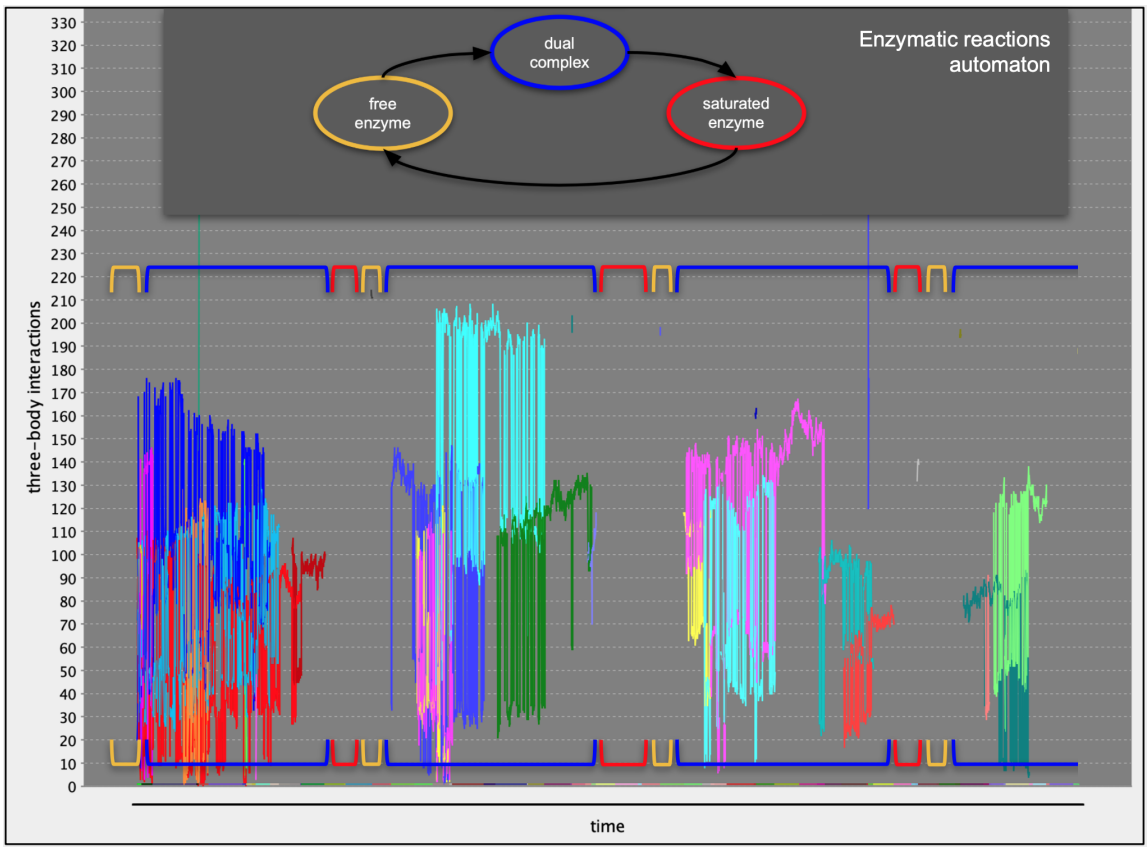


Figure 7.3 – Changes over time of the number of 2-simplices associated with each edge representing a dual complex; they are plotted along with the number of the stable 2-simplices (corresponding to saturated enzymes). The aim of this plot is to provide a global view of how, on a long temporal horizon, highly reactive blocks alternate with time intervals dominated by stable 2-simplices. Each block is correlated to the automaton states representing the three steps of the enzymatic reaction, respectively dominated by high concentrations of free enzymes (yellow state), dual complexes (blue state) and saturated enzymes (red state). Their iteration drives the evolution of each reactivity block shown in the plot, as identified by the square brackets coloured as the related state of the automaton. Due to the large number of complexes represented, a complete legend describing all of them would impact the readability of the figure.

7.5 Conclusions

Agent-based computational models and simplicial data analysis are well-suited methods for simulating and visualising the dynamics of complex systems, which are characterised by a high number of entities interacting in a bounded space. Moreover, they allow us to represent some specific features of the system to be compared with empirical observations or experimental data in a future work. By studying the global behaviour of an agent-based simulation with the simplicial data analysis, we have advanced the visualisation capabilities of the Orion simulator, and we were able to identify the simplicial structures associated with the reaction space over time. This result might be a valuable validation tool for the agent-based simulation. Indeed, it opens the possibility of performing the same simplicial data analysis on empirically retrieved data to verify the faithfulness of the simulation to the actual biological process [73].

At the same time, identifying patterns in the reactivity associated with the molecular interactions graph might provide computational support for studying therapies based on drug targeting and enzyme inhibition [15, 21, 47, 113].

As further developments, we are working on other validation approaches that could be combined with those mentioned above, particularly those involving innovative applications of formal methods in the analysis of biological processes [10].

Conclusions

The core idea of this dissertation is that a bottom-up modelling approach, capable of grasping the global properties resulting from local interactions, provides the perspective required to fully understand a biological system. However, beyond a strictly theoretical analysis, it is possible to identify another common thread in the issues addressed. Throughout this manuscript, we have indeed characterised the steps needed to develop a computational framework able to contribute to studies performed on experimental data; this type of software platform is intended to meet the growing medical needs for supporting in silico personalised therapy design.

From this standpoint, the models presented in this manuscript can be applied to develop a simulator that generalises interactions in biological systems, whether they are between molecules or other entities (such as cells). Orion, which is described in Part II, is a prototype of this kind of simulator.

The engineering life cycle for the simulation of a biological process can be divided into two phases (which Figure 4.3 on page 81 depicts schematically):

1. process modelling and verification;
2. system modelling, simulation, and validation.

The starting point is the actual biological system, from which we derive an abstraction of the functions we want to model and simulate. These functions are then formally defined using process algebras, and the properties of the resulting models are verified using the most suitable model checking method. Part I of this manuscript delves into this *first phase*.

More precisely, in Chapter 2, we model the folding processes as behaviours resulting from the interactions that nucleotides and amino acids (the monomers that make up RNAs and proteins, respectively) perform on the linear sequences to which they belong. Initially, this approach was intended to provide new knowledge about the studied systems without relying solely on empirical data. Using Milner's Calculus of Communicating Systems (CCS) to highlight the distinguishing features of the two folding processes, we discovered an abstraction level at which they show behavioural equivalence. To this level belong all the functions expressible by non-coding RNAs (ncRNAs), interpreted as a subclass of protein functions. To advance this idea, we used CCS and Hennesey-Miler logic to represent the process that leads to the formation of misfolded proteins. In Chapter 3, a class of pathologies affecting RNA and proteins is modelled as global behaviours

generated by both nucleotide and amino acid interactions; these results allow us to study their different responses to a change in the correct folding pathway.

The algebraic approaches described thus far were not originally intended to develop simulation software, but rather to investigate a theoretical method for acquiring new knowledge about biological processes. By analysing the complexity of the interactions that characterise living systems, we defined a new methodology for understanding biological behaviours. We could, however, construct an algebraic specification for an actual simulation based on these models. For these reasons, in Chapter 4, we look at the expressiveness of process algebras for modelling ncRNA behaviour not only for theoretical purposes, but also for building agent-based models and validating hypotheses through model simulation. We hope, in this manner, to support the study of cellular processes and pathologies involving ncRNAs.

As a first step in implementing these specifications, we conducted preliminary studies to identify an agent-based approach that meets the requirements for simulating molecular interactions. We found the best solution for our needs in Orion, a spatial simulator for metabolic pathways. However, because it was a prototype project, its functionality was required to be significantly improved. The results of this work are described in Part II of the dissertation and represent the *second phase* of the simulator engineering life cycle. It entailed the definition of a low-level specification, the generation of the actual agent-based simulation, and the validation of the obtained results, all to make the agent-based model more faithful to the biological system. These steps led to the development of Orion 2.0.0.

Chapter 6 describes a preliminary study in this direction, where we adapted the original Orion prototype to analyse the effect of long-distance electrodynamic interactions among biomolecules. We put our approach to the test by simulating the glycolytic pathway to observe the collective behaviour of molecules involved in a reaction network; the goal was to detect the role that long-range electrodynamic forces might play in the effectiveness of glucose oxidation. The results obtained demonstrate the ability of our agent-based simulations to manage interactions in complex biological systems; Orion 2.0.0 may thus represent a suitable platform for implementing the algebraic models defined in the first section of this dissertation.

For validating the metabolic simulations, in Chapter 7, we investigate the potentiality of the *interaction-as-perception* detectable in agent-based systems. We performed a topological data analysis on the molecular perception graphs obtained during the formation of the enzymatic complexes to visualise the set of emerging patterns. We were able to identify the simplicial structures associated with the reaction space over time and address the complexity of visualising the global behaviour of a metabolic reaction. This visualisation approach could be a valuable validation tool for our agent-based simulations because it allows the same simplicial data analysis to be performed on empirically retrieved data; it thus supports the verification of the fidelity of the simulation to the actual biological process [73].

Future Directions

As a next step in defining a framework for supporting the in silico design of medical therapies, we are considering the treatment of renal cell carcinoma (RCC). RCC is a type of kidney tumour that is gaining increasing attention; this is due not only to its spread but also to its association with the obesity paradox, a phenomenon in which obese patients, despite having a high risk of developing RCC, have better prognoses than lean individuals [104, 105].

We are modelling the RCC system using a formal approach that allows us to describe oncologists' knowledge and acquire new information from experimental data. Given the characteristics of the RCC system, the model must be able to represent the tumour microenvironment (TME) as the main component in which immune and tumour cells are spatially distributed and move, influenced by blood vessels; the effect of the TME on the tumour is a function of the body mass index (BMI) [11].

We can find many related studies in the literature, the majority of which use mathematical models based on ordinary differential equations and partial differential equations enriched with stochastic elements, as well as other physical models based on complex networks and phase transition analysis [29]. However, none of them allows for the explicit description of the environment as a first component of the model. As a result, we are developing a computational framework to support the learning process in which an RCC model is dynamically defined during immunotherapy treatment given to different patients. This method is inspired by the modelling and simulation approach described in this manuscript and our previous works [9, 73].

Agents are active system components; if they represent molecules in the glycolysis model, they correspond to immune or tumour cells in the RCC model. The global properties are observable in the RCC simulation, whose dynamics are expressed by the interaction-as-perception paradigm, adapted to this new context: a cell agent moves towards another perceived cell agent to interact with it, activating or inhibiting the production of compatible cell agents. The immune-response interactions are all dynamic representations of the cell interaction network, bound by the TME and controlled by the BMI.

We have already begun to validate this new simulation approach using experimental data, and the results are promising. It must, however, be further developed. We also want to build a topological classifier that can distinguish between different microenvironments and identify the reversible RCC behaviour class.

Final Remarks

Although the approach adopted in the first part of this dissertation is strictly theoretical, a process-based view of molecular structures and functions can reveal congruence and dissimilarities difficult to detect through other computational methods or experimental techniques; this perspective can thus inspire the investigation of properties not yet considered in the current studies on RNA structure-function relationship. The proposed results are based on the construc-

tion of algebraic models through process calculi, which provide us with factual knowledge. For this reason, we believe that applying formal models to the study of non-coding RNA functions can pave the way to a better understanding of this class of molecules and therefore contribute with solid support to handling the pathologies in which they are involved. The same approach may be effectively extended to other biological functions and diseases.

The results provided in Part II complement the previous analysis by showing the reliability of agent-based models in capturing the essential features of a complex biochemical process and faithfully reproducing different aspects of its behaviour, even on the basis of little empirical data. Considering these results, we are optimistic that the analysis of agents' interactions will be able to bring new knowledge on the properties of different biological systems. Nonetheless, their global behaviour is not always predictable due to the incompleteness of the observed data. Agents' interactions must be aleatory, or the simulation environment must be unpredictable; this implies that each simulation run should be affected by statistical uncertainty. Additional steps are thus required to provide an accurate environment specification, hopefully referring to interactive computation modelling [74].

Appendices

Appendix A

Supplementary Information to Chapter 2

Process Calculi May Reveal the Equivalence Underlying RNA and Proteins

A.1 Models Construction

In our models of the folding process, non-covalent interactions are classified into three main categories:

- hydrogen bonds;
- electrostatic interactions (ionic and van der Waals);
- hydrophobic and hydrophilic interactions.

The hydrogen bond could be considered an electrostatic interaction, but due to its distinctive properties and the fundamental role it carries out in the folding process, it is categorised separately.

All the non-covalent interactions listed above are modelled to formally describe the whole folding process. Each folding process starts from a linear strand (of nucleotides in RNAs and amino acids in proteins) and is driven by the reduction in free energy between two different folded configurations. The free energy variation during folding, denoted by ΔG , is represented as a process that can produce three possible outputs: negative, positive, or zero.

To better clarify this concept, we can imagine the folding process as a sequence of folding steps, each contributing to the entire process with a new non-covalent interaction between two sequence units (equally for RNAs and proteins). For a folding step to occur, the non-covalent interaction must cause a reduction in the free energy of the system, which means that the folding step must have a negative ΔG.

A.1.1 Base pairing

In *RNA*, hydrogen bonds allow the pairing between two bases. According to Watson-Crick base pairing, adenine (A) always pairs with uracil (U) with two hydrogen bonds, while guanine (G) always pairs with cytosine (C) with three hydrogen bonds. At the same time, the non-canonical base pairing shows various combinations of the four RNA bases, forming two hydrogen bonds (or even only one); it is not infrequent to find in RNA also base triples (indeed, it is possible that a unique base quartet forms between G-C base pairs at the junction of two helices).

The hydrogen bond formation (in both Watson-Crick and Wobble base pair) is modelled generalising this process as an interaction between a purine (adenine or guanine) and a pyrimidine (uracil and cytosine) or between two paired bases and a third base (in this case, a generic purine or pyrimidine). Since purines are **d**ouble-**r**ing bases, they are labelled `dr`; pyrimidines, conversely, are **s**ingle-**r**ing bases and hence labelled `sr`. The base pairing is symmetric, thus `srdr` = `drsr`.

For removing some details not necessary to our model definition, we also opt for another generalisation: we do not explicitly represent all the possible interactions between a couple of paired bases and a third base, but we indicate this process as a "triple base pairing" ($\mathcal{P}_{\texttt{b3}}$) and its output as "three paired bases" (`tpb`). For the same reason, the formation of the G-C base quartet is not treated in the model.

Regarding the number of hydrogen bonds in a base pair, our models allow them to be at least two and at most three. Conversely, the hydrogen bonds that link an unpaired base to a group of two already paired bases must be from one to three. We introduce these constraints because base pairs with a single hydrogen bond can be classified as variants of those linked by two, and the number of hydrogen bonds found in a base triplet is three to six [80]. Moreover, because–up to now–the sole known base pair that involves three hydrogen bonds is the one between cytosine (C) and guanine (G), only the `srdr` base pair is allowed, in the model, to form through a triple hydrogen bonding; this means that AU, GU and CA base pairs could also potentially be linked by three hydrogen bonds, which is a stretch of the current knowledge on hydrogen bonding. Indeed, if we wanted to capture the limiting constraint that allows the formation of three hydrogen bonds only in the GC base pair, we would have to explicitly represent every base and its combination with the others; this would reduce the readability of our models to introduce a property that does not affect the primary purpose for which they are defined.

The *base pairing* process ($\mathcal{P}_{\texttt{b2}}$) takes two unpaired bases (`ub`) as input and provides the corresponding base pair as output only if it can form at least two hydrogen bonds (`hb`) between them.

$\mathcal{P}_{\texttt{b2}}$ is a sub-process of a general $\mathcal{F}^{s}_{\texttt{rna}}$ (*RNA Folding Step*) process, from which it receives its input (the $\mathcal{F}^{s}_{\texttt{rna}}$ process will be described later in this section); it is one of the possible sub-processes that give each folding step its specificity. As also explained in Chapter 2, each folding step, and therefore each base pairing process, is conditioned by the value of the ΔG: it can take place only if its ΔG is negative.

The *triple base pairing* process ($\mathcal{P}_{\mathtt{b3}}$) takes as input (from the $\mathcal{F}^s_{\mathtt{rna}}$ process) a couple of bases, paired by the $\mathcal{P}_{\mathtt{b2}}$ process, and a third unpaired base (`ub`), providing as output a group of three paired bases (`tpb`). The number of hydrogen bonds generated in this process is at least one and at most three.

Like the $\mathcal{P}_{\mathtt{b2}}$ process, $\mathcal{P}_{\mathtt{b3}}$ is a sub-process of $\mathcal{F}^s_{\mathtt{rna}}$ and depends on the value of ΔG (output of the `ΔG` process) to take place.

The following is the specification of the $\mathcal{P}_{\mathtt{b2}}$ and the $\mathcal{P}_{\mathtt{b3}}$ processes using Milner's CCS (in Section A.1.4 on page 140 they will be contextualised in the definition of the whole $\mathcal{F}^s_{\mathtt{rna}}$ process):

$$
\begin{aligned}
\mathcal{P}_{\mathtt{b2}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}1_{\mathtt{b2}};\\
\mathcal{B}1_{\mathtt{b2}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}2_{\mathtt{b2}};\\
\mathcal{B}2_{\mathtt{b2}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}3_{\mathtt{b2}} + \overline{\mathtt{srsr}}.\mathcal{F}^s_{\mathtt{rna}} + \overline{\mathtt{drdr}}.\mathcal{F}^s_{\mathtt{rna}} + \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}};\\
\mathcal{B}3_{\mathtt{b2}} &\stackrel{\text{def}}{=} \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}};
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
\mathcal{P}_{\mathtt{b3}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}1_{\mathtt{b3}};\\
\mathcal{B}1_{\mathtt{b3}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}2_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}};\\
\mathcal{B}2_{\mathtt{b3}} &\stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}3_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}};\\
\mathcal{B}3_{\mathtt{b3}} &\stackrel{\text{def}}{=} \overline{\mathtt{tpb}}.\mathcal{F}^s_{\mathtt{rna}}.
\end{aligned}
$$

$\mathcal{B}1_{\mathtt{b2}}$, $\mathcal{B}2_{\mathtt{b2}}$, $\mathcal{B}3_{\mathtt{b2}}$ (hydrogen bonding between two bases) and $\mathcal{B}1_{\mathtt{b3}}$, $\mathcal{B}2_{\mathtt{b3}}$, $\mathcal{B}3_{\mathtt{b3}}$ (hydrogen bonding between three bases) are states that allow counting the number of the hydrogen bonds.

In *proteins*, a hydrogen bond can form between the amino group of one amino acid and the carboxyl group of another. Every amino acid has an amino group and a carboxyl group covalently linked to the alpha (central) carbon (see Section 1.2.3). In the rest of this chapter, the terms "amino groups" and "carboxyl groups" will refer specifically to such functional groups. In contrast with the base pairing of nucleotides, only a single hydrogen bond is allowed between two amino acids; however, there is no limitation in the length of a sequence of amino acids linked to one another via hydrogen bonds.

Therefore, two amino acids can link to each other through a hydrogen bond only if they meet the following conditions:

- the interaction has a negative ΔG;
- the amino group of one of the two interacting amino acids and the carboxyl group of the other are both free (not involved in a hydrogen bond).

The *amino acid pairing* process ($\mathcal{P}_{\mathtt{aa}}$) is a subprocess of the general $\mathcal{F}^s_{\mathtt{p}}$ (protein folding step), as $\mathcal{P}_{\mathtt{b2}}$ is a subprocess of $\mathcal{F}^s_{\mathtt{rna}}$. $\mathcal{F}^s_{\mathtt{p}}$ provides two amino acids (`aa`) as input to $\mathcal{P}_{\mathtt{aa}}$, which generates a hydrogen bond between the free amino group of the first one (`aa1fnh`) and the free carboxyl group of the second one (`aa2fco`), or between the free carboxyl group of the first amino acids

(aa1fco) and the free amino group of the second one (aa2fnh). The process produces a group of two paired amino acids (paa) as output.

It is important to notice that:

1. although the distinction between "first" and "second" amino acid might appear unnecessary when they are both unpaired, it has to be specified to deal with the situation in which at least one of the two amino acids is already involved in a hydrogen bond through one of its functional groups;

2. when the $\mathcal{P}_{\mathtt{aa}}$ process receives two amino acids as input, we have the certainty that a hydrogen bond will form because the negative ΔG of the interaction has already been checked in the early phases of the $\mathcal{F}^s_{\mathrm{p}}$ process.

The following is the CCS specification of the $\mathcal{P}_{\mathtt{aa}}$ process:

$$\begin{array}{lcl}
\mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa1fnh}.\mathtt{NH}_{\mathtt{aa1}} + \mathtt{aa1fco}.\mathtt{CO}_{\mathtt{aa1}}; \\
\mathtt{NH}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fco}.\mathtt{CO}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fnh}.\mathtt{NH}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathtt{NH}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathcal{B}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{\mathtt{paa}}.\mathcal{F}^s_{\mathrm{p}}.
\end{array} \tag{A.2}$$

$\mathtt{NH}_{\mathtt{aax}}$ and $\mathtt{CO}_{\mathtt{aax}}$ (where x is 1 or 2) are states that indicate the selection of the free amino group or the free carboxyl group, respectively, of the x-th amino acid.

A.1.2 Electrostatic interactions

Two particles electrically charged can interact according to Coulomb's law; however, the model of the folding process does not investigate the interactions at the atomistic level. We consider that two elementary units–of either an RNA or a protein–can be involved in a folding step if they are both charged and if the ΔG of the step is negative. The main purpose of this kind of interaction is to stabilise the folded structure reached through the previous steps.

The electrostatic interaction can be of two types: ionic and van der Waals. The ionic interactions cause the formation of a non-covalent bond between two ions of opposite charge; the van der Waals interactions occur between two molecules oppositely polarised.

The modelling of these interactions is essentially the same in both RNA and protein folding: given as input a couple of bases (in the RNA model) or amino acids (in the protein model), each unpaired or already paired, the *electrostatic interaction* process allows the nondeterministic choice between an ionic interaction (*ii*) or a van der Waals interaction (*vdwi*), which are produced as output.

The *electrostatic interaction between bases* ($\mathcal{I}^e_{\mathtt{b}}$ process) specifies the electrostatic interactions in the RNA folding model:

$$\mathcal{I}^e_{\mathtt{b}} \stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}^s_{\mathtt{rna}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{rna}}. \tag{A.3}$$

The *electrostatic interaction between amino acids* ($\mathcal{I}^e_{\mathtt{aa}}$ process) specifies the electrostatic interactions in the protein folding model:

$$\mathcal{I}^e_{\mathtt{aa}} \stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}^s_{\mathtt{p}} + \overline{vdwi}.\mathcal{F}^s_{\mathtt{p}}; \tag{A.4}$$

$\mathcal{I}^e_{\mathtt{b}}$ is a subprocess of $\mathcal{F}^s_{\mathtt{rna}}$; $\mathcal{I}^e_{\mathtt{aa}}$ is a subprocess of $\mathcal{F}^s_{\mathtt{p}}$.

A.1.3 Hydrophobic interactions

Water is a polar solvent, which means that it easily dissolves charged or polar compounds, which are called, for this reason, hydrophilic (from Greek, "water-loving"). In contrast, nonpolar molecules are hydrophobic.

In *RNA*, the bases are hydrophobic and relatively insoluble in water, while the backbone of alternating ribose and phosphate groups is hydrophilic. During the folding process, the backbone forms the RNA's outer surface to minimise the contact of the bases with water and stabilise the molecule's three-dimensional structure; in contrast, the bases are positioned on its inside, stacked with the planes of their rings parallel to each other (a process called *hydrophobic stacking interaction*).

In the RNA folding model, the *hydrophobic interaction of bases* ($\mathcal{I}^h_{\mathtt{b}}$ process) takes two bases as input, produces a hydrophobic interaction for both of them (*hbi*), and provides as output the same bases buried inside the RNA (`bb`) and stacked to each other (`sb`).

Since $\mathcal{I}^h_{\mathtt{b}}$ is a subprocess of $\mathcal{F}^s_{\mathtt{rna}}$, the negative value of its ΔG has already been checked in the earlier phases of the latter process.

The CSS specification of the $\mathcal{I}^h_{\mathtt{b}}$ process is

$$\begin{array}{lcl} \mathcal{I}^h_{\mathtt{b}} & \stackrel{\text{def}}{=} & hbi.\mathtt{I}_{\mathtt{rna}}; \\ \mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bb}}.\mathcal{S}; \\ \mathcal{S} & \stackrel{\text{def}}{=} & \overline{\mathtt{sb}}.\mathcal{F}^s_{\mathtt{rna}}. \end{array} \tag{A.5}$$

In *proteins*, the specific characteristics of an amino acid are determined by the properties of its R group (also called *side chain*); the polarity of that group varies widely, from nonpolar and hydrophobic to highly polar and hydrophilic. Hydrophobic amino acid side chains tend to be clustered in the protein's interior, away from water, while hydrophilic side chains remain on the protein surface. The folding of a polypeptide chain thus creates an "inside" and an "outside" and generates buried and exposed amino acid side chains.

Hydrophobic interactions during protein folding do not exhibit the stacking phenomenon characterising RNA nucleotides. Therefore, the *hydrophobic/hydrophilic interaction of an amino acid* ($\mathcal{I}^h_{\mathtt{aa}}$ process) takes only one amino acid as input; if its side chain is hydrophilic (`hlsc`), it is exposed on the outside of the protein (`esc`), if it is hydrophobic (`hbsc`), it is buried inside the protein (`bsc`).

The states $\mathtt{I}_\mathtt{p}$ and $\mathtt{O}_\mathtt{p}$ identify the inside and outside of the protein, respectively. $\mathcal{I}^h_{\mathtt{aa}}$ is a subprocess of $\mathcal{F}^s_\mathtt{p}$.

The following is the CCS specification of the $\mathcal{I}^h_\mathtt{b}$ process:

$$\begin{array}{lcl} \mathcal{I}^h_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlsc}.\mathtt{O}_\mathtt{p} + \mathtt{hbsc}.\mathtt{I}_\mathtt{p}; \\ \mathtt{O}_\mathtt{p} & \stackrel{\text{def}}{=} & \overline{\mathtt{esc}}.\mathcal{F}^s_\mathtt{p}; \\ \mathtt{I}_\mathtt{p} & \stackrel{\text{def}}{=} & \overline{\mathtt{bsc}}.\mathcal{F}^s_\mathtt{p}. \end{array} \tag{A.6}$$

A.1.4 Folding step

Now that we have described the model of each non-covalent interaction in both RNA and protein, it is possible to contextualise these models in the folding step they belong to ($\mathcal{F}^s_{\mathtt{rna}}$ or $\mathcal{F}^s_\mathtt{p}$). Each step represents an iteration that allows the nondeterministic choice of one of the possible non-covalent interaction subprocesses. $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_\mathtt{p}$ ensure that each subprocess complies with the specific restrictions on its input (according to the specifications provided above) and that the corresponding interaction has a negative ΔG (i.e., it can be carried out).

The CCS specification of the whole $\mathcal{F}^s_{\mathtt{rna}}$ process is the following:

$$\begin{array}{lcl} \mathcal{F}^s_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \mathtt{ub}.\mathcal{I}1_\mathtt{n} + \mathtt{ub}.\mathcal{I}2_\mathtt{n} + \mathtt{srsr}.\mathcal{I}1_\mathtt{n} + \mathtt{drdr}.\mathcal{I}1_\mathtt{n} + \mathtt{srdr}.\mathcal{I}1_\mathtt{n} + \mathtt{tpb}.\mathcal{I}1_\mathtt{n}; \\ \mathcal{I}1_\mathtt{n} & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}} + \mathtt{srsr}.\Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}} + \mathtt{drdr}.\Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}} + \mathtt{srdr}.\Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}} + \mathtt{tpb}.\Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}}; \\ \mathcal{I}2_\mathtt{n} & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}2}} + \mathtt{ub}.\Delta\mathtt{G}_{\mathcal{I}^h_\mathtt{b}} + \mathtt{srsr}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}3}} + \mathtt{drdr}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}3}} + \mathtt{srdr}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}3}}; \\ \Delta\mathtt{G}_{\mathcal{I}^e_\mathtt{b}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^e_\mathtt{b}; \\ \Delta\mathtt{G}_{\mathcal{I}^h_\mathtt{b}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^h_\mathtt{b}; \\ \Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}2}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b}2}; \\ \Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b}3}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b}3}; \\ \\ \mathcal{P}_{\mathtt{b}2} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}1_{\mathtt{b}2}; \\ \mathcal{B}1_{\mathtt{b}2} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}2_{\mathtt{b}2}; \\ \mathcal{B}2_{\mathtt{b}2} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}3_{\mathtt{b}2} + \overline{\mathtt{srsr}}.\mathcal{F}^s_{\mathtt{rna}} + \overline{\mathtt{drdr}}.\mathcal{F}^s_{\mathtt{rna}} + \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}}; \\ \mathcal{B}3_{\mathtt{b}2} & \stackrel{\text{def}}{=} & \overline{\mathtt{srdr}}.\mathcal{F}^s_{\mathtt{rna}}; \end{array} \tag{A.7}$$

$$
\begin{array}{lcl}
\mathcal{P}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}1_{\mathtt{b3}}; \\
\mathcal{B}1_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}2_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{\mathtt{rna}}; \\
\mathcal{B}2_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}3_{\mathtt{b3}} + \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{\mathtt{rna}}; \\
\mathcal{B}3_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{\mathtt{rna}}; \\
 & & \\
\mathcal{I}^{e}_{\mathtt{b}} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^{s}_{\mathtt{rna}} + \overline{vdwi}.\mathcal{F}^{s}_{\mathtt{rna}}; \\
\mathcal{I}^{h}_{\mathtt{b}} & \stackrel{\text{def}}{=} & hbi.\mathtt{I}_{\mathtt{rna}}; \\
\mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bb}}.\mathcal{S}; \\
\mathcal{S} & \stackrel{\text{def}}{=} & \overline{\mathtt{sb}}.\mathcal{F}^{s}_{\mathtt{rna}}.
\end{array}
$$

$\mathcal{I}1_{\mathtt{n}}$ and $\mathcal{I}2_{\mathtt{n}}$ (nucleotide interaction) are states that allow the selection of the right subprocess based on its permitted inputs. The processes $\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b2}}}$ (ΔG of a base pairing), $\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b3}}}$ (ΔG of a triple base pairing), $\Delta\mathtt{G}_{\mathcal{I}^{e}_{\mathtt{b}}}$ (ΔG of an electrostatic interaction between bases), and $\Delta\mathtt{G}_{\mathcal{I}^{h}_{\mathtt{b}}}$ (ΔG of a hydrophobic interaction of bases) check that the ΔG of the related interaction is negative.

The CCS specification of the whole $\mathcal{F}^{s}_{\mathtt{p}}$ (*protein folding step*) process is:

$$
\begin{array}{lcl}
\mathcal{F}^{s}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\mathcal{I}1_{\mathtt{aa}} + \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{I}^{h}_{\mathtt{aa}}}; \\
\mathcal{I}1_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{I}^{e}_{\mathtt{aa}}} + \mathtt{aa}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}}; \\
\Delta\mathtt{G}_{\mathcal{I}^{e}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^{e}_{\mathtt{aa}}; \\
\Delta\mathtt{G}_{\mathcal{I}^{h}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^{h}_{\mathtt{aa}}; \\
\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa}}; \\
 & & \\
\mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{aa1fnh}.\mathtt{NH}_{\mathtt{aa1}} + \mathtt{aa1fco}.\mathtt{CO}_{\mathtt{aa1}}; \\
\mathtt{NH}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fco}.\mathtt{CO}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa1}} & \stackrel{\text{def}}{=} & \mathtt{aa2fnh}.\mathtt{NH}_{\mathtt{aa2}}; \\
\mathtt{CO}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathtt{NH}_{\mathtt{aa2}} & \stackrel{\text{def}}{=} & \mathtt{hb}.\mathcal{B}_{\mathtt{aa}}; \\
\mathcal{B}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{\mathtt{paa}}.\mathcal{F}^{s}_{\mathtt{p}}; \\
 & & \\
\mathcal{I}^{e}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^{s}_{\mathtt{p}} + \overline{vdwi}.\mathcal{F}^{s}_{\mathtt{p}}; \\
\mathcal{I}^{h}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlsc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbsc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{esc}}.\mathcal{F}^{s}_{\mathtt{p}}; \\
\mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bsc}}.\mathcal{F}^{s}_{\mathtt{p}}.
\end{array}
\tag{A.8}
$$

$\mathcal{I}1_{\mathtt{aa}}$ is a state that allows the selection of the subprocesses that take two amino acids as input. The processes $\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{aa}}}$ (ΔG of an amino acid pairing), $\Delta\mathtt{G}_{\mathcal{I}^{e}_{\mathtt{aa}}}$ (ΔG of an electrostatic interaction between

amino acids), and $\Delta G_{\mathcal{J}^h_{aa}}$ (ΔG of a hydrophobic/hydrophilic interaction of an amino acid) check that the ΔG of the related interaction is negative.

A.1.5 RNA folding and protein folding

To meet the requirement that each interaction must have a negative ΔG, both the $\mathcal{F}^s_{\mathtt{rna}}$ and $\mathcal{F}^s_{\mathtt{p}}$ processes are placed in parallel composition with the ΔG process, defining in this way the overall folding process ($\mathcal{F}_{\mathtt{rna}}$ and $\mathcal{F}_{\mathtt{p}}$ respectively). This is formally stated in Definition 2.2 and the related Equation 2.2, which we write here again to complete the model construction.

$$\begin{aligned}
&\mathcal{F}_{\mathtt{rna}} \stackrel{\text{def}}{=} (\mathcal{F}^s_{\mathtt{rna}} | \Delta G) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\mathcal{F}_{\mathtt{p}} \stackrel{\text{def}}{=} (\mathcal{F}^s_{\mathtt{p}} | \Delta G) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};\\
&\text{where } \Delta G \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta G + \overline{\mathtt{ndg}}.\Delta G + \overline{\mathtt{zdg}}.\Delta G.
\end{aligned} \tag{A.9}$$

A.1.6 Model checking

It is possible to verify that the models described above meet the biochemical properties of the folding processes. For this purpose, we can represent such properties as HML formulas and perform model checking to establish if they are satisfied. We propose here four examples:

1. two unpaired bases (ub) can form a hydrogen bond (hb) if the ΔG of the interaction is negative (ndg):
$$\mathcal{F}^s_{\mathtt{rna}} \vDash \langle \mathtt{ub} \rangle \langle \mathtt{ub} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hb} \rangle \mathbf{tt}\,; \tag{A.10}$$
2. with a single hydrogen bond it is not possible to form a base pair (srsr, drdr, srdr):
$$\mathcal{P}_{b2} \vDash \langle \mathtt{hb} \rangle ([\overline{\mathtt{srsr}}]\mathbf{ff} \wedge [\overline{\mathtt{srdr}}]\mathbf{ff} \wedge [\overline{\mathtt{drdr}}]\mathbf{ff})\,; \tag{A.11}$$
3. it is possible to form a group of three paired bases (tpb) with only a single hydrogen bond (between an unpaired base and a group of two already paired bases - srsr in this case); obviously, the ΔG of the interaction must be negative:
$$\mathcal{F}^s_{\mathtt{rna}} \vDash \langle \mathtt{ub} \rangle \langle \mathtt{srsr} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hb} \rangle \langle \overline{\mathtt{tpb}} \rangle \mathbf{tt}\,; \tag{A.12}$$
4. if an amino acid has a hydrophobic side chain (hbsc), it has to be buried inside (bsc) and not exposed outside (esc) the protein:
$$\mathcal{F}^s_{\mathtt{p}} \vDash \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle (\langle \overline{\mathtt{bsc}} \rangle \mathbf{tt} \wedge [\overline{\mathtt{esc}}]\mathbf{ff})\,; \tag{A.13}$$

The verification that these formulas are satisfied was made with the aid of the model checking function of the CAAL concurrency workbench [3]. The results are shown in Figure A.1.

Status	Time	Property	Verify
✓	25 ms	RNAFS ⊨ <ub><ub><ndg><hb>tt	▶
✓	25 ms	BP ⊨ <hb>(['srsr]ff and ['srdr]ff and ['drdr]ff)	▶
✓	25 ms	RNAFS ⊨ <ub><srsr><ndg><hb><'tpb>tt	▶
✓	25 ms	PFS ⊨ <aa><ndg><hbsc>(<'bsc>tt and ['esc]ff)	▶

Figure A.1 – Verification of some biochemical properties, expressed as HML formulas, performed by CAAL concurrency workbench [3]. The checkmarks on the "Status" column indicate that all the formulas are satisfied. The $\mathcal{F}^s_{\mathtt{rna}}$, $\mathcal{P}_{\mathtt{b2}}$, and $\mathcal{F}^s_{\mathtt{p}}$ processes are transliterated RNAFS, BP, and PFS, respectively (see Table 2.1).

A.1.7 High abstraction level model

We might therefore wonder if *there is an abstraction level at which the two folding processes would show a behavioural equivalence.* As proved in Chapter 2, this level of abstraction can actually be defined. Its construction, however, requires generalising the non-covalent interactions and imposing some limitations on the expressiveness of the protein folding process.

The first of the two modifications mentioned above can be achieved by:

- redefining nucleotides and the amino acids as general elementary units, which can be paired or unpaired;
- abstracting from the specificity of each pairing process by no longer taking into account the number of hydrogen bonds formed between two (or three) paired units;
- generalising the hydrophobic interactions to their key feature of burying the hydrophobic molecules while exposing the hydrophilic ones (no longer considering the stacking process typical of the hydrophobic interactions of nucleotides).

These adjustments to the model do not affect the main properties of each non-covalent interaction; therefore, the model is still fairly faithful to the biological process. However, they are also not sufficient to obtain a behavioural equivalence between the folding processes of RNAs and proteins.

We still need to limit the folding capability of the proteins by reducing the number of amino acids that can interact through hydrogen bonds to the number of three (the maximum number of nucleotides that can pair in RNAs).

With these considerations in mind, we can rewrite the above model of the folding process.This transformation is formally carried out by the *folding step high abstraction function* $\mathcal{H} : \mathsf{P} \to \mathsf{P}$ defined in Equation 2.3, which generates the two *high abstraction folding steps* $\mathbb{F}^s_{\mathtt{rna}}$ and $\mathbb{F}^s_{\mathtt{p}}$. They are composed of the following subprocesses.

Base pairing

Base pairing is modelled through the $\mathcal{P}_{\mathtt{b2}}$ process, which takes two unpaired units (uu) as input (from the $\mathbb{F}^{s}_{\mathtt{rna}}$ process) and produces a paired unit (pu) as output. It should be noted that *the label hb does not indicate a single hydrogen bond but represents the whole interaction based on hydrogen bonding.*

$$\begin{array}{lcl} \mathcal{P}_{\mathtt{b2}} & \stackrel{\text{def}}{=} & hb.\mathtt{B_{sr}B_{sr}} + hb.\mathtt{B_{dr}B_{dr}} + hb.\mathtt{B_{sr}B_{dr}}; \\ \mathtt{B_{sr}B_{sr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^{s}_{\mathtt{rna}}; \\ \mathtt{B_{dr}B_{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^{s}_{\mathtt{rna}}; \\ \mathtt{B_{sr}B_{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^{s}_{\mathtt{rna}}. \end{array} \tag{A.14}$$

$\mathtt{B_{sr}B_{sr}}$, $\mathtt{B_{dr}B_{dr}}$, $\mathtt{B_{sr}B_{dr}}$ are states that specify the type of base pair of the produced paired unit.

Triple base pairing

The triple base pairing is performed by the $\mathcal{P}_{\mathtt{b3}}$ process, taking an unpaired unit (uu) and a paired unit (pu) as input (from the $\mathbb{F}^{s}_{\mathtt{rna}}$ process) and producing a triple unit (tpu) as output.

$$\begin{array}{lcl} \mathcal{P}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & hb.\mathtt{U_{b3}}; \\ \mathtt{U_{b3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}^{s}_{\mathtt{rna}}. \end{array} \tag{A.15}$$

The state $\mathtt{U_{b3}}$ (base triple unit) indicates that a hydrogen bonding interaction (possibly involving more than one hydrogen bond) has occurred.

Amino acid pairing

Amino acid pairing is produced through the $\mathcal{P}_{\mathtt{aa}}$ process; it takes two unpaired units (uu) as input (from the $\mathbb{F}^{s}_{\mathtt{p}}$ process) and generates a paired unit (pu) as output. As with the $\mathcal{P}_{\mathtt{b2}}$ process, *the label hb does not indicate the formation of a single hydrogen bond but a generalised hydrogen bonding interaction.*

$$\begin{array}{lcl} \mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & hb.\mathtt{NC} + hb.\mathtt{CN}; \\ \mathtt{NC} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^{s}_{\mathtt{p}}; \\ \mathtt{CN} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^{s}_{\mathtt{p}}. \end{array} \tag{A.16}$$

The states NC and CN (where N and C represent a free *amino group* and a free *carboxyl group*, respectively) allow the preservation of the correct complementarity of the hydrogen bonding between amino acids.

Triple amino acid pairing

Triple amino acid pairing is a process not present in the original protein folding model; it is necessary to limit amino acids' capabilities to form hydrogen bonds with each other. As in the case of base pairing, at most three amino acids can be connected via the same hydrogen bonding interaction (not to be confused with a single hydrogen bond).

This type of interaction is carried out by the $\mathcal{P}_{\mathtt{aa3}}$ process, which takes an unpaired unit (`uu`) and a paired unit (`pu`) as input and produces a triple unit (`tpu`) as output.

$$\begin{array}{lcl} \mathcal{P}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & hb.\mathtt{U}_{\mathtt{aa3}}; \\ \mathtt{U}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}^{s}_{\mathtt{p}}. \end{array} \tag{A.17}$$

Electrostatic interaction

The electrostatic interaction between bases ($\mathcal{I}^{e}_{\mathtt{b}}$ process) and the electrostatic interaction between amino acids ($\mathcal{I}^{e}_{\mathtt{aa}}$ process) are unchanged compared with the original model (see Section A.1.2).

Hydrophobic/hydrophilic interaction of a nucleotide

Since hydrophobic stacking is no longer considered in the new model, the hydrophobic interaction can affect a single nucleotide per folding step.

The process, renamed $\mathcal{I}^{h}_{\mathtt{n}}$, takes one unpaired unit as input and, through the actions `hbc` and `bc`, indicates that its hydrophobic component is buried inside the RNA; conversely, the actions `hlc` and `ec` denote that the hydrophilic component is exposed on the outside of the molecule.

$$\begin{array}{lcl} \mathcal{I}^{h}_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{rna}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{rna}} \\ \mathtt{O}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}^{s}_{\mathtt{rna}}; \\ \mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}^{s}_{\mathtt{rna}}. \end{array} \tag{A.18}$$

Hydrophobic/hydrophilic interaction of an amino acid

Similarly to the previous process, $\mathcal{I}^{h}_{\mathtt{aa}}$ takes one unpaired unit as input and indicates that its hydrophobic component (`hbc` and `bc`) is pushed inside the protein while the hydrophilic one (`hlc` and `ec`) is exposed on the outside. In this case, the "component" is a generalisation of the side chain; this means that each unpaired unit taken as input can have a hydrophobic or a

hydrophilic component (but not both).

$$\begin{array}{lcl} \mathcal{I}^h_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{p}}; \\ \mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}^s_{\mathtt{p}}; \\ \mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}^s_{\mathtt{p}}. \end{array} \tag{A.19}$$

Folding step

The $\mathbb{F}^s_{\mathtt{rna}}$ and $\mathbb{F}^s_{\mathtt{p}}$ perform the same tasks as in the original model (see Section A.1.4 on page 140). The CCS specification of the whole high abstraction $\mathbb{F}^s_{\mathtt{rna}}$ process is

$$\begin{array}{lcl} \mathbb{F}^s_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\mathcal{J}1_{\mathtt{n}} + \mathtt{pu}.\mathcal{J}1_{\mathtt{n}} + \mathtt{uu}.\Delta\mathtt{G}_{\mathcal{J}^h_{\mathtt{n}}} + \mathtt{uu}.\mathcal{J}2_{\mathtt{n}} + \mathtt{tpu}.\mathcal{J}1_{\mathtt{n}}; \\ \mathcal{J}1_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{pu}.\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{b}}} + \mathtt{tpu}.\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{b}}}; \\ \mathcal{J}2_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b2}}} + \mathtt{pu}.\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b3}}}; \\ \Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{b}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^e_{\mathtt{b}}; \\ \Delta\mathtt{G}_{\mathcal{J}^h_{\mathtt{n}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{J}^h_{\mathtt{n}}; \\ \Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b2}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b2}}; \\ \Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b3}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{b3}}; \\ \\ \mathcal{P}_{\mathtt{b2}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{sr}} + \mathit{hb}.\mathtt{B}_{\mathtt{dr}}\mathtt{B}_{\mathtt{dr}} + \mathit{hb}.\mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{dr}}; \\ \mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{sr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; \\ \mathtt{B}_{\mathtt{dr}}\mathtt{B}_{\mathtt{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; \\ \mathtt{B}_{\mathtt{sr}}\mathtt{B}_{\mathtt{dr}} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}^s_{\mathtt{rna}}; \\ \\ \mathcal{P}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{U}_{\mathtt{b3}}; \\ \mathtt{U}_{\mathtt{b3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}^s_{\mathtt{rna}}. \\ \\ \mathcal{J}^e_{\mathtt{b}} & \stackrel{\text{def}}{=} & \overline{\mathit{ii}}.\mathbb{F}^s_{\mathtt{rna}} + \overline{\mathit{vdwi}}.\mathbb{F}^s_{\mathtt{rna}}; \\ \mathcal{J}^h_{\mathtt{n}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{rna}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{rna}}; \\ \mathtt{O}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}^s_{\mathtt{rna}}; \\ \mathtt{I}_{\mathtt{rna}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}^s_{\mathtt{rna}}. \end{array} \tag{A.20}$$

$\mathcal{J}1_{\mathtt{n}}$ and $\mathcal{J}2_{\mathtt{n}}$ (nucleotide interaction) are states that allow the selection of the right subprocess based on its permitted inputs.

The processes $\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b2}}}$ (ΔG of a base pairing), $\Delta\mathtt{G}_{\mathcal{P}_{\mathtt{b3}}}$ (ΔG of a triple base pairing), $\Delta\mathtt{G}_{\mathcal{J}^e_{\mathtt{b}}}$ (ΔG of an electrostatic interaction between bases), and $\Delta\mathtt{G}_{\mathcal{J}^h_{\mathtt{n}}}$ (ΔG of a hydrophobic/hydrophilic interaction of a nucleotide) check that the ΔG of the related interaction is negative.

The CCS specification of the whole high abstraction $\mathbb{F}_{\mathtt{p}}^{s}$ process is the following:

$$
\begin{array}{lcl}
\mathbb{F}_{\mathtt{p}}^{s} & \stackrel{\text{def}}{=} & \mathtt{uu}.\mathcal{I}1_{\mathtt{aa}} + \mathtt{pu}.\mathcal{I}1_{\mathtt{aa}} + \mathtt{uu}.\Delta G_{\mathcal{I}^{h}_{\mathtt{aa}}} + \mathtt{uu}.\mathcal{I}2_{\mathtt{aa}} + \mathtt{tpu}.\mathcal{I}1_{\mathtt{aa}}; \\
\mathcal{I}1_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{I}^{e}_{\mathtt{aa}}} + \mathtt{pu}.\Delta G_{\mathcal{I}^{e}_{\mathtt{aa}}} + \mathtt{tpu}.\Delta G_{\mathcal{I}^{e}_{\mathtt{aa}}}; \\
\mathcal{I}2_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{uu}.\Delta G_{\mathcal{P}_{\mathtt{aa}}} + \mathtt{pu}.\Delta G_{\mathcal{P}_{\mathtt{aa3}}}; \\
\Delta G_{\mathcal{I}^{e}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^{e}_{\mathtt{aa}}; \\
\Delta G_{\mathcal{I}^{h}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^{h}_{\mathtt{aa}}; \\
\Delta G_{\mathcal{P}_{\mathtt{aa}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa}}; \\
\Delta G_{\mathcal{P}_{\mathtt{aa3}}} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{P}_{\mathtt{aa3}}; \\
 & & \\
\mathcal{P}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{NC} + \mathit{hb}.\mathtt{CN}; \\
\mathtt{NC} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}_{\mathtt{p}}^{s}; \\
\mathtt{CN} & \stackrel{\text{def}}{=} & \overline{\mathtt{pu}}.\mathbb{F}_{\mathtt{p}}^{s}; \\
 & & \\
\mathcal{P}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & \mathit{hb}.\mathtt{U}_{\mathtt{aa3}}; \\
\mathtt{U}_{\mathtt{aa3}} & \stackrel{\text{def}}{=} & \overline{\mathtt{tpu}}.\mathbb{F}_{\mathtt{p}}^{s}; \\
 & & \\
\mathcal{I}^{e}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \overline{\mathit{ii}}.\mathbb{F}_{\mathtt{p}}^{s} + \overline{\mathit{vdwi}}.\mathbb{F}_{\mathtt{p}}^{s}; \\
\mathcal{I}^{h}_{\mathtt{aa}} & \stackrel{\text{def}}{=} & \mathtt{hlc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{ec}}.\mathbb{F}_{\mathtt{p}}^{s}; \\
\mathtt{I}_{\mathtt{p}} & \stackrel{\text{def}}{=} & \overline{\mathtt{bc}}.\mathbb{F}_{\mathtt{p}}^{s}.
\end{array}
\tag{A.21}
$$

$\mathcal{I}1_{\mathtt{aa}}$ and $\mathcal{I}2_{\mathtt{aa}}$ are states that allow the selection of the right subprocess on the basis of its permitted inputs. The processes $\Delta G_{\mathcal{P}_{\mathtt{aa}}}$ (ΔG of an amino acid pairing), $\Delta G_{\mathcal{I}^{e}_{\mathtt{aa}}}$ (ΔG of an electrostatic interaction between amino acids), and $\Delta G_{\mathcal{I}^{h}_{\mathtt{aa}}}$ (ΔG of a hydrophobic/hydrophilic interaction of an amino acid) check that the ΔG of the related interaction is negative.

The whole high abstraction folding processes $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$ are defined as the parallel composition of the folding step process and the folding step ΔG (see Equation 2.5).

In Chapter 2, we prove the existence of a congruence relation between $\mathbb{F}_{\mathtt{rna}}^{s}$ and $\mathbb{F}_{\mathtt{p}}^{s}$ and, consequently, between $\mathbb{F}_{\mathtt{rna}}$ and $\mathbb{F}_{\mathtt{p}}$ (see Theorems 2.1 and 2.2).

Appendix B

Supplementary Information to Chapter 3

An Algebraic Approach to the Study of Protein Misfolding

B.1 Formal Description of HBB Gene Expression

In this appendix, the behaviour of the gene expression process is specified using Hennessy-Milner logic (HML) formulae [54]. More precisely, we show how the HBB gene, which codes for one of the β subunits of the haemoglobin molecule, is expressed through the processes described in Section 3.2.1, that is, $\mathcal{T}$ (transcription), $\mathcal{P}$ (processing), and $\mathcal{L}$ (translation). Each of these processes satisfies the related HML formula of the HBB gene expression.

The DNA sequence of the HBB gene (1742 nucleotides long) has been derived from an HBB transcript variant (1742 nucleotides) [82], which we retrieved from the National Center for Biotechnology Information (NCBI) AceView website [115]; the gene contains three exons (coloured in green in their coding regions) and two introns (coloured in blue). We highlight in red the codon that codes for the Glu 6 of the β subuint amino acid sequence.

The formulae are too long to be entirely displayed in this section; therefore, we show only their beginning part (one or two rows), their middle part, where the codon of the Glu 6 is present, and their ending rows.

The process starts from the string δ_{hbb} = "𝕡" γ_{hbb} "𝕥", where

γ_{hbb} ="gccgacagtagtgaatctggagtgggacacctcggtgtgggatcccaaccggttagatgagggtc
ctcgtccctcccgtcctcggtcccgacccgtattttcagtcccgtctcggtagataacgaatgtaaacgaa
gactgtgttgacacaagtgatcgttggagtttgtctgtggtaccacgtagactgaggagtgctcttcagac
ggcaatgacgggacaccccgttccacttgcacctacttcaaccaccactccgggacccgtccaaccatagt
tccaatgttctgtccaaattcctctggttatctttgacccgtacacctctgtctcttctgagaacccaaag
actatccgtgactgagagagacggataaccagataaaagggtgggaatccgacgaccaccagatgggaacc
tgggtctccaagaaactcaggaaacccctagacaggtgaggactacgacaatacccgttgggattccactt
ccgagtaccgttctttcacgagccacggaaatcactaccggaccgagtggacctgttggagttcccgtgga
aacggtgtgactcactcgacgtgacactgttcgacgtgcacctaggactcttgaagtcccactcagatacc
ctgcgaactacaaaagaaaggggaagaaaagataccaattcaagtacagtatccttcccctattcattgtc
ccatgtcaaatcttacccttgtctgcttactaacgtagtcacaccttcagagtcctagcaaaatcaaaga
aaataaacgacaagtattgttaacaaaagaaaacaaattaagaacgaaagaaaaaaaaagaagaggcgtta
aaaatgataatatgaattacggaattgtaacacatattgttttccttatagagactctatgtaattcatt
gaatttttttttgaaatgtgtcagacggatcatgtaatgataaaccttatatacacacgaataaacgtata
agtattagagggatgaaataaaagaaaataaaaattaactatgtattagtaatatgtataaatacccaatt
tcacattacaaaattatacacatgtgtataactggtttagtcccattaaaacgtaaacattaaaatttttt
acgaaagaagaaaattatatgaaaaaacaaatagaataaagattatgaaagggattagagaaagaaagtcc
cgttattactatgttacatagtacggagaaacgtggtaagatttcttattgtcactattaaagacccaatt
ccgttatcgttatagagacgtatatttataaagacgtatatttaacattgactacattctccaaagtataa
cgattatcgtcgatgttaggtcgatggtaagacgaaaataaaataccaaccctattccgacctaataagac
tcaggttcgatccgggaaaacgattagtacaagtatggagaatagaaggagggtgtcgaggacccgttgca
cgaccagacacacgaccgggtagtgaaaccgtttcttaagtggggtggtcacgtccgacggatagtctttc
accaccgaccacaccgattacgggaccgggtgttcatagtgattcgagcgaaagaacgacaggttaaagat
aatttccaaggaaacaagggattcaggttgatgatttgacccccctataatacttcccggaactcgtagacc
taagacggattatttttgtaaataaaagtaacgttactacata"

(B.1)

B.1.1 Transcription

By extending Equation 3.10 to the whole transcription process $\mathcal{T}$, we obtain that:

$$\begin{aligned} &\mathcal{T} \vDash \langle \mathbb{p} \rangle \langle \overline{5} \rangle \mathcal{T}_r \\ &\mathcal{T}_r \equiv \langle b_1 \rangle \langle b_2 \rangle \langle \overline{b_1 b_2} \rangle \langle b_1 b_2 \rangle (\langle \overline{b_2} \rangle \mathbf{tt} \wedge \langle \overline{b_2} \rangle \mathcal{T}_r \wedge \langle \overline{b_2} \rangle \langle \mathbb{t} \rangle \langle \overline{3} \rangle \mathbf{tt}) \end{aligned} \tag{B.2}$$

We can apply Equation B.2 to formally describe the HBB gene transcription on the basis of its DNA sequence; the latter is represented as the γ_{hbb} string of Equation B.1.

$$
\begin{aligned}
&\mathcal{T} \vDash \\
&\langle \mathtt{p} \rangle \langle \overline{5} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \\
&\langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \\
&\vdots \\
&\langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \ \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \ \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \\
&\vdots \\
&\langle \mathtt{g} \rangle \langle \overline{gc} \rangle \langle gc \rangle \langle \overline{\mathtt{c}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \\
&\langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{c} \rangle \langle \overline{cg} \rangle \langle cg \rangle \langle \overline{\mathtt{g}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{t} \rangle \langle \overline{ta} \rangle \langle ta \rangle \langle \overline{\mathtt{a}} \rangle \langle \mathtt{a} \rangle \langle \overline{au} \rangle \langle au \rangle \langle \overline{\mathtt{u}} \rangle \langle \mathtt{t} \rangle \langle \overline{3} \rangle \mathbf{tt};
\end{aligned}
\tag{B.3}
$$

After the $\mathcal{T}$ process has occurred, we obtain the string $\chi_{hbb} =$ "5" θ_{hbb} "3", where

$$
\begin{aligned}
\theta_{hbb} = {} & \texttt{"cggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccag} \\
& \texttt{gagcagggagggcaggagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugcuu} \\
& \texttt{cugacacaacuguguucacuagcaaccucaaacagacaccauggugcaucugacuccugaggagaagucug} \\
& \texttt{ccguuacugcccuguggggcaaggugaacguggaugaaguugguggugaggcccugggcaggugguauca} \\
& \texttt{agguuacaagacagguuuaaggagaccaauagaaacugggcauguggagacagagaagacucuuggguuuc} \\
& \texttt{ugauaggcacugacucucucugccuauuggucuauuuucccacccuuaggcugcuggugguctuacccuugg} \\
& \texttt{acccagagguucuuugaguccuuuggggaucuguccacuccugaugcuguuaugggcaacccuaaggugaa} \\
& \texttt{ggcucauggcaagaaagugcucggugccuuuagugauggccuggcucaccuggacaaccucaagggcaccu} \\
& \texttt{uugccacacugagugagcugcacugugacaagcugcacguggauccugagaacuucagggugagucuaugg} \\
& \texttt{gacgcuugauguuuucuuucccuucuuuucuaugguuaaguucaugucauaggaaggggauaaguaacag} \\
& \texttt{gguacaguuuagaaugggaaacagacgaaugauugcaucaguguggaagucucaggaucguuuuaguuucu} \\
& \texttt{uuuauuugcuguucauaacaauuguuuucuuuuguuuaauucuugcuuucuuuuuuuuucuucuccgcaau} \\
& \texttt{uuuuacuauuauacuuaaugccuuaacauuguguauaacaaaaggaaauaucucugagauacauuaaguaa} \\
& \texttt{cuuaaaaaaaaacuuuacacagucugccuaguacauuacuauuuggaauauaugugugcuuauuugcauau} \\
& \texttt{ucauaaucucccuacuuuauuuucuuuuauuuuuaauugauacauaaucauuauacauauuuauggguuaa} \\
& \texttt{aguguaauguuuuaauauguguacacauauugaccaaaucaggguaauuuugcauuuguaauuuuaaaaaa} \\
& \texttt{ugcuuucuucuuuuaauauacuuuuuuguuuaucuuauuucuaauacuuucccuaaucucuuucuuucagg} \\
& \texttt{gcaauaaugauacaauguaucaugccucuuugcaccauucuaaagaauaacagugauaauuucuggguuaa} \\
& \texttt{ggcaauagcaauaucucugcauauaaauauuucugcauauaaauuguaacugauguaagagguuucauauu} \\
& \texttt{gcuaauagcagcuacaauccagcuaccauucugcuuuuauuuuaugguuggggauaaggcuggauuauucug} \\
& \texttt{aguccaagcuaggcccuuuugcuaaucauguucauaccucuuaucuuccucccacagcuccuggggcaacgu} \\
& \texttt{gcuggucugugugcuggcccaucacuuuggcaaagaauucaccccaccagugcaggcugccuaucagaaag} \\
& \texttt{ugguggcuggugugguaaugcccuggcccacaaguaucacuaagcucgcuuucuugcuguccaauuucua} \\
& \texttt{uuaaagguuccuuuguucccuaaguccaacuacuaaacuggggguauauuaugaagggccuugagcaucugg} \\
& \texttt{auucugccuaauaaaaaacauuuauuuucauugcaaugauguau"}
\end{aligned}
\tag{B.4}
$$

The starting and ending substrings of each intron ("gu" and "ag") are highlighted in orange. They are identified during the *processing* phase of HBB expression.

B.1.2 Processing

Considering Equation 3.16 in the context of the whole $\mathcal{P}$ process, we obtain that

$$\begin{aligned}
&\mathcal{P} \vDash \langle 5\rangle\langle \overline{\mathtt{c}}\rangle \mathcal{S}_r \wedge \langle 3\rangle\langle \overline{\mathtt{a}}\rangle \mathbf{tt} \\
&\mathcal{S}_r \equiv \langle b\rangle\langle \overline{b}\rangle \mathcal{S}_r \wedge \langle \mathtt{g}\rangle\langle \mathtt{u}\rangle \mathcal{E} \\
&\mathcal{E} \equiv \langle b\rangle \mathcal{E} \wedge \langle \mathtt{a}\rangle(\langle \mathtt{g}\rangle \mathbf{tt} \wedge \langle \mathtt{g}\rangle \mathcal{S}_r)
\end{aligned} \tag{B.5}$$

Therefore, by applying $\mathcal{P}$ to the string χ_{hbb}, the process satisfies the following specification:

$\mathcal{P} \vDash$
$\langle 5\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle$
$\vdots$
$\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle$
$\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle$
$\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle$
$\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle$
$\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle$
$\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle$
$\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{u}\rangle$
$\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{a}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{c}\rangle\langle \mathtt{u}\rangle\langle \mathtt{u}\rangle\langle \mathtt{a}\rangle\langle \mathtt{g}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle$
$\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\vdots$
$\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{c}\rangle\langle \overline{\mathtt{c}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle$
$\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{g}\rangle\langle \overline{\mathtt{g}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle \mathtt{a}\rangle\langle \overline{\mathtt{a}}\rangle\langle \mathtt{u}\rangle\langle \overline{\mathtt{u}}\rangle\langle 3\rangle\langle \overline{\mathtt{a}}\rangle \mathbf{tt}$;

(B.6)

Consequently, the result of $\mathcal{P}$ on χ_{hbb} is the *mRNA string* represented by

ρ_{hbb} ="ccggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccc
aggagcagggagggcaggagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugc
uucugacacaacuguguucacuagcaaccucaaacagacaccauggugcaucugacuccugaggagaaguc
ugccguuacugcccuguggggcaaggugaacguggaugaaguugguggugaggccccugggcaggcugcugg
uggucuacccuuggacccagagguucuuugaguccuuuggggaucuguccacuccugaugcuguuaugggc
aacccuaaggugaaggcucauggcaagaaagugcucggugccuuuagugauggccuggcucaccuggacaa
ccucaagggcaccuuugccacacugagugagcugcacugugacaagcugcacguggauccugagaacuuca
ggcuccugggcaacgugcuggucugugugcuggcccaucacuuuggcaaagaauucaccccaccagugcag
gcugccuaucagaaagugguggcuggugugguggcuaaugcccuggcccacaaguaucacuaagcucgcuuucu
ugcuguccaauuucuauuaaaggguuccuuuguucccuaaguccaacuacuaaacuggggggauauuaugaag
ggccuugagcaucuggauucugccuaauaaaaaacauuuauuuucauugcaaugauguauaa"

(B.7)

We coloured in orange the start codon "aug" and the stop codon "uaa" since they are taken as starting and ending points to carry out the *translation* of ρ_{hbb} into the β-globin amino acid sequence.

B.1.3 Translation

If we relax the length constraint of Equation 3.21 through recursion, we obtain that

$$\begin{aligned}
&\mathcal{L} \vDash \langle \mathtt{c} \rangle \mathcal{L}_r \\
&\mathcal{L}_r \equiv \langle b \rangle \mathcal{L}_r \wedge \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{imet}} \rangle \mathcal{C} \\
&\mathcal{C} \equiv \langle b_1 \rangle \langle b_2 \rangle \langle b_3 \rangle (\langle \overline{a} \rangle \mathbf{tt} \wedge \langle \overline{a} \rangle \mathcal{C}) \wedge \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt} \\
&\qquad \wedge \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt} \wedge \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt}
\end{aligned} \tag{B.8}$$

We can thus apply $\mathcal{L}$ to ρ_{hbb}:

$$\begin{aligned}
&\mathcal{L} \vDash \\
&\langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \\
&\langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \\
&\langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \\
&\vdots \\
&\langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{imet}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{val}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{his}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{leu}} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{thr}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \\
&\langle \overline{\mathtt{pro}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{glu}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{glu}} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{lys}} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{ser}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{ala}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \\
&\langle \mathtt{u} \rangle \langle \overline{\mathtt{val}} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{thr}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{ala}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{leu}} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{trp}} \rangle \\
&\vdots \\
&\langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{leu}} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{ala}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{his}} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \overline{\mathtt{lys}} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \overline{\mathtt{tyr}} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \overline{\mathtt{his}} \rangle \\
&\langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \overline{\mathtt{s}} \rangle \mathbf{tt};
\end{aligned} \tag{B.9}$$

As a result, the amino acid sequence of the haemoglobin β subunit is represented by the string

$$\begin{aligned}
&\psi_{hbb} = \texttt{"imet val his leu thr pro glu glu lys ser ala val thr ala leu trp} \\
&\texttt{gly lys val asn val asp glu val gly gly glu ala leu gly arg leu leu val} \\
&\texttt{val tyr pro trp thr gln arg phe phe glu ser phe gly asp leu ser thr pro} \\
&\texttt{asp ala val met gly asn pro lys val lys ala his gly lys lys val leu gly} \\
&\texttt{ala phe ser asp gly leu ala his leu asp asn leu lys gly thr phe ala thr} \\
&\texttt{leu ser glu leu his cys asp lys leu his val asp pro glu asn phe arg leu} \\
&\texttt{leu gly asn val leu val cys val leu ala his his phe gly lys glu phe thr} \\
&\texttt{pro pro val gln ala ala tyr gln lys val val ala gly val ala asn ala leu} \\
&\texttt{ala his lys tyr his"}
\end{aligned} \tag{B.10}$$

Similarly to what is shown in Section 3.3, all the specifications in this appendix can be verified with the aid of the CAAL concurrency workbench [3].

Appendix C

Supplementary Information to Chapter 6

Detecting In Silico the Driving Forces of Biomolecular Interactions

C.1 Plots of the Concentration Changes during the Agent-based Simulations

This appendix provides some of the plots generated for the study proposed in Chapter 6. We report the concentration changes over time of the metabolites (Section C.1.1) and a selection of the complex formations (Section C.1.2). We choose specifically these plots since they are relevant to highlight the differences between the electrostatic and electromagnetic potentials modelled through our agent-based approach.

We ran the simulations on cloud-based virtual machines powered by 8 vCPUs and 32 GB of memory. With these hardware resources, simulating 0.1 seconds requires roughly 24 hours. For this reason, we chose 1 second (about ten days of simulation) as the standard time interval for our study. Even if it could be too short for observing some biological phenomena, such as the effects of enzyme activations and inhibitions, it turned out to be sufficient to highlight the impact of the long-distance electrodynamic interactions on the glycolytic pathway.

In what follows, we refer to each type of simulation as:

- `300 Å simulation`, representing a system where molecular interactions are driven by long-range forces;
- `10 Å simulation`, based on the model of a system where the Debye screening limits the capability of biomolecules to perceive each other;

- `5 Å simulation`, reproducing a biochemical system whose reactions rely only on random encounters and chemical affinity.

This convention is based on the *perception distances* that characterise the space inside which an enzyme can identify a cognate metabolite, as explained in Section 6.2.2 on page 107.

C.1.1 Metabolite concentration changes

The following plots represent the molar concentration changes, observable in the interval of 1 second, of the metabolites simulated during our study. We exclude from the plots the `AMP` and the `F26bP` because, although they are present in the simulated environment, they are involved in enzyme regulation, a feature we have not yet modelled; their concentrations, thus, remain constant. To make each plot more readable, we define four subsets of metabolites (a complete list of which can be found in Table 6.1). According to the Smallbone2013 - Iteration 18 model [107], all of them are already present in the environment at the beginning of the glycolytic process.

Every kind of simulation produces plots with common properties; therefore, we prefer to describe and compare such features in this introductory discussion instead of providing redundant captions for each figure.

Observing the plots, we can notice the high rates through which the `300 Å simulation` produces or consumes the metabolites in the environment; this behaviour is less evident in the `10 Å simulation` and almost absent in the `5 Å simulation`. We largely discuss the reasons behind these phenomena in Section 6.3 on page 110, where, in particular, we explain the concentration variations of `ADP`, `ATP`, `NADH`, and `F16bP`; their plots are shown in the following Figures C.1 and C.3. To summarise some of our findings, we can say that, by limiting the molecular interactions to those allowed by short-range van der Waals-like potentials (5 Å perception distance), most enzymes cannot bind part of their substrate. For example, in the plots generated by the `5 Å simulation`, the number of `ATP` molecules can only increase because, during the preparation phase of glycolysis, neither the `hexokinases` (`HXK`)–and `glucokinases` (`GLK`)–nor the `phosphofructokinases` (`PFK`) are able to bind this metabolite and complete the catalysis of their respective reactions; `ATP` is instead produced in the payoff phase. This particular aspect is better shown in the next subsection, which provides the plots of the complexes formation. Similarly, the concentration of `NADH` never changes during the entire interval of the `5 Å simulation` because the `glycerol-3-phosphate dehydrogenase` (`GDP`) is not able to bind this molecule (see Figures C.3, C.8, and C.9).

In general, the concentration curves of the `5 Å simulation`, besides showing, for several species, no changes in metabolite amounts, have two peculiar trends; precisely, they can

1. increase/decrease until they reach a plateau (as in the case of `GLC`, in Figure C.2);

2. get to a value after which they roughly oscillate for the remainder of the simulation (a behaviour characterising `F6P`, in Figure C.1, and `G6P`, in Figure C.2).

The two situations are linked to the inability of specific enzymes to saturate when allowed to perceive their cognate metabolites through a small perception sphere. In the first case, they bind their substrate until all the enzymes of the same kind are partly saturated (i.e., forming the dual-complexes defined in Section 5.3.1); at this point, they are not able to complete the catalysis of the reaction, and the substrate concentration stabilises at the reached value. The second case is due to a similar reason but relates to metabolites also involved in reversible reactions that do not require additional substrate (e.g., an energy donor) to be catalysed. Instead of reaching a plateau, when the partly-saturated enzyme cannot consume them, they start to "move back and forth" in the interconversion process of the reversible reaction, showing the oscillations mentioned above. The most relevant examples of these phenomena are the reactions catalysed by HXK (and GLK) and the one carried out by PFK:

$$\begin{aligned}
&\texttt{hexokinase:} && \texttt{GLC} + \texttt{ATP} \xrightarrow{\texttt{HXK}} \texttt{G6P} + \texttt{ADP} \\
&\texttt{glucokinase:} && \texttt{GLC} + \texttt{ATP} \xrightarrow{\texttt{GLK}} \texttt{G6P} + \texttt{ADP} \\
&\texttt{phosphofructokinase:} && \texttt{ATP} + \texttt{F6P} \xrightarrow{\texttt{PFK}} \texttt{ADP} + \texttt{F16bP}
\end{aligned}$$

Since, in the 5 Å simulation, PFK cannot bind ATP (Figure C.7), all the molecules of this enzyme remain partly saturated, and the F6P still present in the environment begins to be interconverted to and from G6P by the phosphoglucose isomerase (PGI1–see Table 6.2). Since HXKs and GLKs also reach a point in which they are all partly saturated (Figure C.5) and cannot produce G6P from glucose, the concentration of G6P starts to oscillate. The phosphoglucomutase (PGM), which interconverts G6P and G1P, equally participates in this process (the sequence of glycolysis reactions schematised in Figure 6.1 on page 108 may help the reader to understand these behaviours). In both plateaus and oscillations, the glycolytic process is blocked by the reactions catalysed by enzymes unable to saturate.

Conversely, the concentration changes of the 10 Å simulation have trends similar to those of the 300 Å simulation, although they show significantly lower rates. Among the others, it is important to notice this property in glucose (Figure C.2), pyruvate (Figure C.3), and in the products of the branches considered in our models, that is, glycerol (Figure C.2) and trehalose (Figure C.4).

We also provide the plots generated by a numerical time-course simulation, carried out through Copasi [55], of the Smallbone2013 kinetic model [107]. These plots are just for comparison; in Chapter 6, we propose various considerations over the limitations of this modelling and simulation approach. The main discrepancy with the agent-based simulations is the inability of the kinetic model to grasp the fluctuations of the species concentrations, producing more homogeneous curves. Moreover, a system of differential equations is less flexible than an agent-based model and removing enzyme regulation would have compromised its consistency, making the numerical simulation impossible. This property resulted in concentration changes closer to those observable at steady state, a condition unlikely to be reached by our agent-based simulations (see Section 6.2.1). If we exclude these differences, the numerical simulation, in some cases,

shows concentration trends loosely closer to those observable in the 300 Å simulation and 10 Å simulation (see the GLC curve in Figure C.2), while, in others, to the variations produced by the 5 Å simulation. In several cases, they are completely different from the corresponding concentration changes generated by the agent-based simulations.

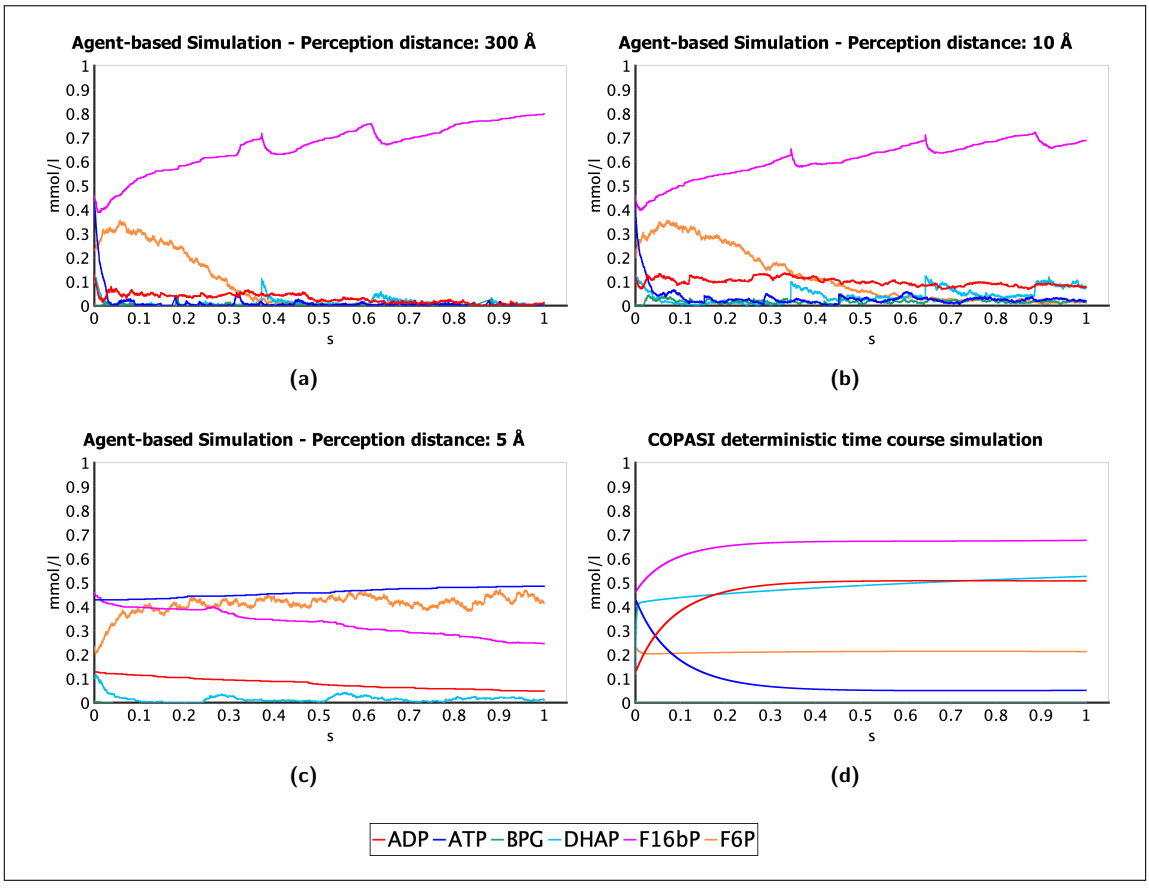


Figure C.1 – Concentration changes over time of adenosine diphosphate (ADP), adenosine triphosphate (ATP), 1,3-bisphosphoglycerate (BPG), dihydroxyacetone phosphate (DHAP), fructose 1,6-bisphosphate (F16bP), and fructose 6-phosphate (F6P).

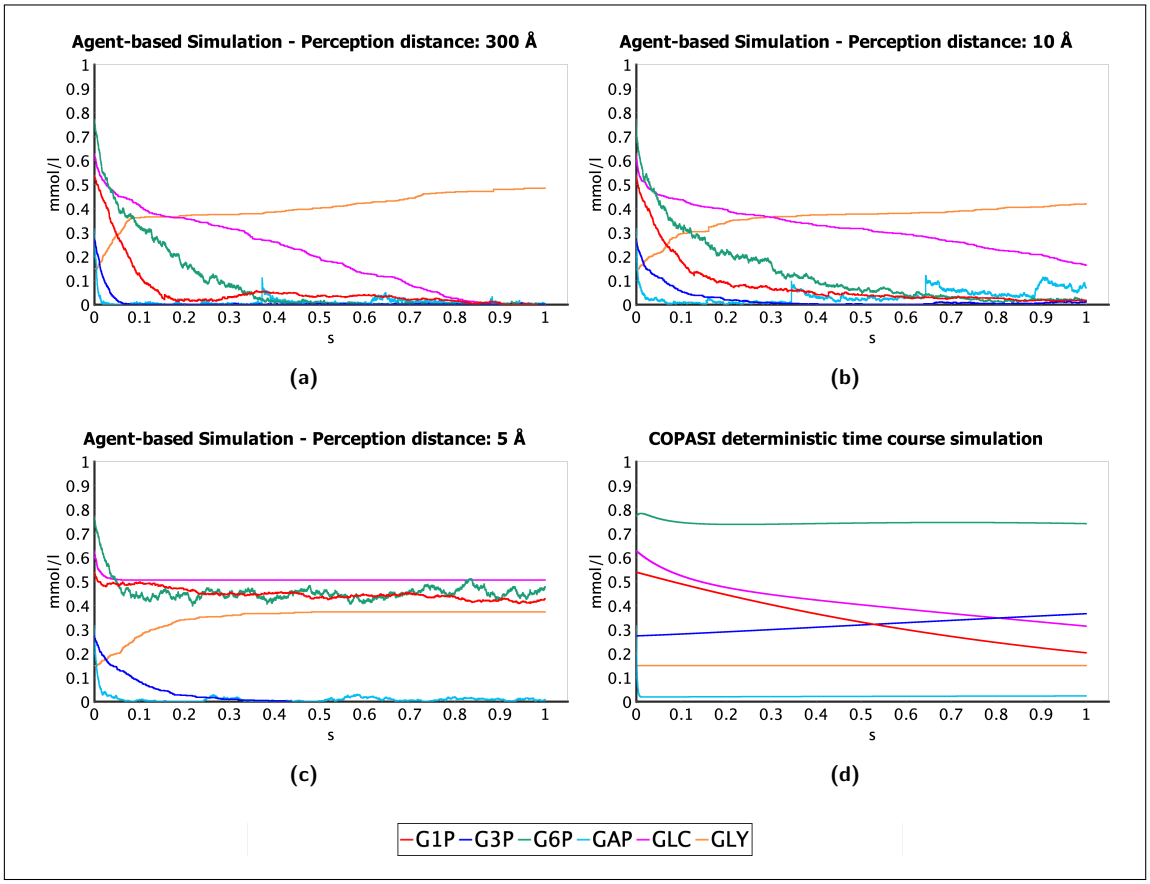


Figure C.2 – Concentration changes over time of `glucose 1-phosphate` (`G1P`), `glycerol 3-phosphate` (`G3P`), `glucose 6-phosphate` (`G6P`), `glyceraldehyde 3-phosphate` (`GAP`), `glucose` (`GLC`), and `glycerol` (`GLY`).

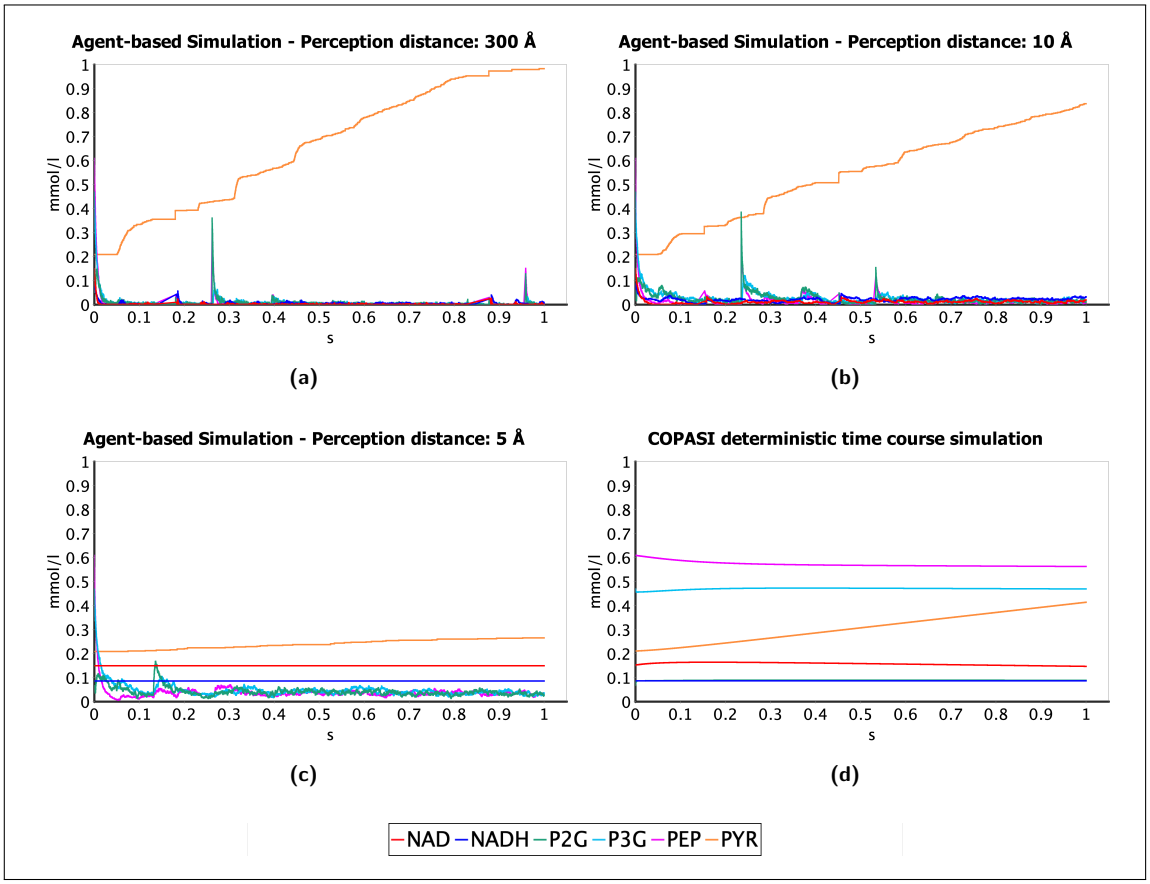


Figure C.3 – Concentration changes over time of nicotinamide adenine dinucleotide (NAD), nicotinamide adenine dinucleotide plus hydrogen (NADH), 2-phosphoglycerate (P2G), 3-phosphoglycerate (P3G), phosphoenolpyruvate (PEP), and pyruvate (PYR).

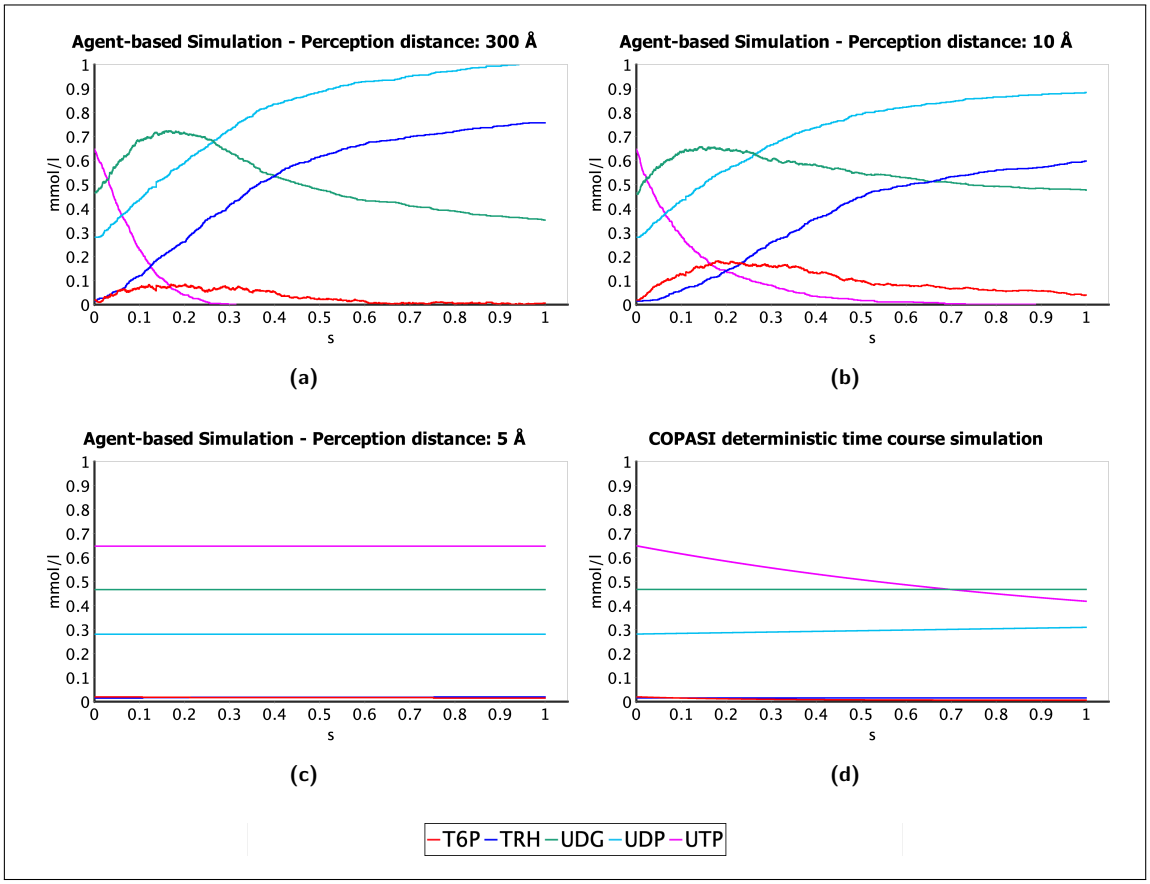


Figure C.4 – Concentration changes over time of `trehalose 6-phosphate` (`T6P`), `trehalose` (`TRH`), `uridine diphosphate glucose` (`UDG`), `uridine diphosphate` (`UDP`), and `uridine triphosphate` (`UTP`).

C.1.2 Comparison of relevant complexes formation

In this subsection, we provide a comparison of the plots of metabolites and complexes involved, during 5, 10 and 300 Å simulations, in the following reactions:

- phosphorylation of glucose (GLC) to glucose-6-phosphate (G6P), performed by hexokinase (HXK) and glucokinase (GLK);
- phosphorylation of fructose 6-phosphate (F6P) to fructose 1,6-bisphosphate (F16bP), carried out by phosphofructokinase (PFK);
- conversion of dihydroxyacetone phosphate (DHAP) to glycerol 3-phosphate (G3P), catalysed by glycerol-3-phosphate dehydrogenase (GDP).

The reason for this choice is to show how the agent-based approach can highlight the limitations of a system in which molecular interactions are driven only by random encounters and chemical affinity (that is, the 5 Å simulation). This is possible thanks to the capability of the agent-based simulations to reproduce local molecular interactions and thus the formation of partly saturated enzymes. A similar analysis cannot be carried out over the numerical time-course simulations, because they consider each reaction as a mathematical function from reactants to products, not explicitly taking into account the effects of the local interactions. For this reason, all the plots generated through Copasi cannot show the concentration changes of the complexes (whether they are enzymes partly or fully saturated).

In the proposed reactions, we can observe that, differently from the 300 and 10 Å simulations, the 5 Å simulation does not allow the binding of the enzymes with the needed energy donor (ATP for the reactions catalysed by HXK, GLK and PFK, and NADH for the reaction performed by GDP). This property produces the peculiar concentration changes discussed in the previous subsection and in Chapter 6. To make such a phenomenon more evident, each plot is provided with a dedicated legend; in this way, it is possible to notice at a glance that fully saturated enzymes are not present in the plots of the 5 Å simulation.

We show the concentration changes of reactants, products, and complexes for each reaction catalysed by the isoenzymes considered in our model of glycolysis (see Table 6.2 on page 109 for the related chemical equations).

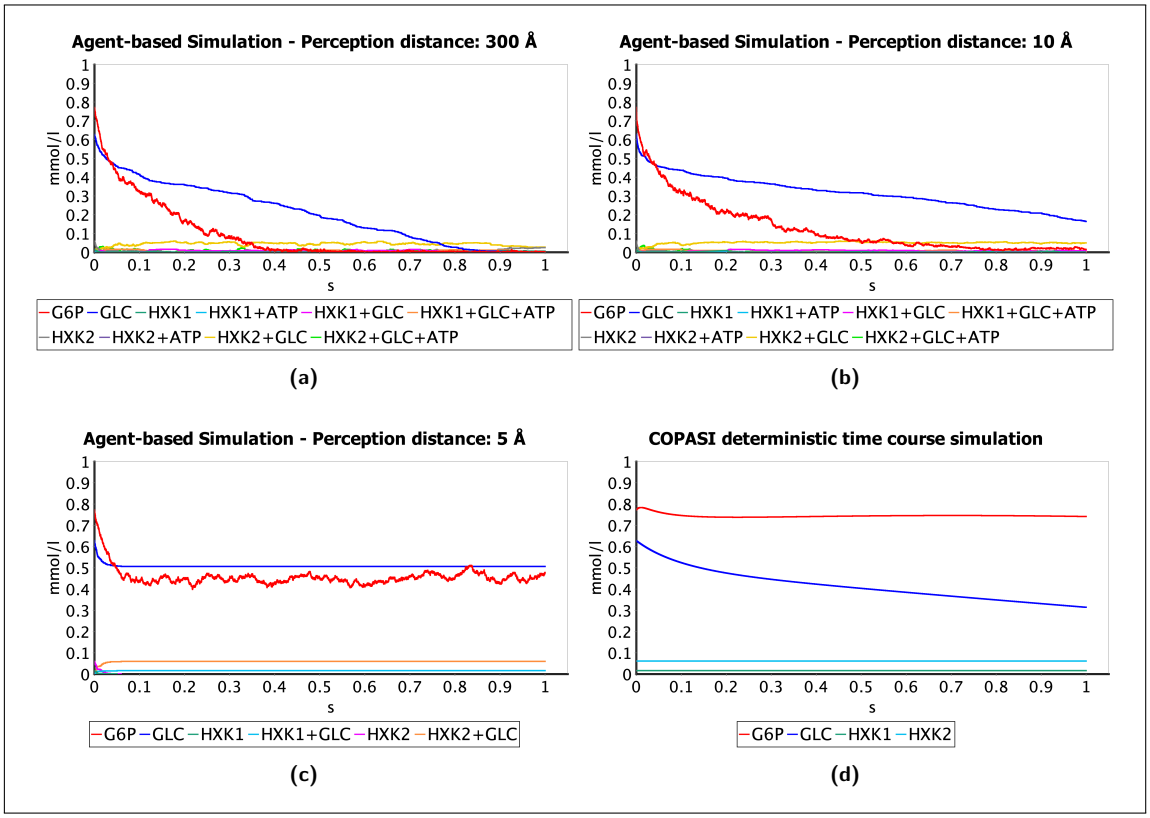


Figure C.5 – Phosphorylation of `glucose` (`GLC`) to `glucose-6-phosphate` (`G6P`), performed by `hexokinase` (`HXK1` and `HXK2` isoenzymes). The reaction requires the energy generated by the hydrolysis of `ATP` to `ADP`.

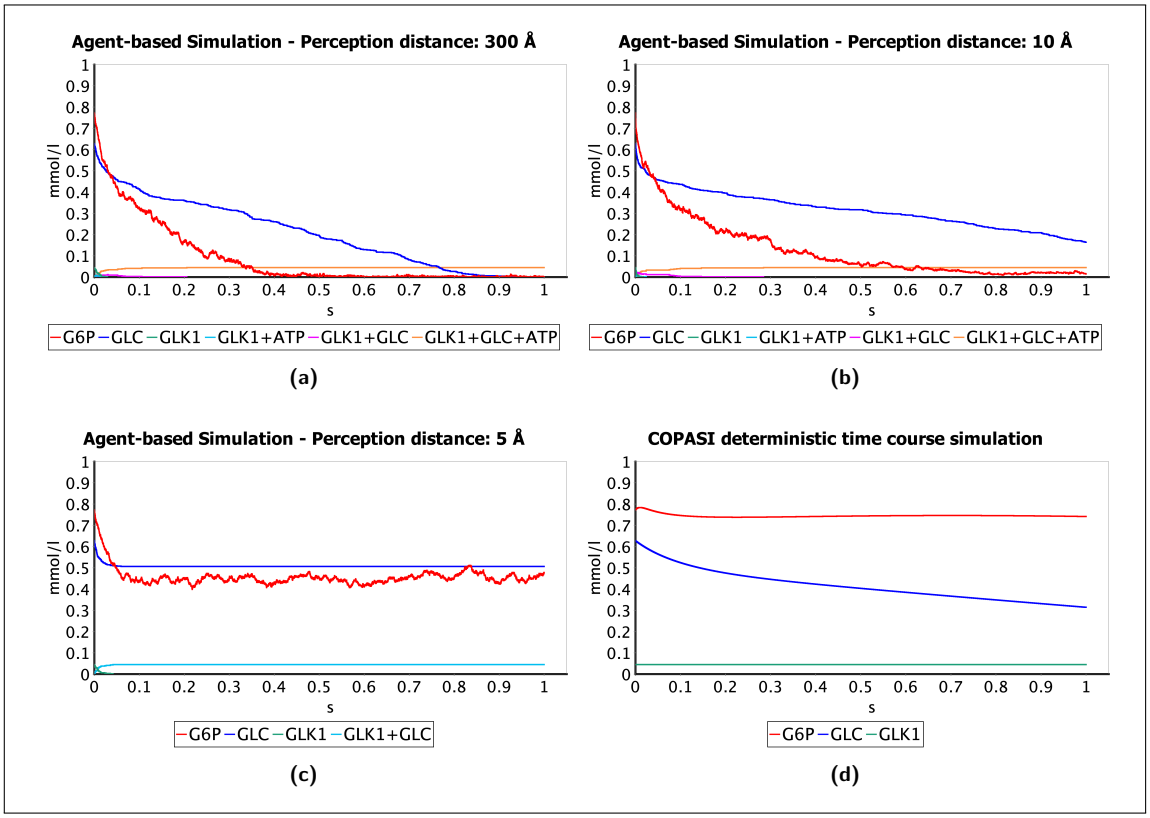


Figure C.6 – Phosphorylation of `glucose` (`GLC`) to `glucose-6-phosphate` (`G6P`), performed by `glucokinase` (`GLK1`). The reaction requires the energy generated by the hydrolysis of `ATP` to `ADP`.

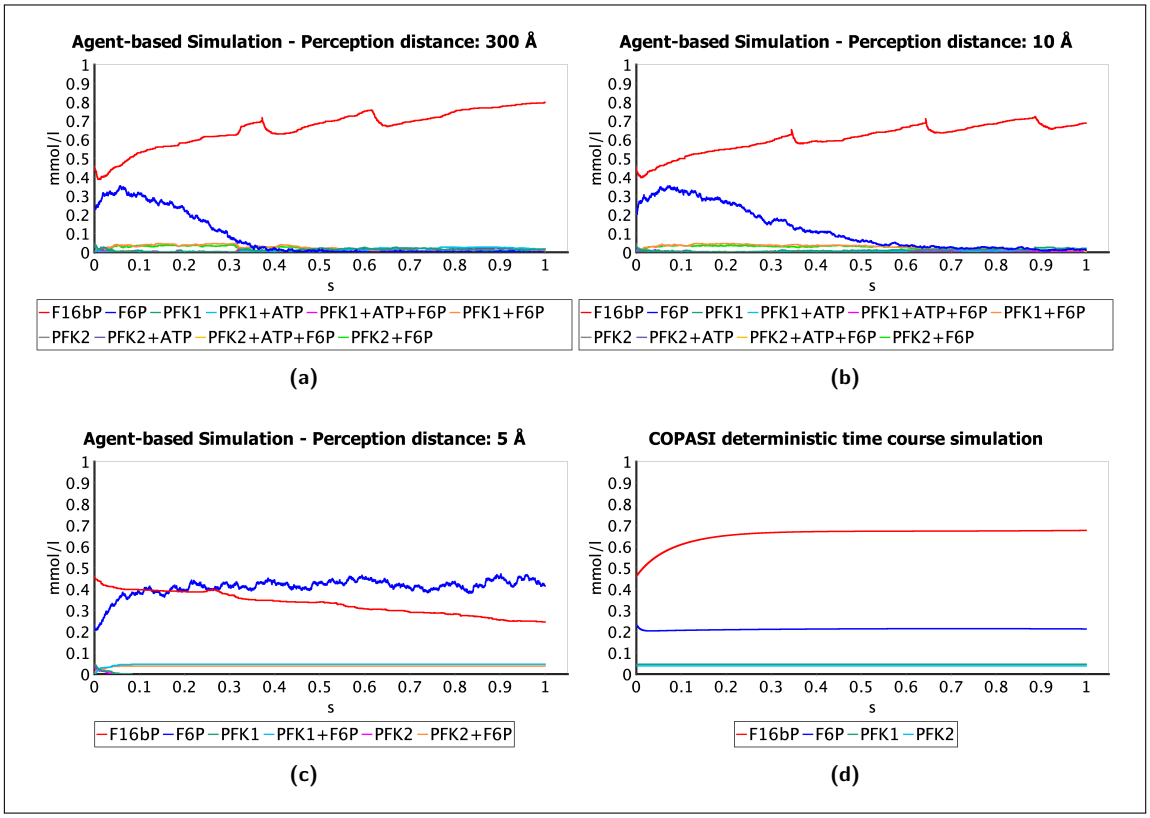


Figure C.7 – Phosphorylation of `fructose 6-hosphate` (F6P) to `fructose 1,6-bisphosphate` (F16bP), carried out by `phosphofructokinase` (PFK1 and PFK2 isoenzymes). The reaction requires the energy generated by the hydrolysis of `ATP` to `ADP`.

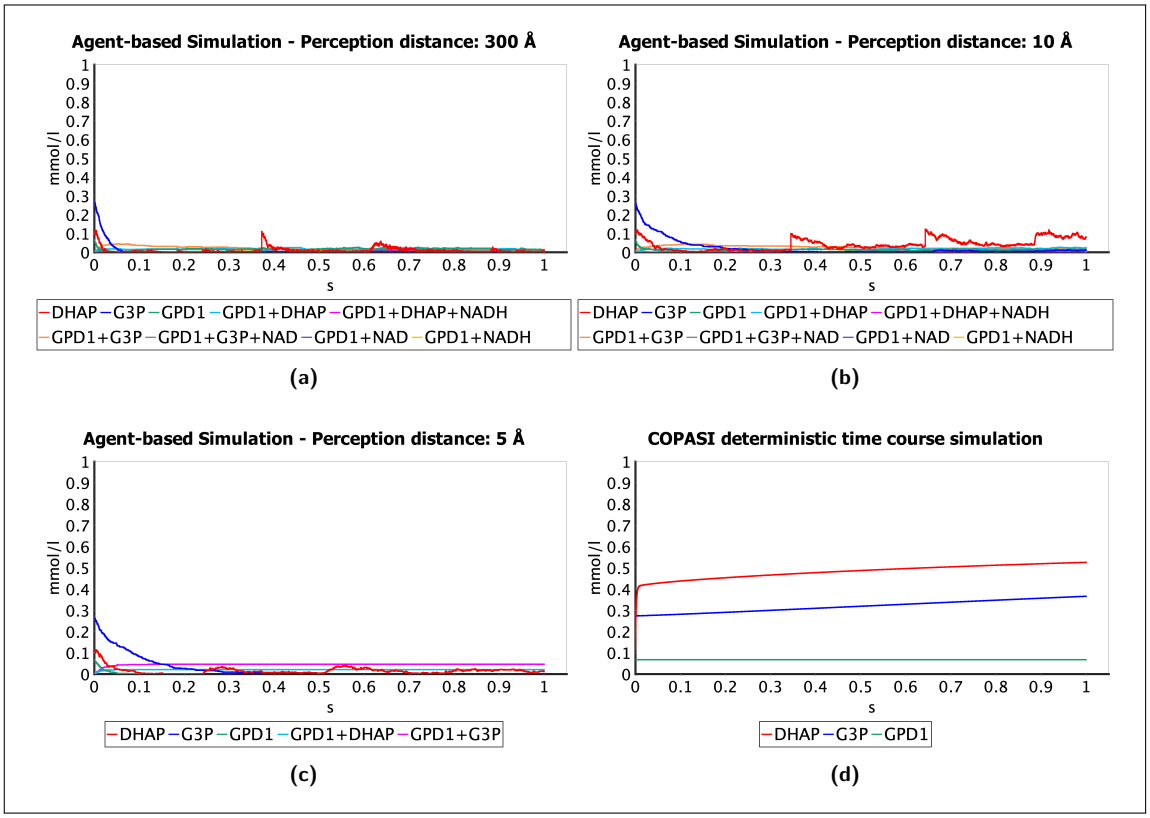


Figure C.8 – Conversion of `dihydroxyacetone phosphate` (`DHAP`) to `glycerol 3-phosphate` (`G3P`), catalysed by `glycerol-3-phosphate dehydrogenase` (`GPD1` isoenzyme). To be performed, the reaction must be coupled with the conversion of `NADH` to `NAD`.

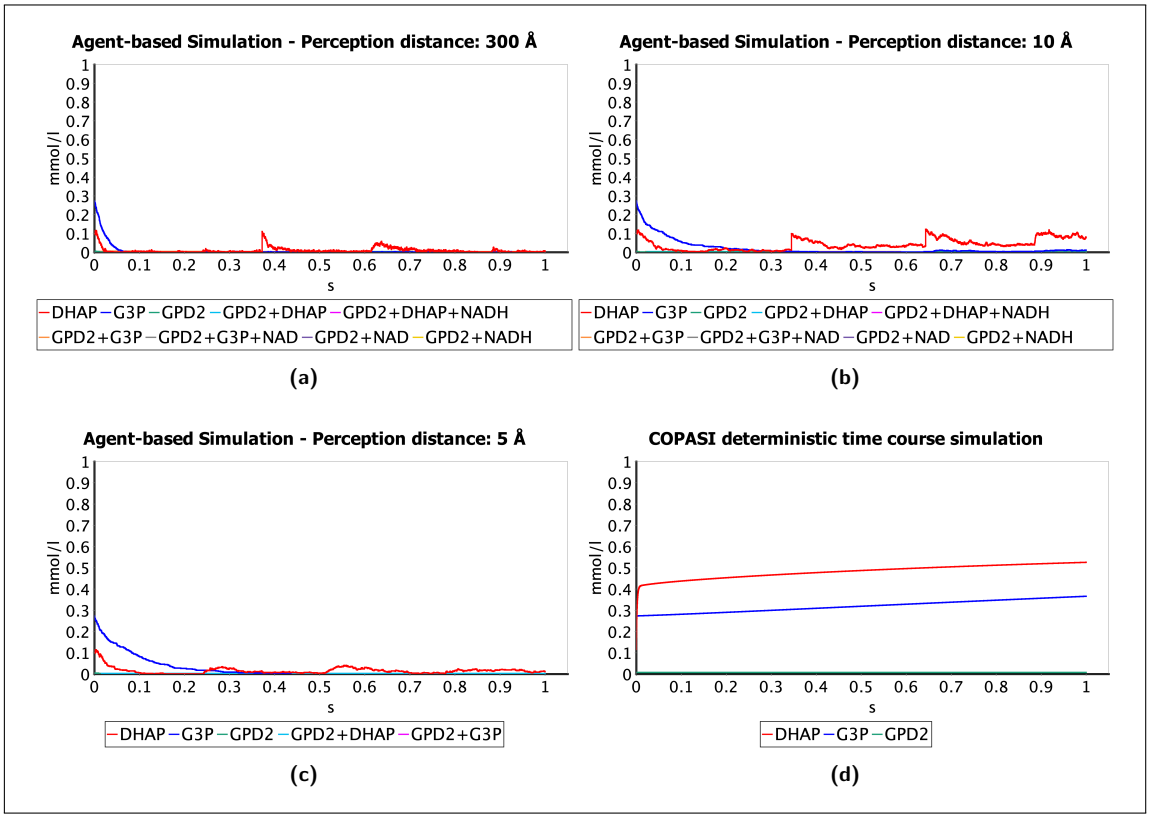


Figure C.9 – Conversion of `dihydroxyacetone phosphate` (`DHAP`) to `glycerol 3-phosphate` (`G3P`), catalysed by `glycerol-3-phosphate dehydrogenase` (`GPD2` isoenzyme). To be performed, the reaction must be coupled with the conversion of `NADH` to `NAD`.

References

[1] Luca Aceto. *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511814105.

[2] Bruce Alberts. *Molecular Biology of the Cell*. Sixth edition. New York, NY: Garland Science, Taylor and Francis Group, 2015. ISBN: 978-0-8153-4432-2.

[3] Jesper R. Andersen et al. "CAAL: Concurrency Workbench, Aalborg Edition". In: *Theoretical Aspects of Computing - ICTAC 2015*. Springer International Publishing, 2015, pp. 573–582. DOI: 10.1007/978-3-319-25150-9_33.

[4] Philip W. Anderson. "More Is Different". In: *Science* 177.4047 (1972), pp. 393–396. DOI: 10.1126/science.177.4047.393.

[5] Mauro Angeletti et al. "Spatial Behavioral Modeling and Simulation of Metabolic Pathways with Orion". In: *IV Bioinformatics ITalian Society Meeting (BITS 2007)*. Napoli, Italy, Apr. 2006, p. 70. URL: http://hdl.handle.net/11581/407946.

[6] Aristote. *Aristote : Oeuvres complètes et annexes (annotées, illustrées)*. Ed. by Jules Barthélemy-Saint-Hilaire. 1st edition. Arvensa Editions, 2019. ISBN: 9791027305896.

[7] Aristotle. *Aristotle's Metaphysics: a revised text with introduction and commentary*. Ed. by W. D. Ross. Oxford : Clarendon Press, 1924.

[8] Arianna Baldoncini. "Orion: A Spatial Multi Agent System Framework for Computational Cellular Dynamics of Metabolic Pathways". MSc Thesis. University of Camerino, 2004.

[9] Ezio Bartocci et al. "An Agent-Based Multilayer Architecture for Bioinformatics Grids". In: *IEEE Trans. NanoBioscience* 6.2 (June 2007), pp. 142–148. DOI: 10.1109/TNB.2007.897492.

[10] Ezio Bartocci et al. "Detecting Synchronisation of Biological Oscillators by Model Checking". English. In: *Theor. Comput. Sci.* 411.20 (Apr. 2010), pp. 1999–2018. DOI: 10.1016/j.tcs.2009.12.019.

[11] Matteo Belenchia et al. "Agent-based Learning Model for the Obesity Paradox in RCC". In: *Front. Bioeng. Biotechnol., section Nanobiotechnology* (2020). Submitted.

[12] Jeremy M. Berg, John L Tymoczko, and Lubert Stryer. “The Glycolytic Pathway Is Tightly Controlled”. In: *Biochemistry*. 5th edition. New York: W H Freeman, 2002. URL: https://www.ncbi.nlm.nih.gov/books/NBK22395/.

[13] Andrea Bernini et al. “Process calculi for biological processes”. In: *Nat. Comput.* 17 (2018). DOI: 10.1007/s11047-018-9673-2.

[14] J. Gordon Betts et al. *Anatomy and Physiology*. OpenStax, Apr. 2013. URL: https://openstax.org/books/anatomy-and-physiology/pages/1-introduction.

[15] Jeffrey S. Borer. “Angiotensin-Converting Enzyme Inhibition: A Landmark Advance in Treatment for Cardiovascular Diseases”. In: *Eur. Heart J. Suppl.* 9.suppl$_{\mathrm{E}}$ (Sept. 2007), E2–E9. DOI: 10.1093/eurheartj/sum037.

[16] Frances Brazier, Catholijn Jonker, and Jan Treur. “Compositional Design and Reuse of a Generic Agent Model”. en. In: *Appl. Artificial Intelligence* 14.5 (June 2000), pp. 491–538. DOI: 10.1080/088395100403397.

[17] David Brett et al. “Alternative Splicing and Genome Complexity”. In: *Nat Genet* 30.1 (Jan. 2002), pp. 29–30. DOI: 10.1038/ng803.

[18] George Edward Briggs and John Burdon Sanderson Haldane. “A Note on the Kinetics of Enzyme Action”. In: *Biochem. J.* 19.2 (Jan. 1925), pp. 338–339. DOI: 10.1042/bj0190338.

[19] H. Franklin Bunn. “Pathogenesis and Treatment of Sickle Cell Disease”. In: *N Engl J Med* 337.11 (Sept. 1997). Ed. by Franklin H. Epstein, pp. 762–769. DOI: bk6cj6.

[20] Federico Buti et al. “BioShape: A Spatial Shape-Based Scale-Independent Simulation Environment for Biological Systems”. In: *Procedia Comput. Sci.* 1.1 (May 2010), pp. 827–835. DOI: 10.1016/j.procs.2010.04.090.

[21] Birce Buturak et al. “Designing of Multi-Targeted Molecules Using Combination of Molecular Screening and in Silico Drug Cardiotoxicity Prediction Approaches”. English. In: *J. Mol. Graph. Model.* 50 (May 2014), pp. 16–34. DOI: 10.1016/j.jmgm.2014.02.007.

[22] Nicola Cannata, Flavio Corradini, and Emanuela Merelli. “Multiagent modelling and simulation of carbohydrate oxidation in cell”. In: *Int J Model. Identif. Control* 3 (2008). DOI: 10.1504/IJMIC.2008.018191.

[23] Nicola Cannata et al. “Agent-Based Models of Cellular Systems”. In: *Computational Toxicology: Volume II*. Ed. by Brad Reisfeld and Arthur N. Mayeno. Totowa, NJ: Humana Press, 2013, pp. 399–426. ISBN: 978-1-62703-059-5. DOI: 10.1007/978-1-62703-059-5_18.

[24] Luca Cardelli. “Brane Calculi”. In: *Computational Methods in Systems Biology: International Conference CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 257–278. ISBN: 978-3-540-25974-9. DOI: 10.1007/978-3-540-25974-9_24.

[25] Gunnar Carlsson et al. “Persistence Barcodes for Shapes”. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. ACM. 2004, pp. 124–135. DOI: 10.1145/1057432.1057449.

[26] Andrey G. Cherstvy, Anatoly B. Kolomeisky, and Alexei A. Kornyshev. “Protein–DNA interactions: reaching and recognizing the targets”. In: *J. Phys. Chem. B* 112.15 (Apr. 2008), pp. 4741–4750. DOI: 10.1021/jp076432e.

[27] Vincent Danos and Cosimo Laneve. “Formal Molecular Biology”. In: *Theor. Comput. Sci.* 325.1 (2004). Computational Systems Biology, pp. 69 –110. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2004.03.065.

[28] Mehdi Dastani, Farhad Arbab, and Frank de Boer. “Coordination and Composition in Multi-Agent Systems”. en. In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems - AAMAS ’05*. The Netherlands: ACM Press, 2005, p. 439. ISBN: 978-1-59593-093-4. DOI: 10.1145/1082473.1082540.

[29] Paul C.W. Davies, Lloyd Demetrius, and Jack A. Tuszynski. “Cancer as a Dynamical Phase Transition”. In: *Theor. Biol. Med. Model.* 8.1 (2011), p. 30. DOI: 10.1186/1742-4682-8-30.

[30] Ildefonso M. De la Fuente and Jesus M. Cortes. “Quantitative Analysis of the Effective Functional Structure in Yeast Glycolysis”. en. In: *PLoS One* 7.2 (Feb. 2012). Ed. by Christos A. Ouzounis, e30162. DOI: 10.1371/journal.pone.0030162.

[31] Jennifer Doudna and Jon Lorsch. “Ribozyme catalysis: Not different, just worse”. In: *Nat. Struct. Mol. Biol.* 12 (2005), pp. 395–402. DOI: 10.1038/nsmb932.

[32] D. Allan Drummond and Claus O. Wilke. “Mistranslation-Induced Protein Misfolding as a Dominant Constraint on Coding-Sequence Evolution”. In: *Cell* 134.2 (July 2008), pp. 341–352. DOI: 10.1016/j.cell.2008.05.042.

[33] Herbert Edelsbrunner and John Harer. “Persistent Homology-a Survey”. In: *Surveys on Discrete and Computational Geometry: Twenty Years Later*. Vol. 453. Providence, RI: American Mathematical Society, 2008, pp. 257–282. ISBN: 978-0-8218-8132-3. DOI: 10.1090/conm/453/08802.

[34] Harold P. Erickson. “Size and Shape of Protein Molecules at the Nanometer Level Determined by Sedimentation, Gel Filtration, and Electron Microscopy”. In: *Biol. Proced. Online* 11.1 (Dec. 2009), pp. 32–51. DOI: 10.1007/s12575-009-9008-x.

[35] Adrian Ferré-D’Amaré. “The glmSribozyme: Use of a small molecule coenzyme by a gene-regulatory RNA”. In: *Q. Rev. Biophys.* 43 (2010), pp. 423–47. DOI: 10.1017/S0033583510000144.

[36] Bernd M. Fischer, Markus Walther, and Peter Uhd Jepsen. “Far-infrared vibrational modes of DNA components studied by terahertz time-domain spectroscopy”. In: *Phys. Med. Biol.* 47.21 (Nov. 2002), pp. 3807–3814. DOI: 10.1088/0031-9155/47/21/319.

[37] Walter Fontana. “Systems Biology, Models, and Concurrency”. In: *SIGPLAN Not.* 43.1 (Jan. 2008), pp. 1–2. ISSN: 0362-1340. DOI: 10.1145/1328897.1328439.

[38] Claudio Forcato. “Orion: A Spatial Multiagent System Framework for Cellular Simulation. Implementation of Molecular Movement at the Mesoscale”. MSc Thesis. University of Camerino, 2005.

[39] Gianluigi Forloni et al. “Protein misfolding in Alzheimer’s and Parkinson’s disease: genetics and molecular mechanisms”. In: *Neurobiol. Aging* 23.5 (2002). Brain Aging: Identifying the Brakes and Accelerators, pp. 957 –976. ISSN: 0197-4580. DOI: 10.1016/S0197-4580(02)00076-3.

[40] Pamela A. Frischmeyer et al. “An mRNA Surveillance Mechanism That Eliminates Transcripts Lacking Termination Codons”. In: *Science* 295.5563 (Mar. 2002), pp. 2258–2261. DOI: 10.1126/science.1067338.

[41] Akiko Fukushima et al. “Development of a chimeric DNA-RNA hammerhead ribozyme targeting SARS virus”. In: *Intervirology* 52 (2009), pp. 92–9. DOI: 10.1159/000215946.

[42] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Los Altos, Calif: Morgan Kaufmann, 1987. ISBN: 978-0-934613-31-6.

[43] Marian Gidea and Yuri Katz. “Topological Data Analysis of Financial Time Series: Landscapes of Crashes”. In: *Phys. Stat. Mech. Its Appl.* 491 (2018), pp. 820–834. DOI: 10.1016/j.physa.2017.09.028.

[44] Walter Gilbert. “Origin of life: The RNA world”. In: *Nature* 319.6055 (1986), p. 618. DOI: 10.1038/319618a0.

[45] M. Gori et al. “Investigation of Brownian diffusion and long-distance electrodynamic interactions of biomolecules”. In: *2015 International Conference on Noise and Fluctuations (ICNF)*. 2015, pp. 1–4. DOI: 10.1109/ICNF.2015.7288566.

[46] Niels Gregersen et al. “Protein misfolding and human disease”. In: *Annu. Rev. Genomics Hum. Genet.* 7 (2006), pp. 103–124. DOI: 10.1146/annurev.genom.7.080505.115737.

[47] Darren Griffith, James P. Parker, and Celine J. Marmion. “Enzyme Inhibition as a Key Target for the Development of Novel Metal-Based Anti-Cancer Therapeutics”. In: *Anticancer Agents Med. Chem.* 10.5 (2010), pp. 354–370. DOI: 10.2174/1871520611009050354.

[48] Pedro M. R. Guimarães and John Londesborough. “The Adenylate Energy Charge and Specific Fermentation Rate of Brewer’s Yeasts Fermenting High- and Very High-Gravity Worts”. en. In: *Yeast* 25.1 (Jan. 2008), pp. 47–58. DOI: 10.1002/yea.1556.

[49] Antarip Halder et al. “How Does Mg2+ Modulate the RNA Folding Mechanism: A Case Study of the G: CW: W Trans Basepair”. In: *Biophys. J.* 113.2 (2017), pp. 277–289. DOI: 10.1016/j.bpj.2017.04.029.

[50] Yehouda Harpaz, Mark Gerstein, and Cyrus Chothia. “Volume Changes on Protein Folding”. en. In: *Structure* 2.7 (July 1994), pp. 641–649. DOI: 10.1016/S0969-2126(00)00065-4.

[51] Franz-Ulrich Hartl, Andreas Bracher, and Manajit Hayer-Hartl. “Molecular chaperones in protein folding and proteostasis”. In: *Nature* 475.7356 (2011), p. 324. DOI: 10.1038/nature10317.

[52] Janna Hastings et al. “ChEBI in 2016: Improved Services and an Expanding Collection of Metabolites”. In: *Nucleic Acids Res* 44.D1 (Jan. 2016), pp. D1214–D1219. DOI: 10.1093/nar/gkv1031.

[53] Robert M. Hazen. “The Emergence of Patterning in Lifes Origin and Evolution”. en. In: *Int. J. Dev. Biol.* 53.5-6 (2009), pp. 683–692. DOI: 10.1387/ijdb.092936rh.

[54] Matthew Hennessy and Robin Milner. “Algebraic laws for nondeterminism and concurrency”. In: *J. ACM* 32.1 (1985), pp. 137–161. DOI: 10.1145/2455.2460.

[55] Stefan Hoops et al. “COPASI–a COmplex PAthway SImulator”. en. In: *Bioinformatics* 22.24 (Dec. 2006), pp. 3067–3074. DOI: 10.1093/bioinformatics/btl485.

[56] Michael Hucka et al. “The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models”. en. In: *Bioinformatics* 19.4 (Mar. 2003), pp. 524–531. DOI: 10.1093/bioinformatics/btg015.

[57] Nicholas R Jennings. “An Agent-Based Approach for Building Complex Software Systems”. In: *Commun. ACM* 44.4 (2001), pp. 35–41. DOI: 10.1145/367211.367250.

[58] Randi Jimenez, Julio Polanco, and Andrej Lupták. “Chemistry and Biology of Self-Cleaving Ribozymes”. In: *Trends Biochem. Sci* 40 (2015). DOI: 10.1016/j.tibs.2015.09.001.

[59] Kenneth A. Johnson and Roger S. Goody. “The Original Michaelis Constant: Translation of the 1913 Michaelis–Menten Paper”. In: *Biochemistry* 50.39 (Oct. 2011), pp. 8264–8269. DOI: 10.1021/bi201284u.

[60] Melissa S. Jurica et al. “The Allosteric Regulation of Pyruvate Kinase by Fructose-1,6-Bisphosphate”. en. In: *Structure* 6.2 (Feb. 1998), pp. 195–210. DOI: 10.1016/S0969-2126(98)00021-5.

[61] Lee D. Kapp and Jon R. Lorsch. “The Molecular Mechanics of Eukaryotic Translation”. In: *Annu. Rev. Biochem.* 73.1 (June 2004), pp. 657–704. DOI: 10.1146/annurev.biochem.73.030403.080419.

[62] Robert M. Keller. “Formal verification of parallel programs”. In: *Commun. ACM* 19.7 (1976), pp. 371–384. DOI: 10.1145/360248.360251.

[63] Kim Larsen. “Proof systems for satisfiability in Hennessy-Milner Logic with recursion”. In: *Theor. Comput. Sci.* 72 (1990), pp. 265–288. DOI: 10.1016/0304-3975(90)90038-J.

[64] Michel Laurent, François Seydoux, and Philippe Dessen. “Allosteric Regulation of Yeast Phosphofructokinase. Correlation between Equilibrium Binding, Spectroscopic and Kinetic Data”. eng. In: *J. Biol. Chem.* 254.16 (Aug. 1979), pp. 7515–7520. DOI: 10.1016/s0021-9258(18)35974-x.

[65] Albert L. Lehninger, David L. Nelson, and Michael M. Cox. *Lehninger Principles of Biochemistry*. 4th ed. New York: W.H. Freeman, 2005. ISBN: 978-0-7167-4339-2.

[66] Maria V. Liberti and Jason W. Locasale. “The Warburg Effect: How Does It Benefit Cancer Cells?” In: *Trends Biochem. Sci* 41.3 (Mar. 2016), pp. 211–218. DOI: 10.1016/j.tibs.2015.12.001.

[67] Stefano Maestri and Emanuela Merelli. “Algebraic Characterisation of Non-coding RNA”. In: *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Ed. by Paolo Cazzaniga et al. Springer International Publishing, 2020, pp. 145–158. ISBN: 978-3-030-63061-4. DOI: 10.1007/978-3-030-63061-4_14.

[68] Adane Mamuye, Emanuela Merelli, and Luca Tesei. “A graph grammar for modelling RNA folding”. In: *Electron. Proc. Theor. Comput. Sci., EPTCS* 231 (2016), pp. 31–41. DOI: 10.4204/EPTCS.231.3.

[69] Adane Mamuye et al. “Persistent Homology Analysis of RNA”. In: *Mol. Based Math. Biol.* 4 (2016). DOI: 10.1515/mlbmb-2016-0002.

[70] Alain J. Marengo-Rowe. “Structure-Function Relations of Human Hemoglobins”. In: *Proc (Bayl Univ Med Cent)* 19.3 (July 2006), pp. 239–245. DOI: 10.1080/08998280.2006.11928171.

[71] Michele Mattioni. “Orion: A Multiagent System Framework for Cellular Simulation. Implementation of Metabolic Reaction at the Mesoscale”. MSc Thesis. University of Camerino, 2005.

[72] Andrew D. McLachlan. “A Variational Solution of the Time-Dependent Schrodinger Equation”. In: *Molecular Physics* 8.1 (Jan. 1964), pp. 39–44. DOI: 10.1080/00268976400100041.

[73] Emanuela Merelli, Marco Pettini, and Mario Rasetti. “Topology Driven Modeling: The IS Metaphor”. In: *Nat. Comput.* 14.3 (2015), pp. 421–430. DOI: 10.1007/s11047-014-9436-7.

[74] Emanuela Merelli and Anita Wasilewska. “Topological Interpretation of Interactive Computation”. In: *From Reactive Systems to Cyber-Physical Systems: Essays Dedicated to Scott A. Smolka on the Occasion of His 65th Birthday*. Springer International Publishing, 2019, pp. 205–224. ISBN: 978-3-030-31514-6. DOI: 10.1007/978-3-030-31514-6_12.

[75] Emanuela Merelli and Michal Young. “Validating MAS simulation models with mutation”. In: *Multiagent Grid Syst.* 3 (2007), pp. 225–243. DOI: 10.3233/MGS-2007-3206.

[76] Emanuela Merelli et al. “A Topological Approach for Multivariate Time Series Characterization: The Epileptic Brain”. In: *EAI Endorsed Trans. Self-Adapt. Syst.* 2.7 (2016), e5. DOI: 10.4108/eai.3-12-2015.2262525.

[77] Robin Milner. *Communication and concurrency*. Prentice Hall International, UK, 1989. ISBN: 978-0131149847.

[78] Ron Milo et al. “BioNumbers—the Database of Key Numbers in Molecular and Cell Biology”. In: *Nucleic Acids Res.* 38.suppl_1 (Jan. 2010), pp. D750–D753. DOI: 10.1093/nar/gkp889.

[79] Ellen M. Moody and Philip C. Bevilacqua. “Folding of a Stable DNA Motif Involves a Highly Cooperative Network of Interactions”. In: *J. Am. Chem. Soc.* 125.52 (Dec. 2003), pp. 16285–16293. DOI: 10.1021/ja038897y.

[80] Uma Nagaswamy et al. “Database of non-canonical base pairs found in known RNA structures”. In: *Nucleic Acids Res.* 28.1 (2000), pp. 375–376. DOI: 10.1093/nar/28.1.375.

[81] Ilaria Nardecchia et al. “Detection of Long-Range Electrostatic Interactions between Charged Molecules by Means of Fluorescence Correlation Spectroscopy”. en. In: *Phys. Rev. E* 96.2 (Aug. 2017), p. 022403. DOI: 10.1103/PhysRevE.96.022403.

[82] NCBI. *Homo sapiens gene HBB, encoding hemoglobin, beta*. Accessed on 20.11.2020. URL: https://ncbi.nlm.nih.gov/ieb/research/acembly/av.cgi?db=human&c=Gene&l=HBB.

[83] Kuniko Nielsen et al. “Sustained oscillations in glycolysis: an experimental and theoretical study of chaotic and complex periodic behavior and of quenching of simple oscillations”. In: *Biophys. Chem.* 72.1-2 (May 1998), pp. 49–62. DOI: 10.1016/S0301-4622(98)00122-7.

[84] Kerri-Ann Norton et al. “Multiscale Agent-Based and Hybrid Modeling of the Tumor Immune Microenvironment”. In: *Processes* 7.1 (Jan. 2019), p. 37. DOI: 10.3390/pr7010037.

[85] Paul C. Painter, Lue Mosher, and Carol Rhoads. “Low-Frequency Modes in the Raman Spectrum of DNA”. en. In: *Biopolymers* 20.1 (Jan. 1981), pp. 243–247. DOI: 10.1002/bip.1981.360200119.

[86] Asha Pandey et al. “Advancements in Nucleic Acid Based Therapeutics against Respiratory Viral Infections”. In: *J. Clin. Med.* 8 (2018), p. 6. DOI: 10.3390/jcm8010006.

[87] Nina Parker et al. *Microbiology*. OpenStax, Nov. 2016. URL: https://openstax.org/books/microbiology/pages/1-introduction.

[88] Giovanni Petri et al. “Topological Strata of Weighted Complex Networks”. In: *PLoS One* 8.6 (2013). DOI: 10.1371/journal.pone.0066506.

[89] Andrew Phillips, Luca Cardelli, and Giuseppe Castagna. “A Graphical Representation for Biological Processes in the Stochastic pi-Calculus”. In: *Transactions on Computational Systems Biology VII*. Ed. by Corrado Priami et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 123–152. ISBN: 978-3-540-48839-2. DOI: 10.1007/11905455_7.

[90] Marco Piangerelli, Luca Tesei, and Emanuela Merelli. “A Persistent Entropy Automaton for the Dow Jones Stock Market”. In: *Fundamentals of Software Engineering. FSEN 2019. Lect. Notes Comput. Sci.* Ed. by Hossein Hojjat and Mieke Massink. Vol. 11671. Cham: Springer International Publishing, 2019, pp. 37–42. ISBN: 978-3-030-31517-7. DOI: 10.1007/11419822.

[91] Marco Piangerelli et al. “Topological Classifier for Detecting the Emergence of Epileptic Seizures”. In: *BMC Res. Notes* 11 (2018), p. 392. DOI: 10.1186/s13104-018-3482-7.

[92] Kristine Potter, Nicole Cremona, and Jo Ann Wise. “Messenger RNA Processing in Eukaryotes”. In: *Encyclopedia of Biological Chemistry*. Elsevier, 2013, pp. 59–64. ISBN: 978-0-12-378631-9. DOI: 10.1016/B978-0-12-378630-2.00627-7.

[93] Matthew W. Powner, Béatrice Gerland, and John D Sutherland. “Synthesis of activated pyrimidine ribonucleotides in prebiotically plausible conditions”. In: *Nature* 459.7244 (2009), pp. 239–242. DOI: 10.1038/nature08013.

[94] Jordane Preto, Marco Pettini, and Jack A. Tuszynski. “Possible Role of Electrodynamic Interactions in Long-Distance Biomolecular Recognition”. In: *Phys. Rev. E* 91.5 (May 2015), p. 052710. DOI: 10.1103/PhysRevE.91.052710.

[95] Jordane Preto et al. “Experimental Assessment of the Contribution of Electrodynamic Interactions to Long-Distance Recruitment of Biomolecular Partners: Theoretical Basis”. In: *Phys. Rev. E* 85.4 (Apr. 2012), p. 041904. DOI: 10.1103/PhysRevE.85.041904.

[96] Michela Quadrini, Luca Tesei, and Emanuela Merelli. “An algebraic language for RNA pseudoknots comparison”. In: *BMC Bioinf.* 20 (2019). DOI: 10.1186/s12859-019-2689-5.

[97] Zahra Rahmani, Majid Mojarrad, and Meysam Moghbeli. “Long non-coding RNAs as the critical factors during tumor progressions among Iranian population: an overview”. In: *Cell Biosci.* 10 (2020). DOI: 10.1186/s13578-020-0373-0.

[98] Mario Rasetti and Emanuela Merelli. “Topological Field Theory of Data: Mining Data Beyond Complex Networks”. In: *Advances in Disordered Systems, Random Processes and Some Applications*. Ed. by Pierluigi Contucci and Cristian Giardinà. Cambridge University Press, 2016, pp. 1–42. DOI: 10.1017/9781316403877.002.

[99] Arun Renganathan and Emanuela Felley-Bosco. “Long Noncoding RNAs in Cancer and Therapeutic Potential”. In: *Adv. Exp. Med. Biol.* 1008 (2017), pp. 199–222. DOI: 10.1007/978-981-10-5203-3_7.

[100] Peter Richard. “The Rhythm of Yeast”. In: *FEMS Microbiol. Rev.* 27.4 (Oct. 2003), pp. 547–557. DOI: 10.1016/S0168-6445(03)00065-2.

[101] Frederic M. Richards. “AREAS, VOLUMES, PACKING, AND PROTEIN STRUCTURE”. In: *Annu. Rev. Biophys. Bioeng.* 6.1 (1977), pp. 151–176. DOI: 10.1146/annurev.bb.06.060177.001055.

[102] Edward A. Rietman et al. “An Integrated Multidisciplinary Model Describing Initiation of Cancer and the Warburg Hypothesis”. In: *Theor. Biol. Med. Model.* 10.1 (Dec. 2013), p. 39. DOI: 10.1186/1742-4682-10-39.

[103] Matteo Rucco et al. “Characterisation of the Idiotypic Immune Network Through Persistent Entropy”. In: *Proceedings of ECCS 2014*. Ed. by Stefano Battiston et al. Cham: Springer International Publishing, 2016, pp. 117–128. ISBN: 978-3-319-29226-7 978-3-319-29228-1. DOI: 10.1007/978-3-319-29228-1_11.

[104] Alejandro Sanchez et al. “Transcriptomic Signatures Related to the Obesity Paradox in Patients with Clear Cell Renal Cell Carcinoma: A Cohort Study”. In: *Lancet Oncol.* 21.2 (Feb. 2020), pp. 283–293. DOI: 10.1016/S1470-2045(19)30797-1.

[105] Matteo Santoni, Alessio Cortellini, and Sebastiano Buti. “Unlocking the Secret of the Obesity Paradox in Renal Tumours”. In: *Lancet Oncol.* 21.2 (Feb. 2020), pp. 194–196. DOI: 10.1016/S1470-2045(19)30783-1.

[106] Alexander Serganov and Dinshaw Patel. “Ribozymes, riboswitches and beyond: regulation of gene expression without proteins.” In: *Nat. Rev. Genet.* 8 (2007), pp. 776–790. DOI: 10.1038/nrg2172.

[107] Kieran Smallbone et al. “A Model of Yeast Glycolysis Based on a Consistent Kinetic Characterisation of All Its Enzymes”. In: *FEBS Lett.* A Century of Michaelis - Menten Kinetics 587.17 (Sept. 2013), pp. 2832–2841. DOI: 10.1016/j.febslet.2013.06.043.

[108] Kieran Smallbone et al. “A Model of Yeast Glycolysis Based on a Consistent Kinetic Characterisation of All Its Enzymes”. In: *FEBS Lett.* A Century of Michaelis - Menten Kinetics 587.17 (Sept. 2013), pp. 2832–2841. DOI: 10.1016/j.febslet.2013.06.043.

[109] Michael J. Stephen. “First-Order Dispersion Forces”. In: *J. Chem. Phys.* 40.3 (Feb. 1964), pp. 669–673. DOI: 10.1063/1.1725188.

[110] Jeffrey Strathern et al. “The Fidelity of Transcription”. In: *J. Biol. Chem.* 288.4 (Jan. 2013), pp. 2689–2699. DOI: 10.1074/jbc.M112.429506.

[111] XiaoBo Tang, Gerd Hobom, and Dong Luo. “Ribozyme mediated destruction of influenza A virus”. In: *J. Med. Virol.* 42 (1994), pp. 385–95. DOI: 10.1002/jmv.1890420411.

[112] Bas Teusink et al. “Can Yeast Glycolysis Be Understood in Terms of in Vitro Kinetics of the Constituent Enzymes? Testing Biochemistry: Do We Understand Yeast Glycolysis?” en. In: *Eur. J. Biochem.* 267.17 (Sept. 2000), pp. 5313–5329. DOI: 10.1046/j.1432-1327.2000.01527.x.

[113] Nguyen Quoc Thai et al. “Protocol for Fast Screening of Multi-Target Drug Candidates: Application to Alzheimer’s Disease”. In: *J. Mol. Graph. Model.* 77 (2017), pp. 121–129. DOI: 10.1016/j.jmgm.2017.08.002.

[114] The UniProt Consortium. “UniProt: A Worldwide Hub of Protein Knowledge”. In: *Nucleic Acids Res.* 47.D1 (Jan. 2019), pp. D506–D515. DOI: 10.1093/nar/gky1049.

[115] Danielle Thierry-Mieg and Jean Thierry-Mieg. “AceView: A Comprehensive cDNA-Supported Gene and Transcripts Annotation”. In: *Genome Biol* 7.Suppl 1 (2006), S12. DOI: 10.1186/gb-2006-7-s1-s12.

[116] Joost van den Brink et al. “Dynamics of Glycolytic Regulation during Adaptation of Saccharomyces Cerevisiae to Fermentative Metabolism†”. en. In: *AEM* 74.18 (Sept. 2008), pp. 5710–5723. DOI: 10.1128/AEM.01121-08.

[117] Bryan J. Venters and B. Franklin Pugh. “How Eukaryotic Genes Are Transcribed”. In: *Crit. Rev. Biochem. Mol. Biol.* 44.2-3 (June 2009), pp. 117–141. DOI: 10.1080/10409230902858785.

[118] Otto Warburg. “The Metabolism of Carcinoma Cells”. In: *The Journal of Cancer Research* 9.1 (Mar. 1925), pp. 148–163. DOI: 10.1158/jcr.1925.148.

[119] James D. Watson, ed. *Molecular Biology of the Gene*. Seventh edition. Boston: Pearson, 2014. ISBN: 978-0-321-76243-6.

[120] Hans V. Westerhoff et al. “From Silicon Cell to Silicon Human”. en. In: *BetaSys*. Ed. by Bernhelm Booß-Bavnbek et al. New York, NY: Springer New York, 2011, pp. 437–458. ISBN: 978-1-4419-6955-2 978-1-4419-6956-9. DOI: 10.1007/978-1-4419-6956-9_19.

[121] Jana Wolf et al. “Transduction of Intracellular and Intercellular Dynamics in Yeast Glycolytic Oscillations”. In: *Biophys. J.* 78.3 (Mar. 2000), pp. 1145–1153. DOI: 10.1016/S0006-3495(00)76672-0.

[122] World Wide Web Consortium (W3C). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Feb. 2013. URL: https://www.w3.org/TR/REC-xml/.

[123] Kelin Xia et al. “Persistent Homology Analysis of Osmolyte Molecular Aggregation and Their Hydrogen-Bonding Networks”. In: *Phys. Chem. Chem. Phys.* 21.37 (2019), pp. 21038–21048. DOI: 10.1039/C9CP03009C.

[124] A.A. Zamyatnin. “Protein Volume in Solution”. en. In: *Prog. Biophys. Mol. Biol.* 24 (Jan. 1972), pp. 107–123. DOI: 10.1016/0079-6107(72)90005-3.

[125] Jinwei Zhang, Matthew Lau, and Adrian Ferré-D’Amaré. “Ribozymes and Riboswitches: Modulation of RNA Function by Small Molecules”. In: *Biochemistry* 49 (2010), pp. 9123–31. DOI: 10.1021/bi1012645.

[126] Afra Zomorodian. “Topological Data Analysis”. In: *Advances in Applied and Computational Topology. Proceedings of Symposia in Applied Mathematics*. Vol. 70. 2007, pp. 1–39. ISBN: 978-0-8218-8997-8. DOI: 10.1090/psapm/070.