HAL
open science

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

Stefano Maestri

To cite this version:

Stefano Maestri. Modelling the effects of long-range forces in biological systems to better understand the global behaviour of molecular interactions. Physics [physics]. Aix-Marseille Université (AMU); School of Computer Science, University of Camerino, Italy; Centre de Physique Théorique - UMR 7332, 2020. English. NNT : . tel-03598318

HAL Id: tel-03598318

https://hal.science/tel-03598318v1

Submitted on 4 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

School of Advanced Studies

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:
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
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

School of Advanced Studies

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 des effets des forces à longue portée dans les systèmes biologiques afin de 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
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

Contents

1 Introduction **1**
1.1 Algebraic Modelling of RNA and Proteins . . . 2
1.2 Agent-based Modelling and Simulation . . . 3
1.3 Organisation of the Manuscript . . . 4
1.4 Modélisation algébrique de l'ARN et des protéines . . . 8
1.5 Modélisation et simulation basées sur des agents . . . 9
1.6 Organisation du manuscrit . . . 11

I Algebraic Models **13**

2 Background and Methods for the Part I **15**
2.1 Basic Introduction to Molecular Biology and Gene Expression . . . 15
2.1.1 RNA Translation . . . 18
2.1.2 Protein Structure and Folding . . . 20
2.1.3 Functional RNA . . . 24
2.1.4 RNA World . . . 26
2.1.5 Haemoglobin and Anaemias . . . 28
2.2 Algebraic Modelling of Biological Systems . . . 32
2.2.1 Calculus of Communicating Systems . . . 32
2.2.2 Labelled Transition Systems . . . 35
2.2.3 Hennessy-Milner Logic . . . 35
2.2.4 From Algebraic to Agent-based Models . . . 36

3 RNAs and Proteins equivalence **39**
3.1 Introduction . . . 39
3.2 Results . . . 40
3.2.1 Bisimilarity equivalence . . . 44
3.2.2 Higher abstraction level model . . . 46
3.3 Discussion . . . 48

3.4 Conclusions 49

4 Algebraic Study of Protein Misfolding **51**
4.1 Introduction 51
4.1.1 DNA replication and gene expression models 52
4.1.2 Formal description of HBB gene replication and expression 59
4.2 Results 61
4.3 Discussion 64
4.4 Conclusions 70

5 Algebraic Characterisation of Non-coding RNA **71**
5.1 Introduction 71
5.2 **Results** 72
5.2.1 Ligand Binding Function 73
5.2.2 Enzymatic Function 74
5.2.3 Model checking 78
5.3 **Conclusions** 80

II Agent-based Simulation of Metabolic pathways **83**

6 Background and Methods for the Part II **85**
6.1 Introduction to Yeasts' Glycolysis 85
6.2 Agent-based approach 89
6.2.1 Agent-based Simulator for Metabolic Pathways 89
6.2.2 From a Kinetic Model to a Multiagent Simulation 90
6.2.3 Choosing a Reference Kinetic Model 93
6.2.4 Defining the Input for the Simulation 93
6.2.5 Simulation Output and Visualisation 96

7 Testing in Silico the Bimolecular Interactions **103**
7.1 Introduction 103
7.2 Integrative Methods for this Chapter 104
7.2.1 Long-distance Electrodynamic Interactions 104
7.2.2 Modelling the Whole Glycolytic Pathway 104
7.2.3 Simulating a Large Number of Molecules 105
7.3 Results 108
7.4 Discussion 112
7.5 Conclusions 113

8 Interaction-as-perception in Metabolic Reactions **115**
8.1 Introduction 115

8.2 Integrative Methods for this Chapter . . . 116
8.2.1 Multi-agent Modelling and Simulation . . . 116
8.2.2 Simplicial Data Analysis . . . 116
8.2.3 Interaction-as-perception Paradigm . . . 117
8.3 Results . . . 121
8.4 Discussion . . . 125
8.5 Conclusions . . . 126

9 Conclusions 127

Appendices 131

A Supplementary Information to Chapter 2 133
A.1 Symbols and their transliteration . . . 133
A.2 Models Construction . . . 138
A.2.1 Base pairing . . . 138
A.2.2 Electrostatic interactions . . . 141
A.2.3 Hydrophobic interactions . . . 142
A.2.4 Folding step . . . 144
A.2.5 RNA folding and protein folding . . . 147
A.2.6 Model checking . . . 147
A.2.7 Higher abstraction level model . . . 148

B Supplementary Information to Chapter 4 157
B.1 Gene Expression Model . . . 157
B.1.1 Transcription process . . . 158
B.1.2 RNA Processing . . . 162
B.1.3 Translation process . . . 166
B.2 Formal description of HBB gene expression . . . 171
B.2.1 Replication . . . 172
B.2.2 Mismatch repair . . . 172
B.2.3 Transcription . . . 174
B.2.4 Processing . . . 175
B.2.5 Translation . . . 176

C Supplementary Information to Chapter 7 179
C.1 Plots of the Concentration Changes in the Agent-based Simulations . . . 179

Introduction

Our understanding of a biological process is often held back by the entanglement of interactions at its basis, since the relation between these local connections and the process as a whole appears blurred from a top-down perspective. To use the words of Aristotle in the "Metaphysics": "*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 beside the parts, there is a cause*" [5]. The actual effect of this cause on the "whole thing" remains hidden when investigated through a reductionist approach.

By taking a complex system apart recursively to its smallest component, we can certainly gain relevant knowledge about each accounted structure, but the behaviour of the system is understandable only in terms of its global properties [39, 60]. In this way, sophisticated biological functions arise as new entities, by eclipsing the simple local rules through which the basic components interact.

In this manuscript, we analyse the behaviour characterising biological macromolecules, from the steps that lead them to reach their three-dimensional conformations, to the way they interact with one another.

We exploit an algebraic modelling approach to provide a formal definition of the local interactions characterising the nucleotides in RNA molecules and the amino acids in the polypeptide sequence of proteins; identifying their collective properties in the expression of a fully functional macromolecule brings out congruences and dissimilarities, which can in some cases be associated with genetic pathologies. We also investigate the global behaviour of the long-distance electrodynamic interactions in metabolic pathways through a specifically designed agent-based paradigm.

The core idea of our work is to show how algebraic and agent-based approaches are highly suitable to uncover complex phenomena in biological processes and give a new light in the interpretation of biological systems and genetic diseases.

In the following sections, we provide a brief overview and contextualisation of the topics addressed in the rest of this manuscript. This is structured in two main parts; the first one focusses on the algebraic models of RNAs and proteins, while the second part provides a description of our agent-based studies on biomolecular interactions.

Although these two approaches can be connected to each other (as shown in Chapter 5), we separate them to allow the reader to clearly identify the work carried out mainly in the context of the University of Camerino (Part 1) and the one that is the result of the collaboration with the Aix-Marseilles University (Part 2).

1.1 Algebraic Modelling of RNA and Proteins

The relation between structures and functions is a relevant topic in biology, whose investigation received a significant contribution by different computational approaches, from process calculi to topological data analysis [9, 15, 51, 54, 68].

In particular, formal languages and graph grammars have been successfully applied in modelling the properties that correlate the functions expressible by RNA molecules and specific substructures involved in their folding - the process that allows a linear biopolymer to reach a three-dimensional conformation by forming hydrogen bonds between non-consecutive monomers [52, 73].

In this manuscript, we push forward this approach and prove that the complexity of RNA functions can be traced back to the inner potentiality of each nucleotide to interact with the others in the same sequence. This result is obtained by comparing the RNA folding with that performed by proteins, in order 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 *congruence level*. Reaching such a goal is possible thanks to the expressiveness of process algebras [1], through which we model both RNA and protein folding.

During the second half of the last century, investigating the reasons of existence of so similar molecules led to the formulation of the RNA World hypothesis: RNA might be a “fossil” of an RNA world, existed on Earth before modern cells appeared, in which RNA fulfilled the roles of both DNA and proteins. This theory is still highly debated and, indeed, beyond their similarities, proteins and RNAs show profound structural differences, which affect the way they perform their functions [30, 72].

As the first part of our work, we provide a formal description of the folding process of proteins compared to the one of RNAs. Our purpose is to identify, by highlighting their key properties, clues of the validity of the RNA World hypothesis. We focus our study on the interactions carried out by the elementary units that compose RNAs and proteins (on their respective linear

sequences), describing the whole folding process as the resulting behaviour of such interactions.

Subsequently, we concentrate on a class of pathologies that affects the folding processes to study how the differences between the structural components of proteins and RNAs cause a dissimilar response to an alteration of the correct folding pathway. This part of our study starts from the formal description of how such pathologies originate as an error of the genetic code (a mutation, in biological terms) and can propagate through each step of the gene expression, affecting both the RNA and the protein structures. We formally describe how the mutation of even a single gene (point mutation) can alter the final conformation of a protein while, at the same time, it is harmless for the structure of RNAs. We show how a well-known pathology affecting haemoglobin, the sickle-cell anaemia, can be considered as a global property of the interactions among amino acids as well as nucleotides.

We finally move another step forward, by hypothesising the functions that characterise the *congruence level* mentioned above and further exploring the applicability of process algebras in modelling the related biological processes. The resulting models will eventually form the basis of a multiagent simulation [43].

In an agent-based simulation, agents are discrete software elements whose interactions correspond to those performed by the components of the modelled system, quite faithfully to the actual behaviour of a biological process [56]. 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 multiagent systems.

1.2 Agent-based Modelling and Simulation of Biomolecular Interactions

In the second part of this manuscript, we propose a multiagent simulator developed to study the molecular interactions characterising metabolic pathways, and analyse its global properties starting from local interactions [13, 69]. We are able to simulate complete enzymatic reactions by modelling the molecules involved (enzymes, metabolites and complexes) as autonomous and interactive agents.

We explore the capabilities of the provided simulator to deal with the long-distance electrodynamic interactions that shape the behaviour of bimolecular systems, and analyse their effect on the evolution of a metabolic pathway, such as the yeast glycolysis. This investigation has been conducted in the context of our collaboration with the Centre de Physique Théorique of the Aix-Marseilles University.

In vitro studies showed that charge oscillating at high frequency (in the range of $10^{10} - 10^{11}$ Hz) does not suffer the Debye screening effect by the ions of the medium and a biological macromolecule behaves like an oscillating dipole; long-range forces may be activated between two resonant dipolar systems [31, 62].

Our aim is to provide an in silico validation to these experiments. Each molecule is represented by an agent able to perceive the environment and the cognate partners with which it can interact. A similar approach may also be adopted by defining a molecular dynamics model; however, this kind of method places the analysis at an atomistic level and the related simulations have a high computational load. The compositionality of the agent-based models, instead, permits to conduct the study at a macromolecular level, without loosing in accuracy and performing light-weighted simulations.

However, understanding and representing as a whole the agent dynamics characterising a metabolic reaction made by a large number of molecules still constitutes a big issue.

For this reason, we also 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. In this way we can derive the graph of perceptions at a given step; on those graphs, we apply the topological data analysis to capture the 3-body interactions through the interpretation of 2-simplices as observable structures, which are convex hulls of three points. We use the 2-simplex formation as a valid semantic to represent the global dynamics of the system.

Nevertheless, biological processes are complex systems whose global behaviour is not always possible to predict, due to the incompleteness of observed data.To incorporate this property in an agent-based model of a biological system, agents' interactions should have an aleatory nature or the simulation environment should be non-predictable (this implies that each run of the simulation is affected by statistical uncertainty). Further steps are needed to provide an effective specification of the environment, hopefully by referring to interactive computation modelling [55].

1.3 Organisation of the Manuscript

Each part of this manuscript is correlated with a first introductory chapter (Chapters 2 and 6), which describes the basic biological concepts needed to better comprehend our studies and the modelling approaches we adopted to reach the provided results.

The first part dedicated to our findings is composed by Chapters 3, 4 and 5. More specifically:

- in Chapter 3, we provide the algebraic models of RNA and protein folding and prove how

it is possible to formally define a level of abstraction in which such processes show a behavioural equivalence (congruence level). Its definition allowed us to hypothesise some of the reasons that lead the evolution of life to the formation of proteins and to take them on as the main catalysts in the biological processes.

- Chapter 4 analyses a class of pathologies that affects the folding processes to study how the differences between the structural components of proteins and RNAs cause a dissimilar response to an alteration of the correct folding pathway.
- In Chapter 5, we explore the expressiveness of process algebras in modelling the functions representing the behaviour of non-coding RNA molecules, as a result of the characterisation of the congruence level defined in Chapter 3. Basing on these results, we propose a methodology suitable to generate an algebraic specification of a multiagent simulation.

The second part of the body of this manuscript comprises Chapters 7 and 8:

- in Chapter 7, we describe a simulation environment dedicated to study molecular long-distance interactions in metabolic reactions; we propose a many-body approach, implemented as a multiagent system (MAS).
- Chapter 8 moves a step up form the previous chapter by using the MAS simulation to generate the dynamics of the biological complex system in order to visualise and understand the global behaviour of that system; this is possible thanks to the introduction of the interaction-as-perception paradigm.

Chapter 9 concludes the manuscript and provides our considerations on the results obtained, the limitations encountered during the studies and the possible improvements to take into account for future works.

Introduction

Notre compréhension d'un processus biologique est souvent freinée par l'enchevêtrement des interactions à sa base, car la relation entre ces connexions locales et le processus dans son ensemble semble floue d'un point de vue « top-down ». Pour reprendre les mots d'Aristote dans la " Métaphysique ": "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" [4]. L'effet réel de cette cause sur le « tout » reste caché lorsqu'il est étudié à travers une approche réductionniste.

En séparant récursivement un système complexe à son plus petit composant, nous pouvons certainement acquérir des connaissances pertinentes sur chaque structure pris en considération, mais le comportement du système n'est compréhensible qu'en termes de ses propriétés globales [39, 60]. De cette manière, des fonctions biologiques sophistiquées apparaissent comme de nouvelles entités, en éclipsant les règles locales simples à travers lesquelles les composants de base interagissent.

Dans ce manuscrit, nous analysons le comportement caractérisant les macromolécules biologiques, depuis les étapes qui les amènent à atteindre leurs conformations tridimensionnelles, à la manière dont elles interagissent les unes avec les autres.

Nous exploitons une approche de modélisation algébrique pour fournir une définition formelle des interactions locales caractérisant les nucléotides dans les molécules d'ARN et les acides aminés dans la séquence polypeptidique des protéines; l'identification de leurs propriétés collectives dans l'expression d'une macromolécule pleinement fonctionnelle fait ressortir des congruences et des dissemblances, qui peuvent dans certains cas être associées à des pathologies génétiques. Nous étudions également le comportement global des interactions électrodynamiques à longue distance dans les voies métaboliques à travers un paradigme basé sur des agents spécialement conçu.

L'idée centrale de notre travail est de montrer comment les approches algébriques et basées sur les agents sont parfaitement adaptées pour découvrir des phénomènes complexes dans les processus biologiques et donner un nouvel éclairage à l'interprétation de systèmes biologiques

et maladies génétiques.

Dans les sections suivantes, nous fournissons un bref aperçu et une contextualisation des sujets abordés dans le reste de ce manuscrit. Ceci est structuré en deux parties principales; la première se concentre sur les modèles algébriques des ARN et des protéines, tandis que la seconde partie fournit une description de nos études basées sur les agents sur les interactions biomoléculaires.

Bien que ces deux approches puissent être liées l'une à l'autre (comme indiqué au chapitre 5), nous les séparons pour permettre au lecteur d'identifier clairement le travail effectué principalement dans le cadre de l'Université de Camerino (partie 1) et celui qui est fruit de la collaboration avec l'Université Aix-Marseille (partie 2).

1.4 Modélisation algébrique de l'ARN et des protéines

La relation entre les structures et les fonctions est un sujet pertinent en biologie, dont l'investigation a reçu une contribution significative par différentes approches informatiques, du calcul de processus à l'analyse de données topologiques [9, 15, 51, 54, 68].

En particulier, les langages formels et les grammaires de graphes ont été appliqués avec succès dans la modélisation des propriétés qui corrèlent les fonctions exprimables par les molécules d'ARN et les sous-structures spécifiques impliquées dans leur repliement - le processus qui permet à un biopolymère linéaire d'atteindre une conformation tridimensionnelle en formant des liaisons hydrogène entre monomères non consécutifs [52, 73].

Dans ce manuscrit, nous faisons progresser cette approche et prouvons que la complexité des fonctions ARN peut être retracée à la potentialité interne de chaque nucléotide pour interagir avec les autres dans la même séquence. Ce résultat est obtenu en comparant le repliement de l'ARN avec celui réalisé par les protéines, afin d'identifier un niveau d'abstraction auquel ces deux classes de molécules présentent la même complexité structurelle et fonctionnelle. Nous appelons ce niveau *niveau de congruence*. Atteindre un tel objectif est possible grâce à l'expressivité des algèbres de processus [1], grâce auxquelles nous modélisons à la fois le repliement de l'ARN et des protéines.

Au cours de la seconde moitié du siècle dernier, l'étude des raisons d'existence de molécules si similaires a conduit à la formulation de l'hypothèse du monde de l'ARN: l'ARN pourrait être un " fossile " d'un monde à ARN, existait sur Terre avant l'apparition des cellules modernes, dans lequel l'ARN remplissait les rôles à la fois de l'ADN et des protéines. Cette théorie est encore très débattue et, en effet, au-delà de leurs similitudes, les protéines et les ARN présentent de profondes différences structurelles, qui affectent la manière dont ils remplissent leurs fonctions [30, 72].

Dans la première partie de notre travail, nous fournissons une description formelle du processus de repliement des protéines par rapport à celui des ARN. Notre objectif est d'identifier, en mettant en évidence leurs propriétés clés, des indices de validité de l'hypothèse RNA World. Nous concentrons notre étude sur les interactions réalisées par les unités élémentaires qui composent les ARN et les protéines (sur leurs séquences linéaires respectives), décrivant l'ensemble du processus de repliement comme le comportement résultant de telles interactions.

Par la suite, nous nous concentrons sur une classe de pathologies qui affecte les processus de repliement pour étudier comment les différences entre les composants structurels des protéines et des ARN provoquent une réponse différente à une modification de la voie de repliement correcte. Cette partie de notre étude part de la description formelle de la manière dont ces pathologies proviennent d'une erreur du code génétique (une mutation, en termes biologiques) et peuvent se propager à chaque étape de l'expression du gène, affectant à la fois l'ARN et les structures protéiques. Nous décrivons formellement comment la mutation d'un seul gène (mutation ponctuelle) peut modifier la conformation finale d'une protéine tout en étant inoffensive pour la structure des ARN. Nous montrons comment une pathologie bien connue affectant l'hémoglobine, la drépanocytose, peut être considérée comme une propriété globale des interactions entre les acides aminés ainsi que les nucléotides.

Nous faisons enfin un autre pas en avant, en émettant l'hypothèse des fonctions qui caractérisent le textit niveau de congruence mentionné ci-dessus et en explorant davantage l'applicabilité des algèbres de processus dans la modélisation des processus biologiques associés. Les modèles résultants formeront finalement la base d'une simulation multi-agent [43].

Dans une simulation à base d'agents, les agents sont des éléments logiciels discrets dont les interactions correspondent à celles effectuées par les composants du système modélisé, assez fidèlement au comportement réel d'un processus biologique [56]. Dans les algèbres de processus, les processus sont simultanés, autonomes et réactifs; toutes ces propriétés sont également partagées par les agents qui peuplent un environnement multi-agents, faisant des algèbres de processus des langages de spécification appropriés pour les systèmes multi-agents.

1.5 Modélisation et simulation basées sur des agents d'interactions biomoléculaires

Dans la seconde partie de ce manuscrit, nous proposons un simulateur multi-agents développé pour étudier les interactions moléculaires caractérisant les voies métaboliques, et analyser ses propriétés globales à partir des interactions locales [13, 69]. Nous sommes capables de simuler des réactions enzymatiques complètes en modélisant les molécules impliquées (enzymes, métabolites et complexes) comme des agents autonomes et interactifs.

Nous explorons les capacités du simulateur fourni pour traiter les interactions électrodynamiques

à longue distance qui façonnent le comportement des systèmes bimoléculaires, et analysons leur effet sur l'évolution d'une voie métabolique, telle que la glycolyse de la levure. Cette enquête a été menée dans le cadre de notre collaboration avec le Centre de Physique Théorique de l'Université Aix-Marseille.

Des études in vitro ont montré qu'une charge oscillant à haute fréquence (de l'ordre de $10^{10} - 10^{11}$ Hz) ne subit pas l'effet de criblage Debye par les ions du milieu et une macromolécule biologique se comporte comme une dipôle; des forces à longue portée peuvent être activées entre deux systèmes dipolaires résonants [31, 62].

Notre objectif est de fournir une validation in silico à ces expériences. Chaque molécule est représentée par un agent capable de percevoir l'environnement et les partenaires apparentés avec lesquels elle peut interagir. Une approche similaire peut également être adoptée en définissant un modèle de dynamique moléculaire; cependant, ce type de méthode place l'analyse à un niveau atomistique et les simulations associées ont une charge de calcul élevée. La compositionnalité des modèles basés sur des agents, au contraire, permet de mener l'étude à un niveau macromoléculaire, sans perdre en précision et en effectuant des simulations légères.

Cependant, comprendre et représenter dans son ensemble la dynamique des agents caractérisant une réaction métabolique réalisée par un grand nombre de molécules constitue toujours un enjeu majeur.

Pour cette raison, nous définissons également un nouveau paradigme de visualisation basé sur le concept de *interaction-as-perception*: chaque fois qu'une molécule en perçoit une autre avec laquelle interagir, un lien potentiel entre les deux est établi. De cette manière, nous pouvons dériver le graphique des perceptions à une étape donnée; sur ces graphiques, nous appliquons l'analyse des données topologiques pour capturer les interactions à 3 corps à travers l'interprétation des 2-simplices comme des structures observables, qui sont des coques convexes de trois points. Nous utilisons la formation 2-simplex comme sémantique valide pour représenter la dynamique globale du système.

Néanmoins, les processus biologiques sont des systèmes complexes dont le comportement global n'est pas toujours possible de prédire, en raison de l'incomplétude des données observées.Pour incorporer cette propriété dans un modèle à base d'agents d'un système biologique, les interactions des agents doivent avoir un caractère aléatoire ou l'environnement de simulation doit être non prévisible (cela implique que chaque exécution de la simulation est affectée par l'incertitude statistique). D'autres étapes sont nécessaires pour fournir une spécification efficace de l'environnement, espérons-le en faisant référence à la modélisation de calcul interactif [55].

1.6 Organisation du manuscrit

Chaque partie de ce manuscrit est corrélée à un premier chapitre d'introduction (chapitres 2 et 6), qui décrit les concepts biologiques de base nécessaires pour mieux comprendre nos études et les approches de modélisation que nous avons adoptées pour atteindre les résultats fournis.

La première partie consacrée à nos résultats est composée des chapitres 3, 4 et 5. Plus précisément:

- au chapitre 3, nous fournissons les modèles algébriques du repliement de l'ARN et des protéines et prouvons comment il est possible de définir formellement un niveau d'abstraction dans lequel de tels processus montrent une équivalence comportementale (niveau de congruence). Sa définition nous a permis d'émettre des hypothèses sur certaines des raisons qui conduisent l'évolution de la vie à la formation de protéines et de les assumer comme les principaux catalyseurs des processus biologiques.
- Le chapitre 4 analyse une classe de pathologies qui affecte les processus de repliement pour étudier comment les différences entre les composants structuraux des protéines et des ARN provoquent une réponse différente à une altération de la voie de repliement correcte.
- Au chapitre 5, nous explorons l'expressivité des algèbres de processus dans la modélisation des fonctions représentant le comportement des molécules d'ARN non codantes, suite à la caractérisation du niveau de congruence défini au chapitre 3. Sur la base de ces résultats, nous proposons une méthodologie adaptée pour générer une spécification algébrique d'une simulation multi-agents.

La deuxième partie du corps de ce manuscrit comprend les chapitres 7 et 8:

- au chapitre 7, nous décrivons un environnement de simulation dédié à l'étude des interactions moléculaires à longue distance dans les réactions métaboliques; nous proposons une approche à plusieurs corps, implémentée comme un système multi-agents (MAS).
- Le chapitre 8 fait progresser le chapitre précédent en utilisant la simulation MAS pour générer la dynamique du système biologique complexe afin de visualiser et de comprendre le comportement global de ce système; cela est possible grâce à l'introduction du paradigme de l'interaction comme perception.

Le chapitre 9 conclut le manuscrit et fournit nos réflexions sur les résultats obtenus, les limites rencontrées au cours des études et les améliorations possibles à prendre en compte pour les travaux futurs.

Part I

Algebraic Models

Background and Methods for the Part I

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

The first section gives an overview on the processes at the basis of protein folding and gene expression; we also introduce the RNA World hypothesis, addressed in Chapter 3. Finally, we briefly describe haemoglobin, a protein that we will analyse in Chapter 4 to model the behaviour of the sickle-cell anemia.

In the second section we provide the basic formalism at the basis of our modelling approaches; in particular, we will define CCS process algebra, Labeled Transition Systems and Hennessy-Milner logic; we also introduce the concept of agent, partly exploited in Chapter 5, even if we will deepen the the agent-based modelling and simulation in second part of this manuscript.

This chapter do not introduce any original content, except for section 2.2.4, where we propose an overview of our modelling approach.

2.1 Basic Introduction to Molecular Biology and Gene Expression

A molecule of deoxyribonucleic acid (DNA) consists of two strands of *nucleotides*, that is compounds made by a *sugar-phosphate group* covalently linked to a *nucleobase* (or just *base*).

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. An Adenine always pairs with a Thymine, and a Guanine always pairs with a Cytosine (that is, a purine always pairs with a pyrimidine). 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. are oriented in opposite polarities), twisted into a double helix.

The possibility of base-pairing nucleotides, also allow the DNA strands to be used as templates for generating a completely new DNA molecule in a process called *DNA replication*. This, as many other processes functions in cells, is performed by an enzyme, a molecule - in this a case protein - that acts as catalyst and helps complex reactions to occur. The replication process is carried out by the *DNA polymerase* enzyme and starts from a defined sequence of nucleotides, the *replication origins*.

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*, that is a mispaired nucleotide. For this reason, a specific complex of proteins 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 the *gene expression*.

Genes are specific sequences of nucleotides that contain the instructions for producing functional molecules, which can be either *proteins* or *functional-RNAs*. The process that converts the information encoded in the nucleotide sequence of a gene in the related functional product is defined as *gene expression*.

In this context, the roles of both intermediate and final product is performed by the RNA molecules.

The function of a protein is determined by its 3D structure, which is in turn determined by the sequence of its component molecules, the *amino acids*.

RNA is a linear molecule very similar to DNA, however it presents some differences. For the purposes of understanding the following chapters, it's important to consider that:

- RNA is composed by the bases Adenine (A), Guanine (G), and Cytosine (C), like DNA, but it contains Uracil (U) instead of Thymine (T). However, a Uracil molecule behaves similarly to Thymine and can base-pair with an Adenine.
- An RNA molecule is single-stranded, meaning that it can fold on itself and form three-dimensional structures. As we will see better in the following sections, this property allows some type of RNA molecules to carry out complex functions in cells.

All of the RNA in a cell is made by *transcription*, a process carry out by enzymes called *RNA polymerases*. During transcription one of the two strands of the DNA double helix acts as a template for the synthesis of RNA, so that, the nucleotide sequence of the RNA chain is built according to the base-pairing with that template. The RNA chain produced by transcription is

called the *transcript* and, because of complementarity, its sequence is equivalent to the sequence of the strand of DNA that doesn't act as template.

The vast majority of genes carried in a cell's DNA specify the amino acid sequence of proteins, and the RNA molecules that are copied from these genes (and that ultimately direct the synthesis of proteins) are collectively called *messenger RNA* (*mRNA*). In eukaryotes, each mRNA typically carries information transcribed from just one gene, coding for a single type of protein.

The final product of other genes, however, is the RNA itself. Important examples are:

- *ribosomal RNA* (*rRNA*), which forms the core of the ribosomes, on which mRNA is translated into protein;
- *transfer RNA* (*tRNA*), which forms the adaptors that select amino acids and hold them in place on a ribosome for their incorporation into protein;
- *microRNAs* (*miRNAs*), which serve as key regulators of eukaryotic gene expression.

Start and stop signals

When an RNA polymerase collides randomly with a piece of DNA, it sticks weakly to the double helix and then slides rapidly along. The enzyme latches on tightly only after it has encountered a region called a *promoter*, which contains a specific sequence of nucleotides indicating the starting point for RNA synthesis. Chain elongation then continues until the enzyme encounters a second signal in the DNA, the *terminator* (or *stop site*), where the polymerase halts and releases both the DNA template and the newly made RNA chain. The promoter is asymmetrical and binds the polymerase in only one orientation; thus, once properly positioned on a promoter, the RNA polymerase has no option but to transcribe the appropriate DNA strand (see Figure 2.1). Because tight binding is required for RNA polymerase to begin transcription, a segment of DNA will be transcribed only if it is preceded by a promoter sequence. This ensures that only those parts of a DNA molecule that contain a gene will be transcribed into RNA.

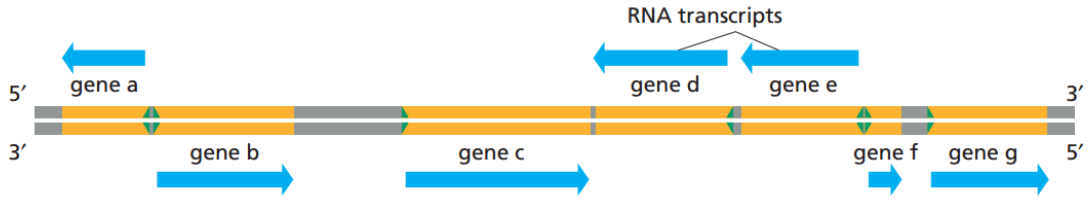


Figure 2.1 – The direction of transcription is determined by the orientation of the promoter at the beginning of each gene (green arrowheads). Bib. Ref. [2].

RNA processing

Because, in eukaryotic cells, DNA is enclosed within the nucleus, transcription takes place in the nucleus itself, but protein synthesis takes place on ribosomes in the cytoplasm. So, before a eukaryotic mRNA can be translated, it must be transported out of the nucleus through small pores in the nuclear envelope. Before a eukaryotic RNA exits the nucleus, however, it must go through several different *RNA processing* steps.

Two processing steps that occur only on transcripts destined to become mRNA molecules are *capping* and *polyadenylation*; for what we are interested in this discussion, we'll focus to a third step common to all kind of RNA, a process called *RNA splicing*.

Most eukaryotic genes have their coding sequences called *exons* (or *expressed sequences*) interrupted by noncoding intervening sequences, called *introns*. In the RNA splicing, intron sequences are removed from the newly synthesized RNA and exons are stitched together. Each intron contains a few short nucleotide sequences that act as cues for its removal. Guided by these sequences, an elaborate splicing machine (mainly composed by small nuclear RNAs or snRNAs) called *spliceosome* cuts out the intron sequence.

Many proteins are composed of a set of smaller *protein domains*. Some proteins are built from multiple copies of the same domain linked together in series. In eukaryotes, each protein domain is usually encoded by a separate exon.

2.1.1 RNA Translation

The genetic code

After the transcription of a nucleotide sequence of DNA into an mRNA molecule, the latter undergo the translation process, which synthesises a new protein.

Proteins are polymers, that is, they are molecules containing many copies of a smaller building block, covalently linked. The building blocks of proteins are amino acids, of which there are 20 that occur regularly in the proteins of living organisms and that are specified by the genetic code (for further details on the structure of proteins, see 2.1.2 on page 20).

Because there are only 4 different types of nucleotides in mRNA but 20 different types of amino acids in a protein, this translation cannot be performed by a direct one-to-one correspondence between a nucleotide in RNA and an amino acid in protein. The rules by which the nucleotide sequence of a gene, through the medium of mRNA, is translated into the amino acid sequence of a protein are known as the *genetic code*. The sequence of nucleotides in the mRNA molecule is read consecutively in groups of three. Because RNA is a linear polymer made of four different type of nucleotides, there are thus 4 x 4 x 4 = 64 possible combinations of three nucleotides: AAA,

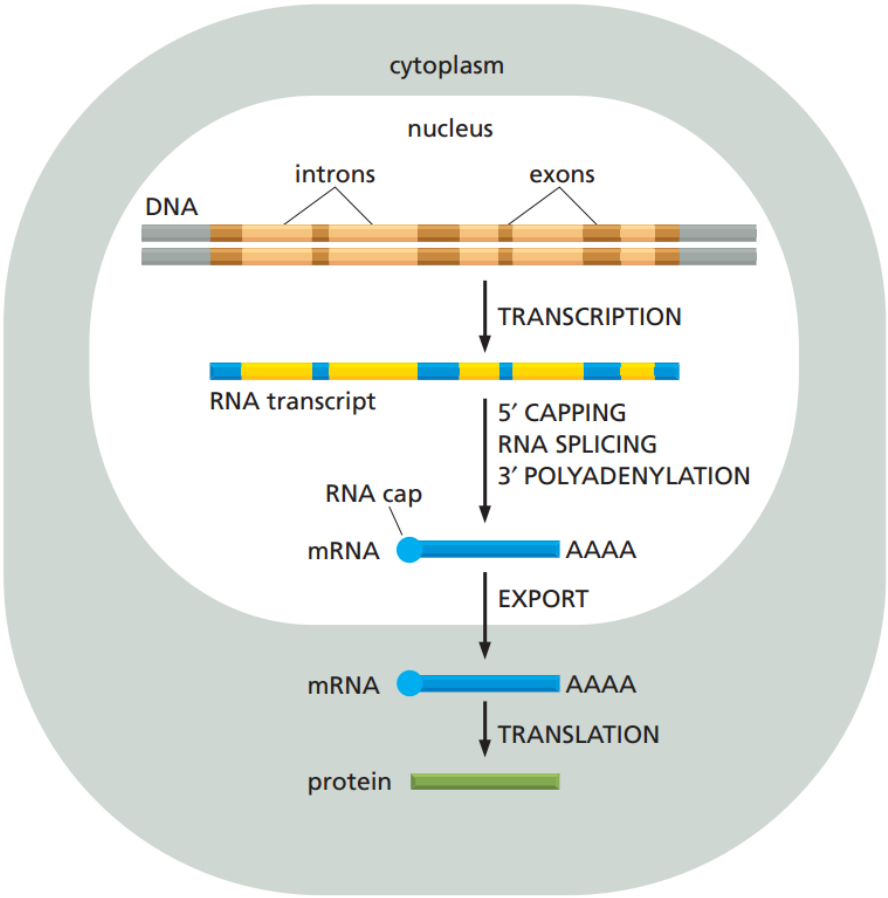


Figure 2.2 – Image that summarizes the steps of gene expression described so far. Bib. Ref. [2].

AUA, AUG, and so on. However, only 20 different amino acids are commonly found in proteins, so the code is redundant and some amino acids are specified by more than one triplet. Each group of three consecutive nucleotides in RNA is called a *codon*, and each specifies one amino acid.

RNA sequence can be translated in any one of three different *reading frames*, depending on where the decoding process begins. However, only one of the three possible reading frames specifies a correct protein.

The codons in an mRNA molecule do not directly recognize and bind the amino acids they specify. Rather, the translation of mRNA into protein depends on adaptor molecules, called *transfer RNAs* (*tRNAs*), that can recognize and bind to a codon at one site on their surface (*anticodon*) and to an amino acid that matches the codon at another site. The anticodon is a set

of three consecutive nucleotides that through base-pairing bind the complementary codon in an mRNA molecule.

The recognition of a codon by the anticodon on a tRNA molecule depends on the same type of complementary base-pairing used in DNA transcription. However, accurate and rapid translation of mRNA into protein requires a large molecular machine that moves along the mRNA, captures complementary tRNA molecules, holds them in position, and covalently links the amino acids that they carry so as to form a protein chain. This protein-manufacturing machine is the *ribosome*, which is a large complex made from more than 50 different proteins (the ribosomal proteins) and several RNA molecules called ribosomal RNAs (rRNAs).

Ribosomes

Ribosomes are composed of one large and one small subunit. The *small subunit* matches the tRNAs to the codons of the mRNA, while the *large subunit* catalyzes the formation of the peptide bonds that covalently link the amino acids together into a polypeptide chain. To begin the synthesis of a protein, the two subunits come together on an mRNA molecule, usually near its beginning (5′ end). The mRNA is then pulled through the ribosome like a piece of tape (see Figure 2.3 on the facing page). As the mRNA moves through it, the ribosome translates the nucleotide sequence into an amino acid sequence one codon at a time, using the tRNAs as adaptors. The translation of an mRNA begins with the codon AUG. The end of the protein-coding message is signaled by the presence of one of several codons called stop codons. These special codons — UAA, UAG, and UGA — are not recognized by a tRNA and do not specify an amino acid, but instead signal to the ribosome to stop translation.

2.1.2 Protein Structure and Folding

Proteins are chains of amino acids, with each amino acid joined to its neighbour by a specific type of covalent bond, called peptide-bond. All 20 of the common amino acids have a *carboxyl group* and an *amino group* bonded to the same carbon atom (the alpha-carbon). They differ from each other in their *side chains*, or *R groups*, which vary in structure, size, and electric charge, and which influence the solubility of the amino acids in water.

The specific characteristics of an amino acid are determined by the properties of its R group. The polarity of the group, which correlates with its solubility in water, is one critical property. The polarity of the R groups varies widely, from non-polar and hydrophobic (water-insoluble) to highly polar and hydrophilic (water-soluble). Therefore, the R groups of the 20 genetically encoded amino acids are clustered into the following categories: neutral (i.e., uncharged) and nonpolar, neutral and polar, charged.

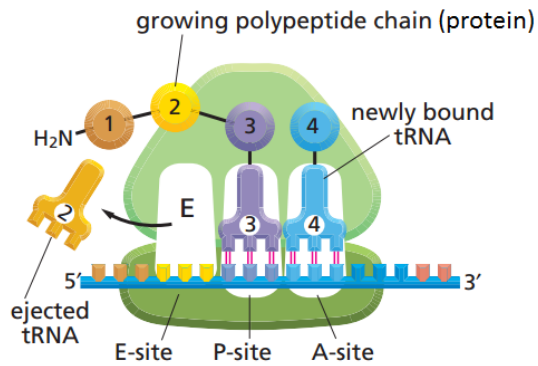


Figure 2.3 – Structure and functioning of a ribosome: the large subunit is represented in light green, the small subunit in dark green; the "blue" strand between them is the mRNA molecule. The small subunit is responsible to pair the anticodon of each tRNA molecule with corresponding mRNA anticodon. The large subunit form a peptide bond between the amino acids bound to the tRNAs, building, in this way, the polypeptide chain (i.e. the protein). Bib. Ref. [2].

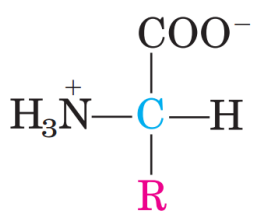


Figure 2.4 – General structure of an amino acid.

In analysing and describing the structure of proteins, it is useful to distinguish four levels of organization.

- The first level, the *primary structure* of a protein, is simply the sequence of amino acid residues in the polypeptide chain. The genetic code specifies the primary structure of a protein directly. The primary structure is thus just a one-dimensional string, specifying a pattern of chemical bonds; the remaining three levels depend on a protein's three-dimensional characteristics.

- *Secondary structure* refers to particularly stable arrangements of amino acid residues giving rise to recurring structural patterns.

- The *tertiary structure* of a protein refers to the usually compact, three-dimensionally folded arrangement that the polypeptide chain adopts under physiological conditions.

- Many proteins are composed of more than one polypeptide chain: *quaternary structure* refers to the way individual, folded chains associate with each other.

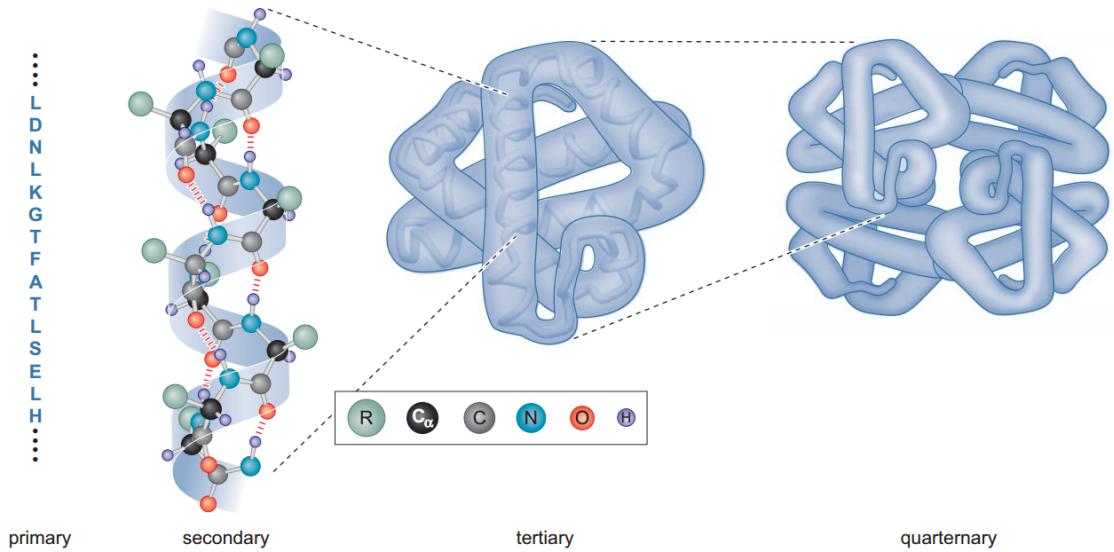


Figure 2.5 – Levels of protein structure, illustrated by hemoglobin. Bib. Ref. [87].

With regard to the three-dimensional structure of the proteins, we can outline three basic rules:

- the three-dimensional structure of a protein is determined by its amino acid sequence;
- the function of a protein depends on its structure;
- the most important forces stabilizing the specific structures maintained by a given protein are non-covalent interactions.

In other words we can state that *the primary structure of a protein determines how it folds up into a unique three-dimensional structure (stabilized by non-covalent interactions), and this in turn determines the function of the protein.*

The spatial arrangement of atoms in a protein is called its *conformation.* The possible conformations of a protein include any structural state that can be achieved without breaking covalent bonds. The conformations existing under a given set of conditions are usually the ones that are thermodynamically the most stable, having *the lowest Gibbs free energy (G).*

The unfolded state of a protein is characterized by a high degree of conformational entropy. This entropy tends to maintain the unfolded state. The chemical interactions that counteract these effects and stabilize the native conformation include disulphide bonds and the weak (non-covalent)

interactions, thus *hydrogen bonds, van der Waals, hydrophobic* and *ionic interactions*.

Individual covalent bonds that contribute to the native conformations of proteins, such as disulphide bonds, are much stronger than individual weak interactions. However, because they are so numerous, weak interactions predominate as a stabilizing force in protein structure. In general, *the protein conformation with the lowest free energy (that is, the most stable conformation) is the one with the maximum number of weak interactions*.

The contribution of weak interactions to protein stability can be understood in terms of the properties of water. When water surrounds a hydrophobic molecule, the optimal arrangement of hydrogen bonds results in a highly structured shell, or solvation layer, correlated with an unfavourable decrease in the entropy of the water. When non-polar groups are clustered together, there is a decrease in the extent of the solvation layer because each group no longer presents its entire surface to the solution. The result is a favourable increase in entropy. Hydrophobic amino acid side chains therefore tend to be clustered in a protein's interior, away from water. Folding of a polypeptide chain thus creates an "inside" and an "outside" and generates *buried* and *exposed* amino acid side chains.

The structure of a protein is also stabilized by hydrogen bond between the carbonyl group of one amino acid and the amide group of an other. One hydrogen bond seems to contribute little to the stability of a protein conformation, but the presence of hydrogen-bonding or charged groups without partners in the hydrophobic core of a protein can be so destabilizing that the favourable free-energy change realized by combining such a group with a partner can be greater than the difference in free energy between the folded and unfolded states.

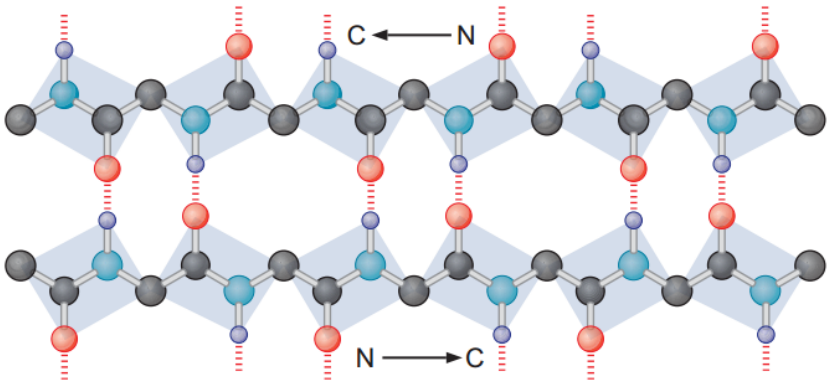


Figure 2.6 – Hydrogen bonds between amino acids are represented by the series of broken red lines. Bib. Ref. [87].

The interaction of oppositely charged groups that form an ion pair may also have a stabilizing

effect on one or more conformations of some proteins.

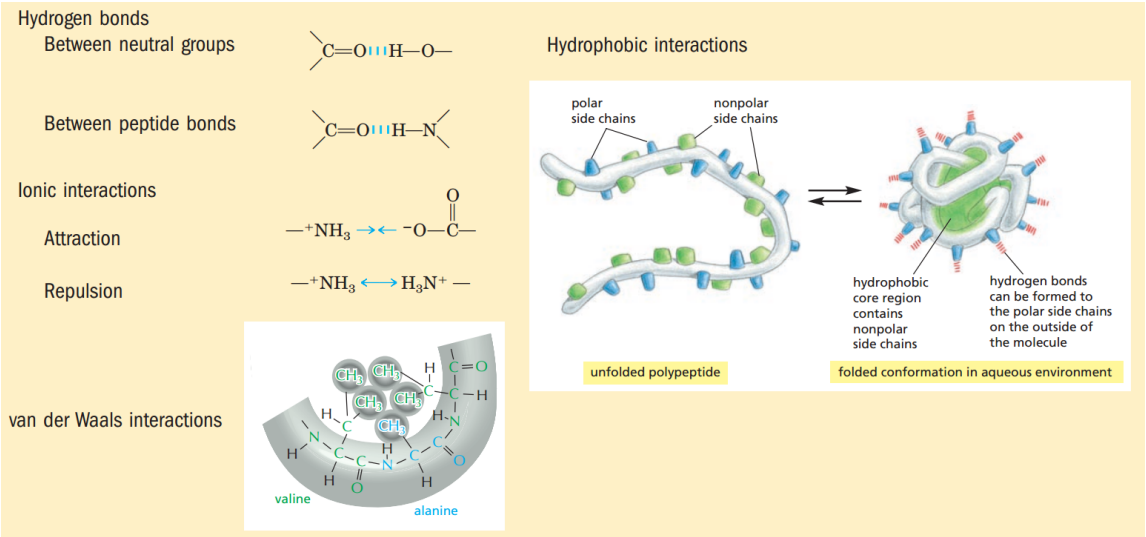


Figure 2.7 – Four types of non-covalent (Weak) interactions among amino acids in aqueous solvent. Image adapted from references [2] and [48].

2.1.3 Functional RNA

The first step a cell takes in reading out one of its many thousands of genes is to copy the nucleotide sequence of that gene into RNA. The process is called transcription because the information, though copied into another chemical form, is still written in essentially the same language, the language of nucleotides. RNA is a linear polymer made of four different types of nucleotide subunits linked together by phosphodiester bonds. Each nucleotide in RNA can contain one of the bases Adenine (A), Guanine (G), Cytosine (C) and Uracil (U). Adenine and Guanine are purine (two-rings bases) while Cytosine and Uracil are pyrimidine (single-ring bases).

The product of transcription of DNA is always single-stranded RNA but it can be of two different types:

- *messenger RNA*, which is eventually translated in the amino acid sequence of a protein;
- *functional RNA*, which can perform an active or a structural role inside the cell.

The single strand of RNA tends to assume a right-handed helical conformation dominated by base-stacking interactions, which are stronger between two purines than between a purine and pyrimidine or between two pyrimidines. Any self-complementary sequences in the molecule produce more complex structures.

Base pairing matches the following pattern: G pairs with C and A pairs with U. In addition to conventional Watson-Crick base pairs, RNA double helices often contain *noncanonical* (*non-Watson-Crick*) *base pairs*. There are more than 20 different types of noncanonical base pairs, involving two or more hydrogen bonds, that have been encountered in RNA structures. The most common are GU and GA pair. In addition, RNA structures frequently involve unconventional base pairing such as *base triples*, typically involving one of the standard base pairs. Noncanonical base pairs and base triples are important mediators of RNA self-assembly and of RNA-protein and RNA-ligand interactions.

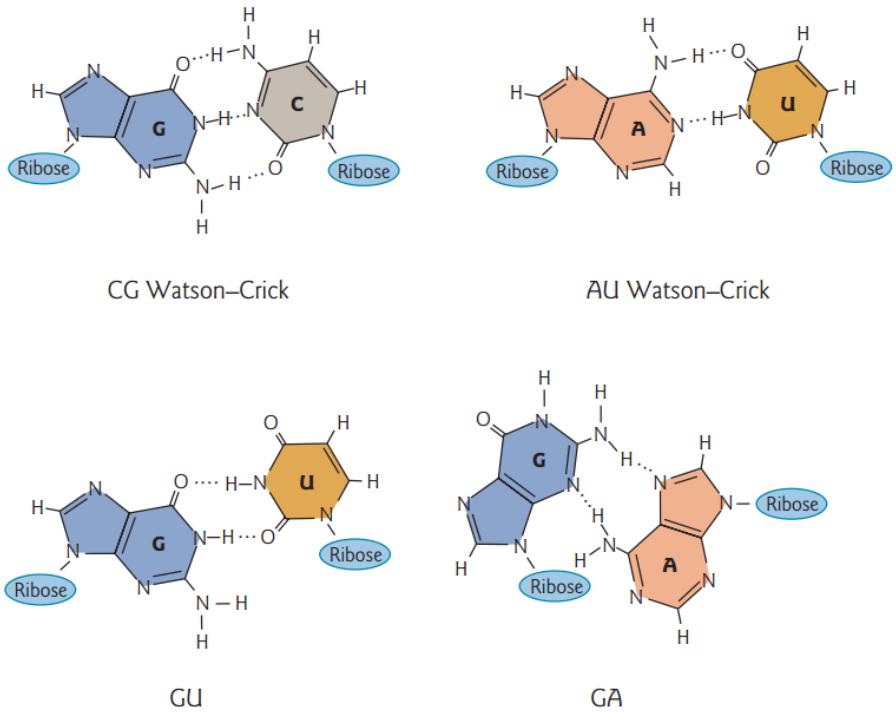


Figure 2.8 – Base pairs found in RNA double helices. Bib. Ref. [21].

RNA has no simple, regular secondary structure that serves as a reference point. The three-dimensional structures of many RNAs, like those of proteins, are complex and unique. Weak interactions, especially *base-stacking interactions*, play a major role in stabilising RNA structures.

Breaks in the helix caused by mismatched or unmatched bases in one or both strands are common and result in bulges or internal loops.

The analysis of RNA structure and the relationship between structure and function is an emerging field of inquiry that has many of the same complexities as the analysis of protein structure.

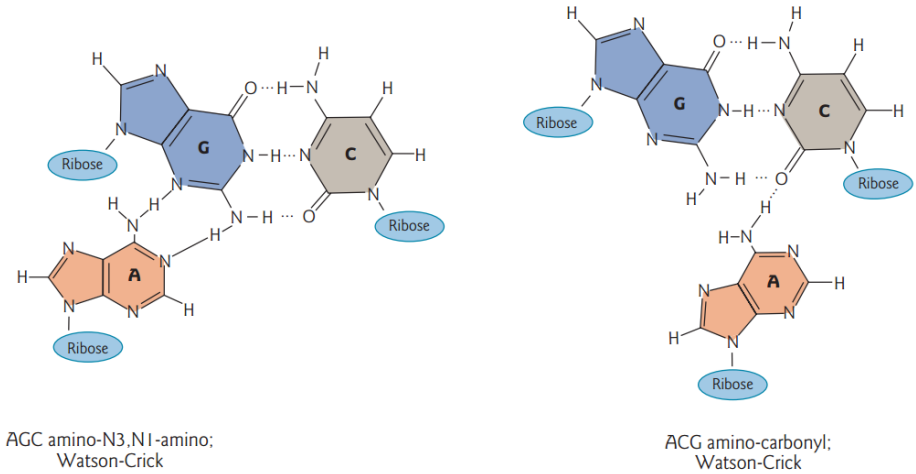


Figure 2.9 – The structures show two examples of hydrogen bonding that allow unusual triple base pairing. In both examples, a standard Watson-Crick GC pair forms the core of the triple. In the example on the left, the third base A is joined to G by two hydrogen bonds, while in the base triple on the right, A is joined to C by only one hydrogen bond. Bib. Ref. [21].

RNAs can adopt complex tertiary structures and can be biological catalysts. Such RNA enzymes are known as *ribozymes*, and they exhibit many of the features of a classical enzyme, such as an active site, a binding site for a substrate, and a binding site for a co-factor, such as a metal ion.

2.1.4 RNA World

The majority of the agents involved in the various stages of gene expression are proteins (e.g. RNA polymerases) or are composed in part by proteins (e.g. ribosomes). Therefore, nucleic acids are required to direct the synthesis of proteins, and proteins are required to synthesize nucleic acids; so we might ask how this system of interdependent components could have arisen. One view is that an RNA world existed on Earth before modern cells appeared. According to this hypothesis, RNA, which today serves as an intermediate between genes and proteins, both stored genetic information and catalysed chemical reactions (like nucleic acids synthesis) in primitive cells. Only later in evolutionary time, DNA took over as the genetic material and proteins became the major catalysts and structural components of cells. If this idea is correct, then the transition out of the RNA world was never completed, since RNA still catalyses several fundamental reactions in modern cells. These RNA catalysts, including the ribosome and RNA-splicing machinery, can thus be viewed as molecular fossils of an earlier world.

An RNA molecule could in principle guide the formation of an exact copy of itself. In the first

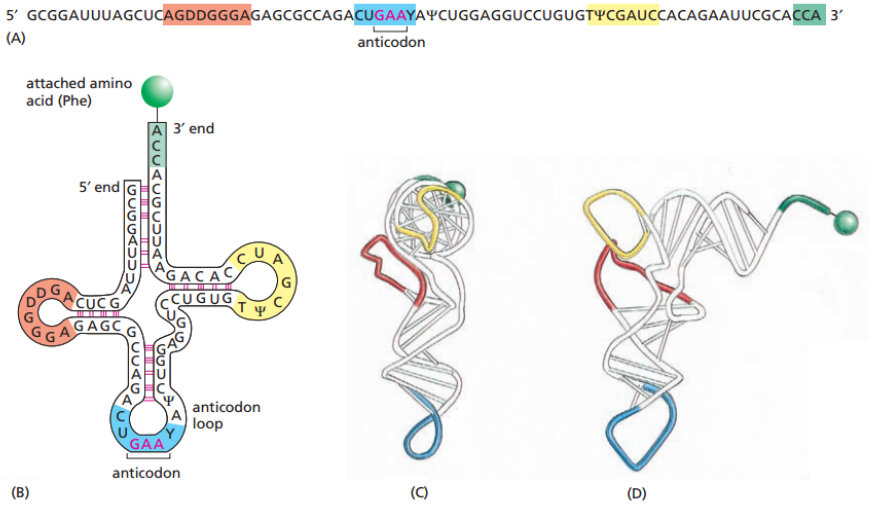


Figure 2.10 – tRNA taken as an example of RNA folding: (A) the primary structure, the linear nucleotide sequence of the molecule; (B) the secondary structure, represented as the conventional "cloverleaf" structure, used to show the complementary base-pairing (red lines) that creates the double-helical regions of the molecule; (C and D) the tertiary structure, the actual L-shaped molecule. Bib. Ref. [2].

step the original RNA molecule acts as a template to form an RNA molecule of complementary sequence; in the second step this complementary RNA molecule itself acts as a template, forming RNA molecules of the original sequence.

But the efficient synthesis of polynucleotides by such complementary templating mechanisms also requires catalysts to promote the polymerization reaction: without catalysts, polymer formation is slow, error-prone, and inefficient. RNA is synthesized as a single-stranded molecule, but complementary base-pairing can occur between nucleotides in the same chain. This base-pairing, along with "nonconventional" hydrogen bonds, can cause each RNA molecule to fold up in a unique way that is determined by its nucleotide sequence. Such associations produce complex three-dimensional patterns of folding, where the molecule adopts a unique shape. RNA molecules, with their folded shapes, can serve as enzymes.

Thus, the unique potential of RNA molecules to act both as information carriers and as catalysts is thought to have enabled them to play the central role in the origin of life.

As cells more closely resembling present-day cells appeared, it is believed that many of the functions originally performed by RNA were taken over by molecules more specifically fitted to the tasks required. Eventually DNA took over the primary genetic function, and proteins

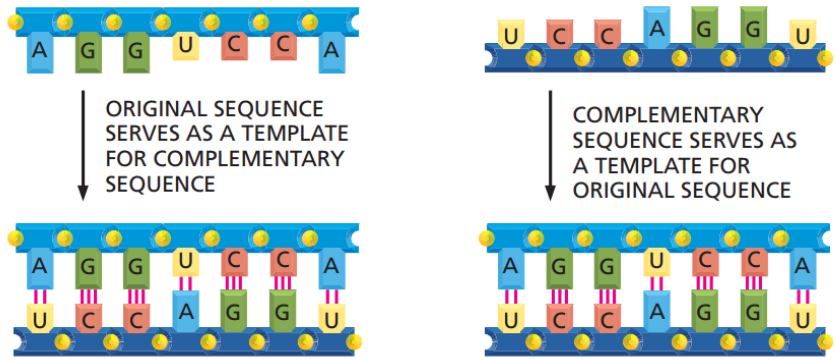


Figure 2.11 – RNA self-replication process. Bib. Ref. [2].

became the major catalysts, while RNA remained primarily as the intermediary connecting the two.

2.1.5 Haemoglobin and Anaemias

Hemoglobin is a protein found in erythrocytes (red blood cells) that carries oxygen from the lungs to the body's tissues and returns carbon dioxide from the tissues back to the lungs. It contains four polypeptide subunits (called *globins*) and four prosthetic groups (one for each subunit), in which an iron atom is able to bind the oxygen. The protein portion, consists of two α chains and two β chains. The subunits of hemoglobin are arranged in symmetric pairs, each pair having one α and one β subunit (Figure 2.13) [48].

The four oxygen-binding sites interact with each other, allowing a conformational (allosteric) change in the molecule as it binds and releases oxygen. This structural shift enables the haemoglobin molecule to efficiently take up and release four oxygen molecules in an all-or-none fashion. When oxygen binds to the first subunit of deoxyhaemoglobin (deoxygenated haemoglobin) it increases the affinity of the remaining subunits for oxygen. As additional oxygen is bound to the second and third subunits oxygen binding is further, incrementally, strengthened until haemoglobin is fully saturated with oxygen. As oxyhaemoglobin circulates to deoxygenated tissue, oxygen is incrementally unloaded and the affinity of haemoglobin for oxygen is reduced. When haemoglobin is deoxygenated, it is also said to be in its *T state*, or tense state; whet it is oxygenated it is said to be in its *R state*, or relaxed state.

A single nucleotide change (mutation) in the β-globin gene produces a β-globin subunit that differs from normal β-globin only by a change from glutamic acid to valine (in the HbS disease) or to lysine (in the HbC disease) at the sixth amino acid position [2]. Both HbS and HbC diseases are hereditary; humans carry two copies of each gene (one inherited from each parent); a point

mutation in one of the two β-globin genes generally causes no harm to the individual, as it is compensated for by the normal gene. However, an individual who inherits two copies of the mutant β-globin gene displays the symptoms of the anaemia.

Haemoglobin also plays an important role in maintaining the shape of the red blood cells. In their natural shape, red blood cells are round with narrow centres.

In sickle-cell haemoglobin the mutation of Glu 6 in the β chain to Val creates an hydrophobic patch on the surface of the molecule. This patch fits and can bind into a hydrophobic pocket in the deoxygenated form of another haemoglobin molecule. This process forms fibrous precipitates, which produce the characteristic sickle shape of affected red blood cells. Because these cells are more fragile and frequently break in the bloodstream, patients with this potentially life-threatening disease have fewer red blood cells than usual, a deficiency that can cause weakness, dizziness, headaches, pain, and organ failure (Figure 2.14) [2].

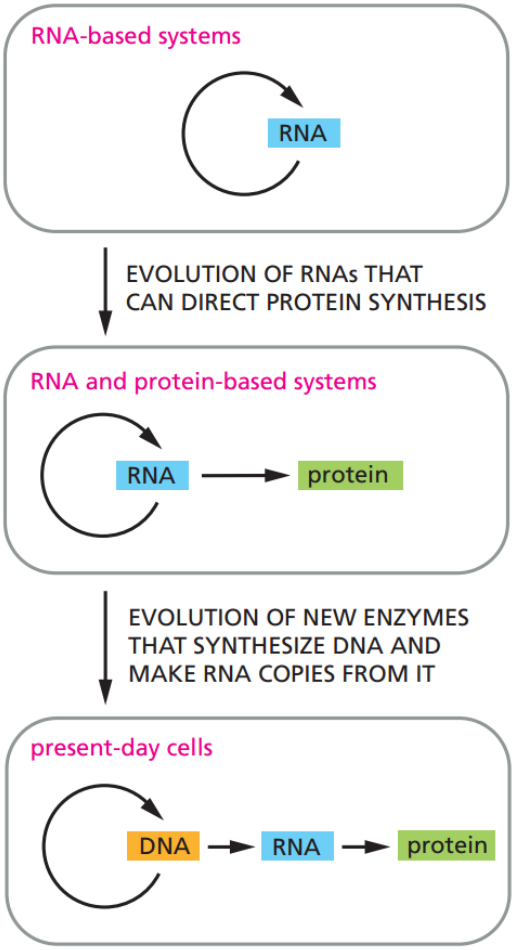


Figure 2.12 – Evolution of RNA from the "RNA world" to present-day cell. Bib. Ref. [2].

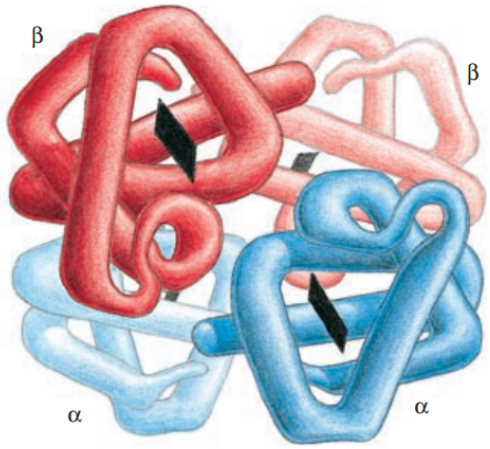


Figure 2.13 – The mammalian hemoglobin molecule with its two α and two β subunits. Bib. Ref. [2].

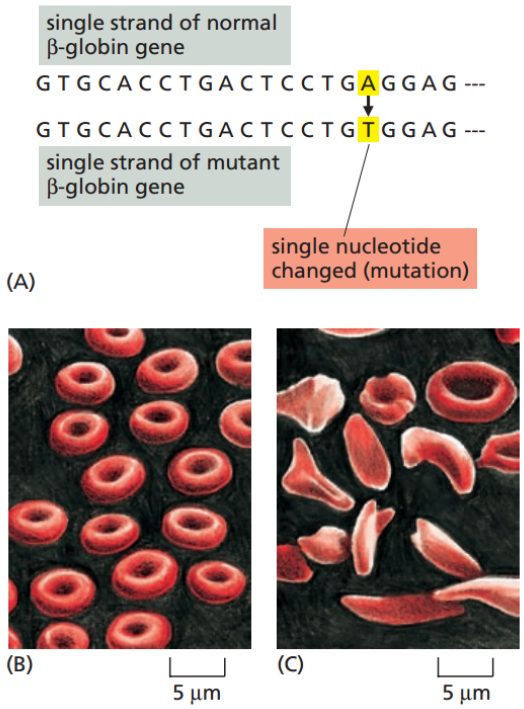


Figure 2.14 – A single nucleotide change causes the disease sickle-cell anaemia Bib. Ref. [2].

2.2 Introduction to the Algebraic Modelling of Biological Systems

2.2.1 Calculus of Communicating Systems

The formal models we provide in this manuscript are specified with Milner's CCS process algebra. It consists of a collection of constructors for building a new process description from existing ones, by representing them as systems that exhibit a 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 output or input action on the channel a is indicated using the labels $\overline{a}$ or a respectively.

Let $\mathscr{P}$, $\mathscr{Q}$ be processes, the main process constructors are:

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

This section presents an essential description of the concepts at the basis of the models proposed in this manuscript. The description is mainly based on the book of Aceto et al [1].

CCS syntax

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

The set $\mathscr{P}$ of the CCS expression, is given by the following grammar:

$$P, Q ::= K \quad | \quad \alpha.P \quad | \quad \sum_{i \in I} P_i \quad | \quad P|Q \quad | \quad P[f] \quad | \quad P \setminus L$$

Where:

- K is a process name in $\mathscr{K}$;

- α is an action in **Act**;

- I is a possibly infinite index set;

- $f : \mathbf{Act} \to \mathbf{Act}$ is a relabelling function satisfying the following constraints:

  - $f(\tau) = \tau$

  - $f(\overline{a}) = \overline{f(a)}$ for each label a;

- L is a set of labels from $\mathscr{L}$.

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

CCS Structural Operational Semantics

$\alpha \in$ **Act** and $a \in \mathscr{L}$,

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \text{Action prefixing}$$

$$\frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j} \text{ where } j \in I \qquad Summation$$

$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \text{Parallel composition (rule 1)}$$

$$\frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \qquad \text{Parallel composition (rule 2)}$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad \text{Parallel composition (rule 3)}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \text{ where } \alpha \notin L \qquad \text{Restriction}$$

$$\frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]} \qquad \text{Relabelling}$$

$$\frac{P \xrightarrow{\alpha} P'}{K \xrightarrow{\alpha} P'} \text{ where } K \stackrel{\text{def}}{=} P \qquad \text{Constant definition}$$

Strong bisimulation

A binary relation $\mathscr{R}$ over the set of states of an LTS is a bisimulation iff whenever $s_1 \mathscr{R} s_2$ and α is an action:

- if $s_1 \xrightarrow{\alpha} s'_1$, then there is a transition $s_2 \xrightarrow{\alpha} s'_2$ such that $s'_1 \mathscr{R} s'_2$;
- if $s_2 \xrightarrow{\alpha} s'_2$, then there is a transition $s_1 \xrightarrow{\alpha} s'_1$ such that $s'_1 \mathscr{R} s'_2$.

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.

2.2.2 Labelled Transition Systems

The biological processes described in this manuscript have been modelled as the result of sub-processes that proceed along a path made by discrete states; this aspect has been highlighted by describing all the modelled processes via Labelled Transition Systems (LTSs) [46]; 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 a and become a process P', we write $P \xrightarrow{a} P'$ [1].

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

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

The LTSs in this manuscript have been generated via the automated tool CAAL - Concurrency Workbench, Alborg Edition [3].

2.2.3 Hennessy-Milner Logic

We represent the gene sequences as Hennessy-Milner formulae that can be satisfied by the gene expression processes.

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

$$\mathtt{F},\mathtt{G} ::= tt|ff|\mathtt{F} \wedge \mathtt{G}|\mathtt{F} \vee \mathtt{G}|\langle w\rangle\mathtt{F}|[w]\mathtt{F}$$

where $a \in Act$, tt and ff are used to denote respectively“true” and “false” [1].

The meaning of a formula in $\mathscr{M}$ is given by characterizing 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 $\mathtt{F} \wedge \mathtt{G}$ (respectively, $\mathtt{F} \vee \mathtt{G}$) if and only if it satisfies both $\mathtt{F}$ and $\mathtt{G}$ (respectively, either $\mathtt{F}$ or $\mathtt{G}$).
- A process satisfies $\langle w\rangle\mathtt{F}$ for some $w \in Act$ if and only if it affords an w-labelled transition leading to a state satisfying $\mathtt{F}$.

- A process satisfies $[w]\mathtt{F}$ for some $w \in Act$ if and only if all of its w-labelled transitions lead to a state satisfying $\mathtt{F}$.

In the HML formulae provided in this manuscript, an output or input action on the channel w is indicated using the labels $'w$ or w respectively.

2.2.4 From Algebraic to Agent-based Models

Agents are software peaces able to perceive changes in the environment and react to them.

To provide some basic formalism, a *reactive agent* is efined by a 6-tuple $\langle E, Per, Ac, see, do, action\rangle$ 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: $E \rightarrow Per$
- $action$: $Per \rightarrow Ac$
- do: $Ac \times E \rightarrow E$

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

In an agent-based simulation, agents interactions correspond to those performed by the components of the modelled system, quite faithfully to the actual behaviour of a biological process [56]. 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 multiagent systems.

However, CCS is a process-based specification language while a multiagent system is an agent-based model suitable for computer simulation. Being able to express agents as processes, in general, allows one to verify if the behaviour of a specified system conforms the simulated model. Moreover, in the the specific case of the RNA domain, these methods are used to verify the interaction properties, among agents as well as between agents and the environment (when the environment is modelled as a process). A schematic representation of the transition from the biological domain (experimental data) to the multiagent simulation, via process-based models, is provided in Figure 2.15.

We will deepen the agent-based modelling and simulation of molecular interactions in the Part II of this manuscript.

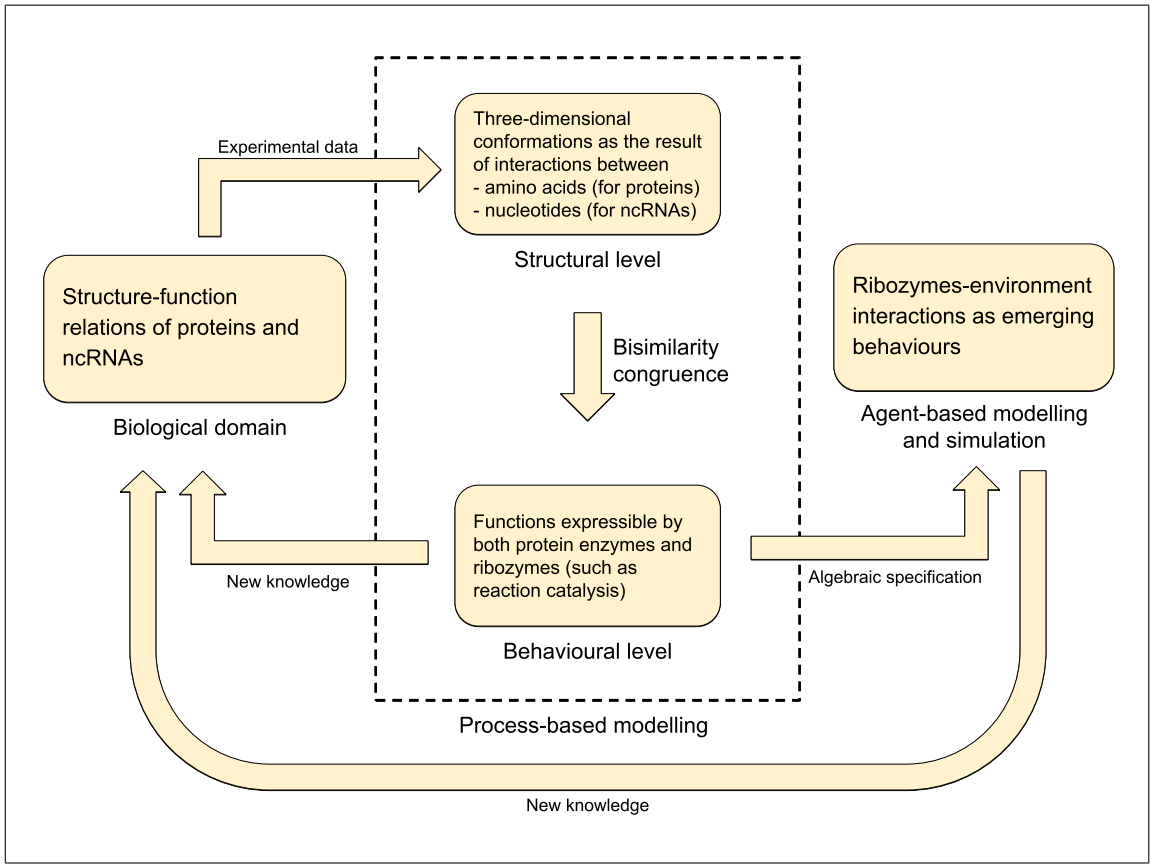


Figure 2.15 – Schematic representation of the modelling approach proposed in our work. *Experimental data* retrieved from *in vivo* and *in vitro* studies on proteins and RNAs provide the fundamental information and knowledge upon which we constructed the CCS models of their respective folding processes. At the *structural level*, these models correlate the interactions between the elementary units of proteins and RNAs (amino acids and nucletides, respectively) to their three-dimensional conformations. Discovering an abstraction level in which the two kinds of folding processes are *bisimilar*, gave us the perspective needed to identify a class of functions of the same complexity, which can be equally performed by proteins and RNAs; it also yielded *new knowledge* on the *biological domain* [50]. In Chapter 5, we will outline an *algebraic specification* of this class of functions, which will be at the basis of an *agent-based model*, eventually resulting in the related computer simulation.

Process calculi may reveal the equivalence underlying RNA and proteins

3.1 Introduction

RNAs (ribonucleic acids) and proteins are two classes of molecules that have drawn the interest of different scientific disciplines due to the fundamental roles they play in many biological processes. The study of their folding processes represents an important issue to discover the qualitative information underlying the relation between their structures and functions.

They perform a similar pathway from their linear sequence to a three-dimensional conformation, which in turn allows them to carry out almost the same functions (i.e. catalytic and structural roles). Investigating the reasons of existence of such similar molecules leads to the formulation of the RNA World hypothesis: RNA might be a "fossil" of an RNA world, existed on Earth before modern cells appeared, in which RNA fulfilled the roles of both DNA and proteins. This theory is still highly debated [30, 72]; indeed, beyond their similarities, proteins and RNAs show profound structural differences, which affect the way they perform their functions.

This chapter is intended to provide a formal description of the folding process of proteins compared to the one of RNAs; our purpose is to identify, by highlighting their key properties, clues of the validity of the RNA World hypothesis. We focus our study on the interactions carried out by the elementary units that compose RNAs and proteins (on their respective linear sequences), describing the whole folding process as the resulting behaviour of such interactions.

3.2 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 modelled as processes. This approach has been applied to study biological systems by modelling entire molecules [9, 68], and can be extended to analyse their substructures or even their elementary units, since it allows describing every kind of interaction they perform; it is also possible to identify similarities among different classes of molecules and in the functions they carry out.

The specification language that better suits our modelling of RNA and protein folding is the process algebra called CCS (Calculus of Communicating Systems), proposed by Milner in 1989 [58]; thanks to this language it has been possible to define the congruence of the folding processes in terms of *behavioural equivalence* and also to perform the model checking with the aid of automated tools.

The whole folding process has been modelled as the result of sub-processes that proceed along a path made by discrete states; this aspect has been highlighted by describing all the modelled processes via Labelled Transition Systems (LTSs) [46].

We want to point out that some aspects contributing to the folding process that can be considered relevant from a biological point of view, like the role of helping molecules (e.g. the modulation performed by Mg^{2+} on the RNA folding or the action of molecular chaperones in protein folding [35, 37]), have not been taken into account in our model. This choice has been driven by the idea of describing the folding process as a behaviour strictly resulting from the peculiar properties of the interactions carried out by nucleotides and amino acids (in their respective linear sequences) and of the informational content brought along by each of them.

If on the one hand such approach led us to define an abstraction of the actual folding mechanisms, on the other it allowed 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 wanted to prove that the inner potentiality of each elementary unit to interact with the others (in the same sequence) is the main property that determines the different complexity eventually reachable by the two classes of molecules.

To demonstrate such statement, we started by defining the models of the two folding processes as a sequence of folding steps, each contributing with a new weak interaction between two units of the linear sequence of the molecule. In order for a folding step to take place, the weak interaction must cause a reduction in the free energy of the system.

Because the folding process relies mainly in the formation of weak, noncovalent interactions in both RNAs and proteins, the stabilising function performed by covalent bonds (like the disulphide bridges between Cys residues) can be considered negligible for the purpose of our

modelisation.

Even if the weak interactions taken into account are the same for RNAs and proteins, the rules that allow two nucleotides to interact are different from the rules that determine the interplay of two amino acids; we modelled such rules starting from the biochemical properties of the weak interactions. Hence, we needed to define two different models, one for each class of molecules.

The differences highlighted affect the whole folding process and led our models to show different traces, which means different sequences of transitions in their respective LTSs.

However, the expressiveness of the modelling approach based on process algebras allowed 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 lead to the formation of structures with the same complexity and hence capable to express the same functions.

If the same abstraction level might represent the actual folding process of RNAs and proteins, there would be no reasons for the existence of both these two classes of molecules in cell, showing the same behaviour. Conversely, according to the RNA World hypothesis, the fact that such similar molecules can still be found in nature, allows us to hypothesise that, in the early stages of cell evolution, RNA might be the only type of molecule that performed structural and catalytic activities; as the complexity of cells increased, also emerged the necessity of molecules able to carry out more complex tasks. Towards the RNA World hypothesis, these molecules (proteins) might be evolved on the same property that was characterising RNAs of being a linear sequence of elementary units able to fold up to a three-dimensional structure, driven by the free energy reduction. As we show with our models, the cells cope with this necessity by the formation of molecules whose elementary units (the amino acids) are able to perform more complex interactions than nucleotides. Our results concern the RNA World hypothesis due to the interpretation of the behavioral equivalence of RNA and protein folding under specific restrictions (as in Theorem 1).

In the models of the folding process that we have defined, the weak interactions are classified in three main categories:

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

The hydrogen bond can be defined as an electrostatic interaction, but, due to its distinctive properties and the fundamental role it carries out in the folding process, it has been represented

separately. Moreover, the model of each weak interaction has to be contextualised in the folding step it belongs to.

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 weak-interactions.

A *Folding Step* process ($\mathcal{F}^s$) ensures that each sub-process complies with the specific restrictions on its input (according to the descriptions given below in this document) and that the interaction has a negative *free-energy change*, ΔG, which measures the amount of disorder created in a system when an interaction takes place. It can assume the value negative (`ndg`), positive (`pdg`) or zero (`zdg`). An interaction is *energetically favorable* if it creates disorder by decreasing the free energy of the system, namely if it has a negative ΔG; this condition is essential for an interaction to be carried out.

In order to meet the last requirement, both the *RNA Folding Step* ($\mathcal{F}^s_{rna}$) and *Protein Folding Step* ($\mathcal{F}^s_p$) processes are placed in parallel composition with the process $\Delta G_{\mathcal{F}^s}$, which represents the ΔG variation during folding. In this way the whole folding processes, $\mathcal{F}_{rna}$ and $\mathcal{F}_p$ respectively, can be defined as following:

$$\mathcal{F}_{rna} \stackrel{\text{def}}{=} (\mathcal{F}^s_{rna} | \Delta G_{\mathcal{F}^s}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};$$

$$\mathcal{F}_p \stackrel{\text{def}}{=} (\mathcal{F}^s_p | \Delta G_{\mathcal{F}^s}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};$$

$$where\ \Delta G_{\mathcal{F}^s} \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta G_{\mathcal{F}^s} + \overline{\mathtt{ndg}}.\Delta G_{\mathcal{F}^s} + \overline{\mathtt{zdg}}.\Delta G_{\mathcal{F}^s}.$$

Both $\mathcal{F}^s_{rna}$ and $\mathcal{F}^s_p$ are structured in sub-processes that can be clustered in three main groups (see Figure 3.1):

group 1 determines the type of the elementary units involved in the ongoing folding step, the interaction that is going to establish 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 modelisation, 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

RNA Folding Step

group 1

$$\mathcal{F}^{s}_{rna} \stackrel{def}{=} \mathtt{ub}.\mathcal{I}1_{n} + \mathtt{ub}.\mathcal{I}2_{n} + \mathtt{srsr}.\mathcal{I}1_{n} + \mathtt{drdr}.\mathcal{I}1_{n} + \mathtt{srdr}.\mathcal{I}1_{n} + \mathtt{tpb}.\mathcal{I}1_{n};$$

$$\mathcal{I}1_{n} \stackrel{def}{=} \mathtt{ub}.\Delta G_{\mathcal{I}^{e}_{b}} + \mathtt{srsr}.\Delta G_{\mathcal{I}^{e}_{b}} + \mathtt{drdr}.\Delta G_{\mathcal{I}^{e}_{b}} + \mathtt{srdr}.\Delta G_{\mathcal{I}^{e}_{b}} + \mathtt{tpb}.\Delta G_{\mathcal{I}^{e}_{b}};$$

$$\mathcal{I}2_{n} \stackrel{def}{=} \mathtt{ub}.\Delta G_{\mathcal{P}_{b2}} + \mathtt{ub}.\Delta G_{\mathcal{I}^{ho}_{b}} + \mathtt{srsr}.\Delta G_{\mathcal{P}_{b3}} + \mathtt{drdr}.\Delta G_{\mathcal{P}_{b3}} + \mathtt{srdr}.\Delta G_{\mathcal{P}_{b3}};$$

$$\Delta G_{\mathcal{I}^{e}_{b}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{I}^{e}_{b};$$

$$\Delta G_{\mathcal{I}^{ho}_{b}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{I}^{ho}_{b};$$

$$\Delta G_{\mathcal{P}_{b2}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{P}_{b2};$$

$$\Delta G_{\mathcal{P}_{b3}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{P}_{b3};$$

group 2

$$\mathcal{P}_{b2} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}1_{b2};$$

$$\mathcal{B}1_{b2} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}2_{b2};$$

$$\mathcal{B}2_{b2} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}3_{b2} + \overline{\mathtt{srsr}}.\mathcal{F}^{s}_{rna} + \overline{\mathtt{drdr}}.\mathcal{F}^{s}_{rna} + \overline{\mathtt{srdr}}.\mathcal{F}^{s}_{rna};$$

$$\mathcal{B}3_{b2} \stackrel{def}{=} \overline{\mathtt{srdr}}.\mathcal{F}^{s}_{rna};$$

$$\mathcal{P}_{b3} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}1_{b3};$$

$$\mathcal{B}1_{b3} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}2_{b3} + \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{rna};$$

$$\mathcal{B}2_{b3} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}3_{b3} + \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{rna};$$

$$\mathcal{B}3_{b3} \stackrel{def}{=} \overline{\mathtt{tpb}}.\mathcal{F}^{s}_{rna};$$

group 3

$$\mathcal{I}^{e}_{b} \stackrel{def}{=} \overline{ii}.\mathcal{F}^{s}_{rna} + \overline{vdwi}.\mathcal{F}^{s}_{rna};$$

$$\mathcal{I}^{ho}_{b} \stackrel{def}{=} hbi.\mathtt{I}_{rna};$$

$$\mathtt{I}_{rna} \stackrel{def}{=} \overline{\mathtt{bb}}.\mathcal{S};$$

$$\mathcal{S} \stackrel{def}{=} \overline{\mathtt{sb}}.\mathcal{F}^{s}_{rna};$$

Protein Folding Step

group 1

$$\mathcal{F}^{s}_{p} \stackrel{def}{=} \mathtt{aa}.\mathcal{I}_{aa} + \mathtt{aa}.\Delta G_{\mathcal{I}^{h}_{aa}};$$

$$\mathcal{I}_{aa} \stackrel{def}{=} \mathtt{aa}.\Delta G_{\mathcal{I}^{e}_{aa}} + \mathtt{aa}.\Delta G_{\mathcal{P}_{aa}};$$

$$\Delta G_{\mathcal{I}^{e}_{aa}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{I}^{e}_{aa};$$

$$\Delta G_{\mathcal{I}^{h}_{aa}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{I}^{h}_{aa};$$

$$\Delta G_{\mathcal{P}_{aa}} \stackrel{def}{=} \mathtt{ndg}.\mathcal{P}_{aa};$$

group 2

$$\mathcal{P}_{aa} \stackrel{def}{=} \mathtt{aa1fnh}.\mathtt{NH}_{aa1} + \mathtt{aa1fco}.\mathtt{CO}_{aa1};$$

$$\mathtt{NH}_{aa1} \stackrel{def}{=} \mathtt{aa2fco}.\mathtt{CO}_{aa2};$$

$$\mathtt{CO}_{aa1} \stackrel{def}{=} \mathtt{aa2fnh}.\mathtt{NH}_{aa2};$$

$$\mathtt{CO}_{aa2} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}_{aa};$$

$$\mathtt{NH}_{aa2} \stackrel{def}{=} \mathtt{hb}.\mathcal{B}_{aa};$$

$$\mathcal{B}_{aa} \stackrel{def}{=} \overline{\mathtt{paa}}.\mathcal{F}^{s}_{p};$$

group 3

$$\mathcal{I}^{e}_{aa} \stackrel{def}{=} \overline{ii}.\mathcal{F}^{s}_{p} + \overline{vdwi}.\mathcal{F}^{s}_{p};$$

$$\mathcal{I}^{h}_{aa} \stackrel{def}{=} \mathtt{hlsc}.\mathtt{O}_{p} + \mathtt{hbsc}.\mathtt{I}_{p};$$

$$\mathtt{O}_{p} \stackrel{def}{=} \overline{\mathtt{esc}}.\mathcal{F}^{s}_{p};$$

$$\mathtt{I}_{p} \stackrel{def}{=} \overline{\mathtt{bsc}}.\mathcal{F}^{s}_{p};$$

Figure 3.1 – In this figure a comparative representation of the two folding step models (RNA on the left side and protein on the right) is proposed. Each model can be ideally divided into three groups of sub-processes; they have the function of determining the type of interacting elementary units and the interaction that is going to bind them (*group 1*), modelling the formation of hydrogen bonds (*group 2*) and of ionic and van der Waals interactions (*group 3*). For detailed information on the construction of the models and on the meaning of the symbols used, see Section 1 and 2 of the Supplementary Information.

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) has been modelled generalising this process as an interaction between a purine (adenine or guanine - labelled `dr`, since they are **d**ouble-**r**ing bases) and a pyrimidine (uracil and cytosine - **s**ingle-**r**ing bases and hence labelled `sr`) or between a two paired bases and a third base (also in this case, a generic purine or pyrimidine). The base pairing is symmetric, thus: `srdr` = `drsr`.

Regarding the number of hydrogen bonds allowed in a base pair, in our models they must be at least two and at most three; the number of hydrogen bonds that link an unpaired base to a group of two already paired bases must be from one to three. It has been decided to limit the minimum number of hydrogen bonds in a base pair (to the number of two) because base pairs with a single hydrogen bond can be classified as a variant of the primary types and because the whole number of hydrogen bonds found in a base triplet is at least three [61].

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 the Supplementary Information, whose Section 1 explains the symbols used in the models and their transliteration while Section 2 the models construction).

3.2.1 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 characterizations 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 basing 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 game is considered won by the defender (because the attacker is unable to distinguish the behaviour of the two processes).

Two states are strongly bisimilar if and only if the defender has a *universal winning strategy*

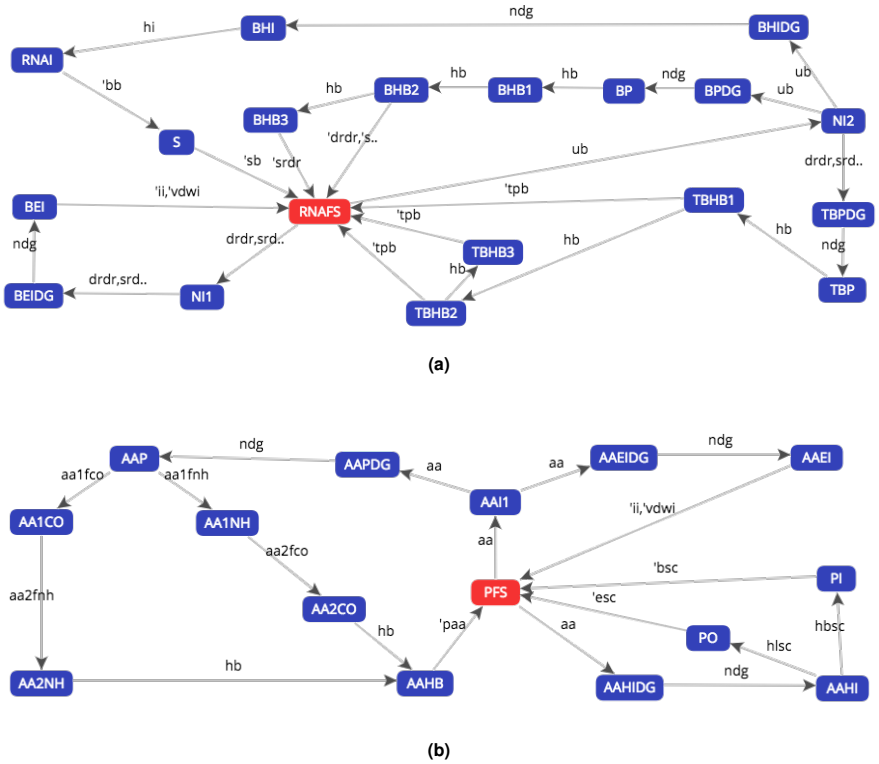


Figure 3.2 – Labelled Transition Systems of **(a)** the $\mathcal{F}^s_{rna}$ process, transliterated RNAFS, and of **(b)** the $\mathcal{F}^s_p$ process, transliterated PFS, generated with the CAAL web-based tool (Concurrency Workbench, Alborg Edition). The symbols are described in Section 1 of the Supplementary Information.

(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 by such states.

If we try to prove the behavioural equivalence of the $\mathcal{F}^s_{rna}$ and $\mathcal{F}^s_p$ processes we can observe, from the LTSs in Figure 3.2, that the bisimulation game ends after only one move, independently of the choice made by the attacker, with the defeat of the defender.

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

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

We might therefore wonder if *there is an abstraction level at which the two folding processes would show a behavioural equivalence*. As it will be proved in this chapter, this level of abstraction can actually be defined. Its construction, however, requires a generalisation of the weak-interaction processes and the imposition of some limitations to the expressiveness of the protein folding process.

3.2.2 Higher abstraction level model

The first of the two aforementioned modifications 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 property of each weak interaction, therefore the model is still faithful to the biological process. However they are not sufficient to obtain a behavioural equivalence between the folding processes of RNAs and proteins.

What we still need to do is limiting 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).

Let $\mathcal{H} : P \rightarrow P$ be the function that maps each folding process to its respective abstraction level, as above defined. The application of $\mathcal{H}$ to the models described in the previous section results in a new representation of the folding processes of RNAs and proteins, indicated by the symbols $\mathscr{F}_{rna}$ and $\mathscr{F}_p$ respectively (see Section 2 of the Supplementary Information for a complete description).

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

Theorem 1. *If $\mathscr{F}_{rna} = \mathcal{H}(\mathcal{F}_{rna})$ and $\mathscr{F}_p = \mathcal{H}(\mathcal{F}_p)$ then $\mathscr{F}_{rna}$ and $\mathscr{F}_p$ are strongly bisimilar ($\mathscr{F}_{rna} \sim \mathscr{F}_p$).*

Proof. The proof is provided via a bisimulation game (see Table 3.1). A winning strategy of the defender starts from the pair of states $(\mathscr{F}^s_{rna}, \mathscr{F}^s_p)$ of the relative LTSs, transliterated (RNAFS,PFS) as in Figure 3.3.

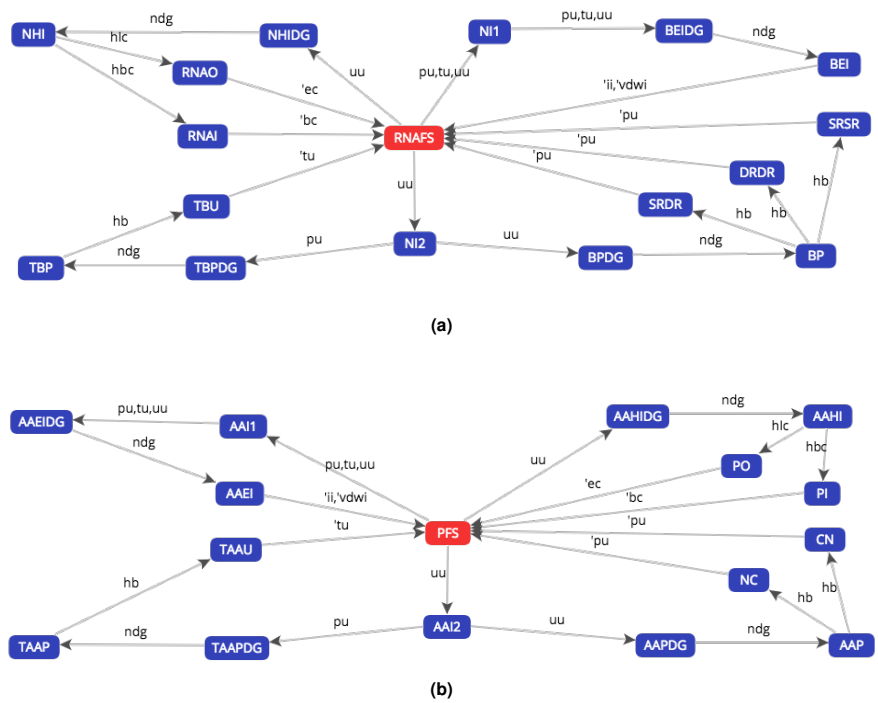


Figure 3.3 – Labelled Transition Systems of **(a)** the redefined $\mathscr{F}^s_{rna}$ process, transliterated RNAFS, and of **(b)** the redefined $\mathscr{F}^s_p$ process, transliterated PFS, generated with the CAAL web-based tool (Concurrency Workbench, Alborg Edition). The symbols are described in Section 1 of the Supplementary Information.

As proved by Milner [58], given two processes P and Q, such that $P \sim Q$, the following two rules are true:

$P|R \sim Q|R$ and $R|P \sim R|Q$, for each process R

$P\backslash L \sim Q\backslash L$, for each set of labels L.

The $\mathscr{F}_{rna}$ and $\mathscr{F}_p$ folding processes, likewise $\mathcal{F}_{rna}$ and $\mathcal{F}_p$, are defined as

$$\mathscr{F}_{rna} \stackrel{\text{def}}{=} (\mathscr{F}^s_{rna}|\Delta G_{\mathscr{F}^s})\backslash\{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};$$

$$\mathscr{F}_p \stackrel{\text{def}}{=} (\mathscr{F}^s_p|\Delta G_{\mathscr{F}^s}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};$$

where $\Delta G_{\mathscr{F}^s} \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta G_{\mathscr{F}^s} + \overline{\mathtt{ndg}}.\Delta G_{\mathscr{F}^s} + \overline{\mathtt{zdg}}.\Delta G_{\mathscr{F}^s}$.

Then they are also strongly bisimilar.

□

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

Table 3.1 – Winning strategy of the defender in the strong bisimulation game that compares the pair of processes $(\mathscr{F}^s_{rna}, \mathscr{F}^s_p)$, transliterated (RNAFS,PFS). The results of this play proves that $RNAFS \sim PFS$, i.e. that the two processes are strongly bisimilar.

In this way 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.

Such proof can also be obtained with the aid of an automated tool; in Figure 3.4 we show the results of the bisimulation game performed with CAAL on the processes $\mathscr{F}_{rna}$ and $\mathscr{F}_p$, transliterated RNAFOLDING and PFOLDING respectively.

Status	Time	Property
✔	150 ms	RNAFOLDING ~ PFOLDING

Figure 3.4 – Bisimulation game performed with the CAAL web-based tool shows that, as the checkmark on the "Status" column indicates, the RNAFOLDING and the PFOLDING processes are strongly bisimilar (relation represented by the symbol ~).

3.3 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 such processes show a behavioural equivalence.

Its existence allows us to hypothesise some of the reasons that led the evolution of life to the formation of the proteins and to take them on, in biological processes, along with RNAs.

We have formally proved how it is possible to reach the behavioural equivalence between the RNA folding and the protein folding by reducing the complexity of the structures expressible, hence the functions they can perform, in the latter process. This demonstration can be interpreted as a clue that, at a point in the early evolution of life on Earth, proteins emerged to answer the necessity of molecules that could carry out more effectively the functions performed by RNA molecules and could also deal with more complex tasks. We are well aware that this demonstration leaves numerous questions open regarding the RNA World theory, such as the function that RNA would play in storing genetic information; it is not in any case the objective of our work to provide a definitive proof of the aforementioned theory. However, we are equally convinced that our work sets a solid foundation for further developments in this direction.

Indeed, thanks to these results, we can observe how it is possible to infer the complexity of a biological structure, and therefore of its function, starting from 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 on the basis of the known properties of the interactions that bind nucleotides (in RNAs) and amino acids (in proteins).

3.4 Conclusions

CCS, due to its expressiveness, turned out to be perfectly suitable to define models based on the application of the aforementioned approach. The use of process algebras to describe molecular interactions can highlight the relation 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 taking into account all the fundamental dynamics characterising a biological system. We are currently involved in defining formal models of the whole gene expression process in order to study the gene mutations which cause protein misfolding [20, 32] and the gene assembly process [23].

Our approach should not be intended as a simulation-based tool, but 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 [52, 54, 73].

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 the extraction and formalization of ideas and their manifold consequences [75] .

An Algebraic Approach to the Study of Protein Misfolding

4.1 Introduction

In this chapter, we use Milner's CCS (Calculus of Communicating Systems) process algebra [58] and the Hennessy-Milner logic [40] to model how genes express the structures of RNAs and proteins and the relation with their folded conformations.

In Chapter 3, we provided the models of RNA and protein folding processes and proved how it is possible to formally define a level of abstraction in which such processes show a behavioural equivalence [50]. This abstraction level can be obtained only by reducing the complexity of the folding process of proteins and hence of the structures it can express; its definition allowed us to hypothesise some of the reasons that lead the evolution of life to the formation of proteins and to take them on as the main catalysts in the biological processes.

We now focus on a class of pathologies that affects the folding processes to study how the differences between the structural components of proteins and RNAs, identified in the aforementioned chapter, cause a dissimilar response to an alteration of the correct folding pathway.

Our study starts from the formal description of how such pathologies originate as an error of the genetic code (a mutation, in biological terms) and can propagate through each step of the gene expression, affecting both the RNA and the protein structure.

Therefore, this approach requires first to define a formal model of the gene expression; this model is specifically focussed on the transformations undergone by the informational content and the possible pathways it can follow (correct or wrong) from the DNA sequence of a gene to the ribonucleotide sequence of an RNA molecule and/or the aminoacidic sequence of a protein.

Starting form this model, we formally describe how the mutation of even a single gene (point

mutation) can alter the final conformation of a protein while, at the same time, it is harmless for the structures of RNAs. We use a well-known pathology affecting haemoglobin, the sickle-cell anaemia, as a case-study to show such properties.

The gene expression has been modelled as the result of sub-processes that proceed along a path made by discrete states; this aspect is highlighted by describing such a process via Labelled Transition Systems (LTSs) [46]. We also represent the gene sequences as Hennessy-Milner formulae that can be satisfied by the gene expression processes. (see Section 2.2 on page 32, for details on the modelling approaches adopted in this chapter).

4.1.1 DNA replication and gene expression models

The first step in the description of the gene expression model is to define the set of nucleotides; they are the elementary units of both DNA and RNA (with some biochemical differences not relevant for the aim of this model) and are identified by the bases they contain.

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

where each letter stands for adenine, thymine, cytosine, guanine and uracil respectively.

DNA should not contain the uracil base, while RNA does not contain the thymine; therefore it is useful to define two subsets of $\mathcal{N}$ as follows:

$$\mathcal{N}_{dna} = \{\mathtt{a}, \mathtt{t}, \mathtt{c}, \mathtt{g}\}$$

$$\mathcal{N}_{rna} = \{\mathtt{a}, \mathtt{u}, \mathtt{c}, \mathtt{g}\}$$

The expression of the DNA sequence of a gene flows through three main processes: **transcription**, **RNA processing** and **translation**.

As shown later in this chapter and in related the Supplementary Information, it is possible to define DNAs and RNAs (and hence genes) as strings of nucleotides while proteins as strings of amino acids.

The three above-mentioned processes can therefore be imagined as functions on strings: the final product of the gene expression will be the result of the composition of these functions.

To outline this idea we can define

- the transcription as the *tsc* function, such that

$$RNA = tsc(gene);$$

- the RNA processing as the *prc* function, such that

$$mRNA = prc(RNA);$$

- the translation as the *tsl* function, such that

$$protein = tsl(mRNA).$$

This means that the overall gene expression process ($GeneExp$) can be defined as:

$$GeneExp(gene) = (tsl \circ prc \circ tsc)(gene) = protein$$

During the transcription process, the sequence of nucleotides taken as template to produce an RNA molecule (transcript) is read from the complementary strand of the actual coding strand (for a specific gene). This is due to the base pairing process, which characterises almost every step of the gene expression (and hence the transcription).

The process in which a strand of DNA is produced using the strand of an already existing molecule as template is called **DNA replication**. A mutation that can affect the expression of a gene often happens during this process, therefore, in order to clearly show how an error in the gene expression can be generated and propagated, an important starting point is to define a model of the DNA replication.

We provide here the model of the DNA replication as an example of the approach we adopt and to highlight the main properties we analyse with our work. The description of the gene expression model can be found in the Supplementary Information.

The replication process is mainly performed by an enzyme called DNA Polymerase (DNAPol); indeed, the biological process also involves other proteins and the dynamic interactions between them. For the aim of our modelling, we focus only on the transformations that this process produces on the genetic information.

The DNA polymerase reads the template strand of DNA and associates to each nucleotide of its sequence a new nucleotide basing on the base pairing complementarity; this process produces a new strand of DNA and hence a new molecule made by the old strand and the newly synthesised one.

The process starts from a *replication origin* (marked by a sequence of nucleotides) and proceeds until it reaches a *replication terminus*.

We can formally define a DNA sequence as a string of elements of the above-defined $\mathcal{N}$ set; therefore the set of all possible DNA sequences is $\mathcal{D}$ defined as follows:

$$\mathscr{D} = \{n_1 n_2 \dots n_k \mid n_i \in \mathscr{N}_{dna},\ i \in \{1, \dots, k\}\},$$

where n stands for *nucleotide*.

The DNA replication is modelled by describing the DNA polymerase as a process (*DNAPOL*) that, starting from the *replication origin*, takes as input the base of a nucleotide (a, t, c or g) and produces as output the association of such a base with its complement (at, ta, cg, gc). The process stops when it reaches the *replication terminus*.

It is important to notice that the *DNAPOL* process does not take the whole string of DNA as input, but works on one nucleotide at time (it is a "function on nucleotides" and not a function on strings).

The correct association of the bases is not the only output that *DNAPOL* can produce; the replication process, indeed, can make mistakes (often called **mispairing**). When such errors occur, a purine (a, g) is associated with the wrong pyrimidine (t, c) - or vice-versa. The output in these cases will be one of the following base pairs (called wobble base pairs'): ac, ca, gt, tg.

If a mispairing remains uncorrected, it will be taken as template in a subsequent replication, becoming in this way a **permanent mutation**.

To try to avoid this possibility the replication process puts in place two mechanisms of error detection and correction: the **proofreading** and the **mismatch repair** processes.

In the model, the *PROOFREAD1* process [1] takes the base pairs produced as output by the *DNAPOL* process and provides the correct nucleotide that has to be added to the new DNA strand. The proofread process is not unerring: it also can make mistakes and leave a mispairing uncorrected. The model describes this possibility too (Figure 4.1).

The mismatch repair takes place after the replication process and is carried out by a complex of proteins that are able to recognise and correct the DNA mismatches. It can also correct two other types of mutations: **depurination** and **deamination**. The depurination consists in the loss of a purine (a or g), giving rise to a lesion that resemble a missing tooth [2]. The deamination is the removal of an amino group from cytosine (c) in DNA to produce the base uracil (u).

The mismatch repair process (*MMREPAIR*) is modelled by defining a subprocess, *MMRPROTEINS*, which read both the strands of the DNA molecule generated by the replication process and produces as output the nucleotide that should be present in the newly synthesised strand due to base complementarity.

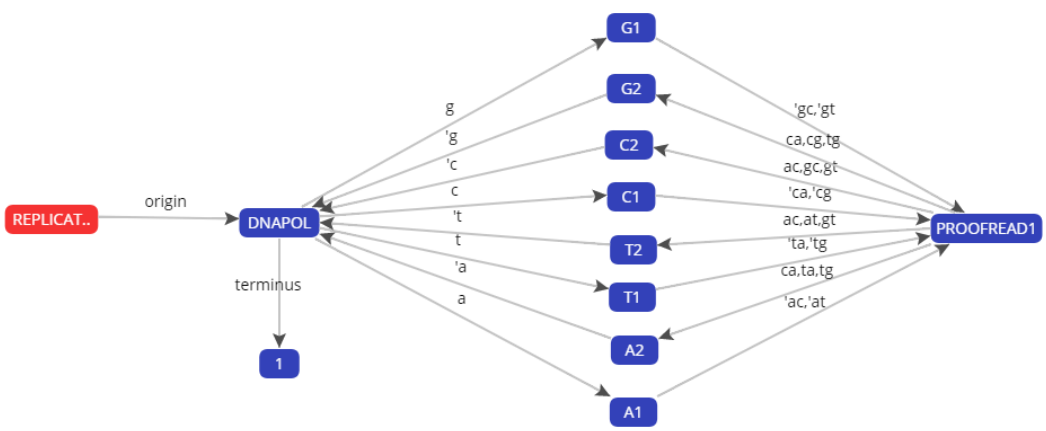
This correction also deal with

[1] The number is used to distinguish this proofread process from the one of the RNA polymerase, described in the Supplementary Information.

$$REPLICATION \stackrel{\text{def}}{=} origin.DNAPOL;$$
$$DNAPOL \stackrel{\text{def}}{=} \mathtt{a}.A1 + \mathtt{t}.T1 + \mathtt{c}.C1 + \mathtt{g}.G1 + terminus.0;$$
$$A1 \stackrel{\text{def}}{=} \overline{\mathtt{at}}.PROOFREAD1 + \overline{\mathtt{ac}}.PROOFREAD1;$$
$$T1 \stackrel{\text{def}}{=} \overline{\mathtt{ta}}.PROOFREAD1 + \overline{\mathtt{tg}}.PROOFREAD1;$$
$$C1 \stackrel{\text{def}}{=} \overline{\mathtt{cg}}.PROOFREAD1 + \overline{\mathtt{ca}}.PROOFREAD1;$$
$$G1 \stackrel{\text{def}}{=} \overline{\mathtt{gc}}.PROOFREAD1 + \overline{\mathtt{gt}}.PROOFREAD1;$$

$$PROOFREAD1 \stackrel{\text{def}}{=}$$
$$\mathtt{at}.T2 + \mathtt{ac}.T2 + \mathtt{ta}.A2 + \mathtt{tg}.A2 + \mathtt{cg}.G2 + \mathtt{ca}.G2 + \mathtt{gc}.C2 + \mathtt{gt}.C2 +$$
$$\mathtt{ac}.C2 + \mathtt{tg}.G2 + \mathtt{ca}.A2 + \mathtt{gt}.T2;$$
$$A2 \stackrel{\text{def}}{=} \overline{\mathtt{a}}.DNAPOL;$$
$$T2 \stackrel{\text{def}}{=} \overline{\mathtt{t}}.DNAPOL;$$
$$C2 \stackrel{\text{def}}{=} \overline{\mathtt{c}}.DNAPOL;$$
$$G2 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.DNAPOL;$$

(a) CCS model of the *REPLICATION* process.



(b) Labelled Transition System of the *REPLICATION* process generated with the CAAL web-based tool.

Figure 4.1 – CCS model of the *REPLICATION* process **(a)** and its related LTS **(b)**.
Given B as one of the possible bases (A, T, C and G), each *B1* state describes the behaviour of the *DNAPOL* process when takes the corresponding nucleotide as input: this behaviour is defined by the nondeterministic choice between the correct and the wobble base pairing; each *B2* state describes which output the proofread process will provide basing on the choice made in the *PROOFREAD1* state; for the sake of clarity, in the first row of the CCS model are indicated the correct choices (including the error corrections), while in the second row the wrong choices (hence the cases in which the proofreading process does not recognise a mispairing).

- depurination, by representing the loss of a base with a x; if the x is "paired" with a `t`, the correct base must be a `a`, if the x is associated with a `c`, the correct output must be a g;
- deamination, by adding a `u` (which stands for uracil) to the possible bases that can be read from both the strands; because a `u` cannot be found in a DNA sequence, such occurrence necessarily identify a deamination, and the right output must be a `c`

In both depurination and deamination cases, if the mutation is found in the template strand, the model shows the right output on the new strand as an implicit description of the correction performed on the original strand.

The mismatch repair can fail in its function, letting a mutation to become permanent. The model describes this case by allowing every mismatch, depurination and deamination as possible outputs (Figure 4.2).

The behaviour of the replication and gene expression processes is described using HML (Hennessy-Milner Logic) formulae, so as to perform the model checking. We can identify in each of their phases a main subprocess, modelling the behaviour of the molecule (or molecules) that carry out the fundamental function of such a phase.

This function is associated with a specific HML formula, satisfied by the related processes. The general formulae that describe the main processes characterising the replication and gene expression models are the following.

Replication

$$DNAPOL \models \langle b1\rangle\langle' b1b2\rangle\langle b1b2\rangle\langle' b2\rangle \mathtt{F} \wedge \langle terminus\rangle \mathbf{tt}$$

where $b1$ and $b2 \in \mathcal{N} - \{\mathtt{u}\}$; $b1$ represents the base of the nucleotide read by the *DNAPOL* process, $'b1b2$ is the base pair provided as output, $b1b2$ is the base pair taken as input by the *PROOFREAD1* process and $'b2$ is the (possibly) correct base that has to be added to the newly synthesised strand.

Mismatch repair

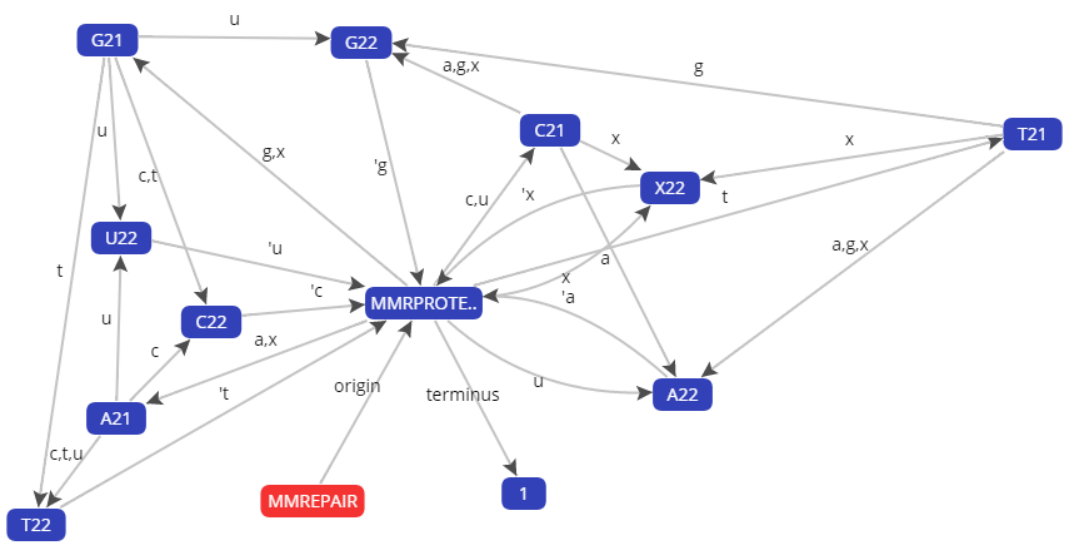
$$MMRPROTEINS \models \langle b1\rangle\langle b2\rangle\langle' b2\rangle\ X \wedge \langle terminus\rangle \mathbf{tt}$$

where $b1$ and $b2 \in \mathcal{N} - \{\mathtt{u}\}$; $b1$ represents the base of the nucleotide read by the *MMRPROTEINS* process on the template strand, $b2$ is the base read on the newly synthesised strand, $'b2$ the (possibly) correct base that should be paired with the first one.

Transcription

$$MMREPAIR \stackrel{\text{def}}{=} origin.MMRPROTEINS;$$
$$MMRPROTEINS \stackrel{\text{def}}{=} \mathtt{a}.A21 + \mathtt{t}.T21 + \mathtt{c}.C21 + \mathtt{g}.G21 +$$
$$x.A21 + x.G21 + x.X22 + u.C21 + u.A22 + terminus.0;$$
$$A21 \stackrel{\text{def}}{=} \mathtt{t}.T22 + \mathtt{c}.T22 + \mathtt{c}.C22 + \mathtt{u}.T22 + \mathtt{u}.U22;$$
$$T21 \stackrel{\text{def}}{=} \mathtt{a}.A22 + \mathtt{g}.A22 + \mathtt{g}.G22 + x.A22 + x.X22;$$
$$C21 \stackrel{\text{def}}{=} \mathtt{g}.G22 + \mathtt{a}.G22 + \mathtt{a}.A22 + x.G22 + x.X22;$$
$$G21 \stackrel{\text{def}}{=} \mathtt{c}.C22 + \mathtt{t}.C22 + \mathtt{t}.T22 + \mathtt{u}.G22 + \mathtt{u}.U22;$$
$$A22 \stackrel{\text{def}}{=} \overline{\mathtt{a}}.MMRPROTEINS;$$
$$T22 \stackrel{\text{def}}{=} \overline{\mathtt{t}}.MMRPROTEINS;$$
$$C22 \stackrel{\text{def}}{=} \overline{\mathtt{c}}.MMRPROTEINS;$$
$$G22 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.MMRPROTEINS;$$
$$U22 \stackrel{\text{def}}{=} \overline{\mathtt{u}}.MMRPROTEINS;$$
$$X22 \stackrel{\text{def}}{=} \overline{x}.MMRPROTEINS;$$

(a) CCS model of the *MMREPAIR* process.



(b) Labelled Transition System of the *MMREPAIR* process generated with the CAAL web-based tool.

Figure 4.2 – CCS model of the *MMREPAIR* process **(a)** and the related LTS **(b)**.
Given B as one of the possible bases (A, T, C and G), each state *B21* describes the behaviour of the *MMPROTEINS* process when it receives the corresponding nucleotide as input (as indicated in the first row) or when chances upon a uracil or a missing purine (the *x* label), indicating that a deamination or a depurination (respectively) has happened; in each of these states is allowed the possibility to produce the correct output, detect and hence correct an error (i.e. a mispairing, a deamination or a depurination) or leave the mutation uncorrected. Each state *B22* defines which output will be produced by the choices made in the above-described processes (a special case are the states *U22* and *X22*, which define the outputs related to deaminations and depurinations respectively).

$$RNAPOL \models \langle b1 \rangle \langle b2 \rangle \langle' b2 \rangle\ X \wedge \langle terminator \rangle \mathbf{tt}$$

where $b1 \in \mathcal{N} - \{\mathtt{t}\}$ and $b2 \in \mathcal{N} - \{\mathtt{u}\}$; $b1$ represents the base of the nucleotide read by the *RNAPOL* process, followed by $'b1b2$ the base pair provided as output, which in turn is taken as input, as $b1b2$, by the *PROOFREAD2* process, which finally provides $'b2$ as the (possibly) correct base that has to be added to the RNA sequence. The *terminator* indicates the end of a gene.

RNA Processing

$$SPLICING \models \mathtt{F}$$

$$\mathtt{F} \equiv \langle b \rangle \langle' b \rangle\, \mathtt{F} \wedge \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \mathtt{X} \wedge \langle tpend \rangle \mathbf{tt}$$

$$\mathtt{X} \equiv \langle b \rangle \mathtt{X} \wedge \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \mathtt{F}$$

where $b \in \mathcal{N} - \{\mathtt{u}\}$ represents the base of the nucleotide read by the *SPLICING* process, followed by the same base produced as output; this is repeated until a gu sequence signals that the beginning of an intron is found. Each base read in this phase (represented by the sub-formula X and performed by the *CUT* process) is not produced as output. The *CUT* process continues until it reaches the ag sequence that signals the end of the intron; after that, the main formula F describes again the behaviour of the process. The label *tpend* indicates the $3'$ end of the RNA sequence.

Translation

$$RIBOSOME \models \mathtt{F}$$

$$\mathtt{F} \equiv \langle b \rangle \mathtt{F} \wedge \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{met} \rangle \mathtt{X}$$

$$\mathtt{X} \equiv \langle b1 \rangle \langle b2 \rangle \langle b3 \rangle \langle' aa \rangle \mathtt{X} \wedge \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle' stop \rangle \mathbf{tt}$$

where $b \in \mathcal{N} - \{\mathtt{u}\}$ represents the base of each nucleotide read before the *RIBOSOME* process reaches the aug codon; after that, the formula X describes how the process translates each codon it encounters until it reaches a stop codon (represented by the formula $\langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle' stop \rangle \mathbf{tt}$). The label $b1$, $b2$ and $b3$ represent the three bases of a codon, while aa is an amino acid belonging to the set $\mathscr{A}$:

$$\mathscr{A} = \{\mathtt{Ala}, \mathtt{Arg}, \mathtt{Asp}, \mathtt{Asn}, \mathtt{Cys}, \mathtt{Glu}, \mathtt{Gln}, \mathtt{Gly}, \mathtt{His}, \mathtt{Ile}, \mathtt{Leu}, \mathtt{Lys}, \mathtt{Met}, \mathtt{Phe}, \mathtt{Pro}, \mathtt{Ser}, \mathtt{Thr}, \mathtt{Trp}, \mathtt{Tyr}, \mathtt{Val}\}$$

4.1.2 Formal description of HBB gene replication and expression

The subunits of haemoglobin are arranged in symmetric pairs, each pair having one α and one β subunit [48].

Basing on the provided models, we describe how the gene that code for one of the β subunits of the haemoglobin molecule (**HBB**) is expressed through each phase detailed in the previous section and in the Supplementary Information (replication, transcription, processing and translation).

$REPLICATION \models$
$\langle origin \rangle \langle \mathtt{c} \rangle \langle' \mathtt{cg} \rangle \langle \mathtt{cg} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{cg} \rangle \langle \mathtt{cg} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{t} \rangle$
$\langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{cg} \rangle \langle \mathtt{cg} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{at} \rangle \langle \mathtt{at} \rangle \langle' \mathtt{t} \rangle$

$\vdots$

$\langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \ \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{at} \rangle \langle \mathtt{at} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \ \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle$

$\vdots$

$\langle \mathtt{a} \rangle \langle' \mathtt{at} \rangle \langle \mathtt{at} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{at} \rangle \langle \mathtt{at} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle$
$\langle' \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{at} \rangle \langle \mathtt{at} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{ta} \rangle \langle \mathtt{ta} \rangle \langle' \mathtt{a} \rangle \langle terminus \rangle \mathbf{tt};$

(a) Each formula of the type $\langle b1 \rangle \langle' b1 b2 \rangle \langle b1 b2 \rangle \langle' b2 \rangle$ represents the base read by the *DNAPOL* process, followed by the base pair provided as output, which in turn is taken as input by the *PROOFREAD1* process, which finally provides the (possibly) correct base that has to be added to the newly synthesised strand.

$MMREPAIR \models$
$\langle origin \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{c} \rangle$
$\langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle$

$\vdots$

$\langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \ \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{g} \rangle \ \langle \mathtt{g} \rangle \langle' \mathtt{gc} \rangle \langle \mathtt{gc} \rangle \langle' \mathtt{c} \rangle$

$\vdots$

$\langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{c} \rangle \langle \mathtt{t} \rangle$
$\langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{t} \rangle \langle' \mathtt{t} \rangle \langle \mathtt{t} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{a} \rangle \langle terminus \rangle \mathbf{tt};$

(b) Each formula of the type $\langle b1 \rangle \langle b2 \rangle \langle' b2 \rangle$ represents the base read by the MMRPROTEINS process on the template strand, the base read on the newly synthesised strand, followed by the (possibly) correct base that should be paired with the first one.

Figure 4.3 – HML formulae of the *REPLICATION* **(a)** and *MMREPAIR* **(b)** processes. The HML formulae describing the behaviour of each step can be very long, therefore they are represented only as their beginning part (one or two rows), their middle part, where the codon of the Glu6 is present, and their ending rows.

The DNA sequence of the HBB gene (1742 bp long) has been retrieved from the NCBI (National Center for Biotechnology Information) site [63] ; the gene contains three exons (coloured in green in their coding regions) and two introns (coloured in blue). In red is highlighted the codon that codes for the Glu 6 of the amino acid sequence produced by the HBB gene.

As for the CCS models, a complete description of the gene expression model for the HBB is provided in the Supplementary Information. In Figure 4.3, we describe the application of our approach to the replication and mismatch repair processes.

4.2 Results

Starting from the description of the correct behaviour that the gene expression of the HBB should show, it is possible to describe how a point mutation can go through each step of the gene expression easily avoiding each error detection.

We propose here a model for the Glu6Val mutation, which cause the sickle-cell anaemia (haemoglobin S with this mutation is referred to as HbS). In the HbS disease, a single nucleotide change (mutation) in the β-globin gene produces a β-globin subunit that differs from normal β-globin only by a change from glutamic acid to valine.

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

Indeed, the mutation of Glu 6 in the β chain to Val creates an hydrophobic patch on the surface of 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 [2].

The formulae that describe the behaviour of the Glu6Val expression are extracted from the formulae of the whole step of the gene expression they belong to.

The following is the part of the gene of the mutant HBB, containing the first exon (coloured in green) and the first intron (coloured in blue), on which the subsequent description is based:

```
 cggctgtcatcacttagacctcaccctgtggagccacaccctagggttggccaatctactcccaggagcaggg
agggcaggagccagggctgggcataaaagtcagggcagagccatctattgcttacatttgcttctgacacaact
gtgttcactagcaacctcaaacagacaccatggtgcatctgactcctgtggagaagtctgccgttactgccctg
tggggcaaggtgaacgtggatgaagttggtggtgaggccctgggcaggttggtatcaaggttacaagacaggtt
taaggagaccaatagaaactgggcatgtggagacagagaagactcttgggtttctgataggcactgactctctc
tgcctattggtctattttcccacccttag
```

The mutated nucleotide (from a to t) is underlined in the above sequence and in the following formulae. They represent sub-formulae of the whole HBB gene formula containing the nucleotide sequences of interest for our analysis of the Glu6Val mutation.

Replication

$DNAPOL \models$
$\langle g\rangle\langle' gc\rangle\langle gc\rangle\langle' c\rangle\underline{\langle t\rangle\langle' ta\rangle\langle ta\rangle\langle' a\rangle}\langle g\rangle\langle' gc\rangle\langle gc\rangle\langle' c\rangle$**tt**

the thymine (t) is converted in adenine (a) in the newly synthesised strand (by the DNA polymerase).

Mismatch repair

$MMRPROTEINS \models$
$\langle g\rangle\langle c\rangle\langle' c\rangle\underline{\langle t\rangle\langle a\rangle\langle' a\rangle}\langle g\rangle\langle c\rangle\langle' c\rangle$**tt**

For the proteins that perform the mismatch repair the base pairing between a thymine (t) and an adenine (a) is correct. Therefore the following is the sequence produced in the complementary strand:

gccgacagtagtgaatctggagtgggacacctcggtgtgggatcccaaccggttagatgagggtcctcgtccc
tcccgtcctcggtcccgacccgtattttcagtcccgtctcggtagataacgaatgtaaacgaagactgtgttga
cacaagtgatcgttggagtttgtctgtggtaccacgtagactgaggac<u>a</u>cctcttcagacggcaatgacgggac
accccgttccacttgcacctacttcaaccaccactccgggacccgtccaaccatagttccaatgttctgtccaa
attcctctggttatctttgacccgtacacctctgtctcttctgagaacccaaagactatccgtgactgagagag
acggataaccagataaaagggtgggaatc

Transcription

$RNAPOL \models$
$\langle c\rangle\langle' cg\rangle\langle cg\rangle\langle' g\rangle\underline{\langle a\rangle\langle' au\rangle\langle au\rangle\langle' u\rangle}\langle c\rangle\langle' cg\rangle\langle cg\rangle\langle' g\rangle$**tt**

The adenine (a) of the DNA strand is converted in uracil (u) in the RNA strand by the RNA polymerase. The sequence of the RNA transcript is the following:

cggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccaggagcaggg
agggcaggagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugcuucugacacaacu
guguucacuagcaaccucaaacagacaccauggugcaucugacuccu<u>gu</u>ggagaagucugccguuacugcccug
uggggcaaggugaacguggaugaaguugguggugaggcccugggcagguugguaucaagguuacaagacagguu
uaaggagaccaauagaaacugggcauguggagacagagaagacucuuggguuucugauaggcacugacucucuc
ugccuauuggucuauuuucccacccuuag

Processing

SPLICING ⊨
⟨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**

The mutated nucleotide (u in this phase) is in an exon, therefore it is not removed from the RNA sequence during the splicing process. The resulting sequence is:

cggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccaggagcaggg
agggcaggagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugcuucugacacaacu
guguucacuagcaaccucaaacagacaccauggugcaucugacuccuguggagaagucugccguuacugcccug
uggggcaaggugaacguggaugaaguuggugugaggcccugggcag

Translation

RIBOSOME ⊨
⟨g⟩⟨a⟩⟨c⟩⟨a⟩⟨c⟩⟨c⟩
⟨a⟩⟨u⟩⟨g⟩⟨′met⟩ ⟨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⟩ **tt**

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

The amino acid sequence coded by the portion of the mutated HBB gene analysed in this section is the following:

MetValHisLeuThrProValGluLysSerAlaValThrAlaLeu
TrpGlyLysValAsnValAspGluValGlyGlyGluAlaLeuGly

(The first Methionine - Met - is removed in the mature protein).

The verification that al the above-described formulae are satisfied by the related process has been made with the aid of the model checking function of the web-based tool CAAL. The results are

Status	Time	Property	Verify
✓	150 ms	DNAPOL ⊨ <g><'gc><gc><'c> <t><'ta><ta><'a> <g><'gc><gc><'c>tt	▶
✓	150 ms	MMRPROTEINS ⊨ <g><c><'c> <t><a><'a> <g><c><'c>tt	▶
✓	150 ms	RNAPOL ⊨ <c><'cg><cg><'g> <a><'au><au><'u> <c><'cg><cg><'g>tt	▶
✓	225 ms	SPLICING ⊨ <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><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><c><u><g><a><u><u><u><g><g><u><c><u><a><u><u><u><u><c><c><c><c><a><c><c><c><u><g><g><u><a><c><u><g><a><u><a><g><g><u><u><u><a><g>tt

RIBOSOME ⊨ <g><a><c><a><c><c> <a><u><g><'met> <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>tt | ▶ |

Figure 4.4 – Verification performed by the CAAL web-based tool of the HML formulae related to the Glu6Val expression. The checkmarks on the "Status" column indicate that all the formulae are satisfied.

shown in Figure 4.4, and prove that the provided models of the replication and gene expression can satisfy not only the formulae of the production of the correct HBB molecule (as extensively shown in the Supplementary Information), but also the formulae of the Glu6Val expression.

4.3 Discussion

In the Glu6Val model description is possible to observe how such a point mutation affects the folding process in relation to the hydrophobic interactions.

To better understand this aspect we can rewrite the model of the *TRANSLATION* process to focus on the type of side chain that characterises each amino acid. For the aim of this modelling approach, we can distinguish two classes of side chains: hydrophobic (`hbsc`) and hydrophilic (`hlsc`).

The CCS specification of the last state of the *TRANSLATION* process becomes as shown in Figure 4.5.

The HML formulae that describe the behaviour of this process therefore are:

- For the normal HBB gene:

  $\mathtt{F}_n \equiv$

$$TRANSLATION \stackrel{\text{def}}{=} cap.RIBOSOME;$$

$$RIBOSOME \stackrel{\text{def}}{=} u.RIBOSOME + c.RIBOSOME + a.STARTCODON1 + g.RIBOSOME;$$
$$STARTCODON1 \stackrel{\text{def}}{=} u.STARTCODON2 + c.RIBOSOME + a.RIBOSOME + g.RIBOSOME;$$
$$STARTCODON2 \stackrel{\text{def}}{=} u.RIBOSOME + c.RIBOSOME + a.RIBOSOME + g.START;$$

$$START \stackrel{\text{def}}{=} \overline{met}.DECODE;$$

$$DECODE \stackrel{\text{def}}{=}$$
$$\mathsf{u}.(\mathsf{u}.(\mathsf{u}.PHE + \mathsf{c}.PHE + \mathsf{a}.LEU + \mathsf{g}.LEU) + \mathsf{c}.(\mathsf{u}.SER + \mathsf{c}.SER + \mathsf{a}.SER + \mathsf{g}.SER) +$$
$$\mathsf{a}.(\mathsf{u}.TYR + \mathsf{c}.TYR + \mathsf{a}.STP + \mathsf{g}.STP) + \mathsf{g}.(\mathsf{u}.CYS + \mathsf{c}.CYS + \mathsf{a}.STP + \mathsf{g}.TRP)) +$$
$$\mathsf{c}.(\mathsf{u}.(\mathsf{u}.LEU + \mathsf{c}.LEU + \mathsf{a}.LEU + \mathsf{g}.LEU) + \mathsf{c}.(\mathsf{u}.PRO + \mathsf{c}.PRO + \mathsf{a}.PRO + \mathsf{g}.PRO) +$$
$$\mathsf{a}.(\mathsf{u}.HIS + \mathsf{c}.HIS + \mathsf{a}.GLN + \mathsf{g}.GLN) + \mathsf{g}.(\mathsf{u}.ARG + \mathsf{c}.ARG + \mathsf{a}.ARG + \mathsf{g}.ARG)) +$$
$$\mathsf{a}.(\mathsf{u}.(\mathsf{u}.ILE + \mathsf{c}.ILE + \mathsf{a}.ILE + \mathsf{g}.MET) + \mathsf{c}.(\mathsf{u}.THR + \mathsf{c}.THR + \mathsf{a}.THR + \mathsf{g}.THR) +$$
$$\mathsf{a}.(\mathsf{u}.ASN + \mathsf{c}.ASN + \mathsf{a}.LYS + \mathsf{g}.LYS) + \mathsf{g}.(\mathsf{u}.SER + \mathsf{c}.SER + \mathsf{a}.ARG + \mathsf{g}.ARG)) +$$
$$\mathsf{g}.(\mathsf{u}.(\mathsf{u}.VAL + \mathsf{c}.VAL + \mathsf{a}.VAL + \mathsf{g}.VAL) + \mathsf{c}.(\mathsf{u}.ALA + \mathsf{c}.ALA + \mathsf{a}.ALA + \mathsf{g}.ALA) +$$
$$\mathsf{a}.(\mathsf{u}.ASP + \mathsf{c}.ASP + \mathsf{a}.GLU + \mathsf{g}.GLU) + \mathsf{g}.(\mathsf{u}.GLY + \mathsf{c}.GLY + \mathsf{a}.GLY + \mathsf{g}.GLY)) +$$
$$polyatail.0;$$

$$\begin{array}{lcl}
ALA \stackrel{\text{def}}{=} \overline{\mathtt{ala}}.DECODE; & \rightarrow & ALA \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
GLY \stackrel{\text{def}}{=} \overline{\mathtt{gly}}.DECODE; & \rightarrow & GLY \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
VAL \stackrel{\text{def}}{=} \overline{\mathtt{val}}.DECODE; & \rightarrow & VAL \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
LEU \stackrel{\text{def}}{=} \overline{\mathtt{leu}}.DECODE; & \rightarrow & LEU \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
ILE \stackrel{\text{def}}{=} \overline{\mathtt{ile}}.DECODE; & \rightarrow & ILE \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
PRO \stackrel{\text{def}}{=} \overline{\mathtt{pro}}.DECODE; & \rightarrow & PRO \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
PHE \stackrel{\text{def}}{=} \overline{\mathtt{phe}}.DECODE; & \rightarrow & PHE \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
MET \stackrel{\text{def}}{=} \overline{\mathtt{met}}.DECODE; & \rightarrow & MET \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
TRP \stackrel{\text{def}}{=} \overline{\mathtt{trp}}.DECODE; & \rightarrow & TRP \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
CYS \stackrel{\text{def}}{=} \overline{\mathtt{cys}}.DECODE; & \rightarrow & CYS \stackrel{\text{def}}{=} \overline{hbsc}.DECODE; \\
 & & \\
ARG \stackrel{\text{def}}{=} \overline{\mathtt{arg}}.DECODE; & \rightarrow & ARG \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
ASP \stackrel{\text{def}}{=} \overline{\mathtt{asp}}.DECODE; & \rightarrow & ASP \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
ASN \stackrel{\text{def}}{=} \overline{\mathtt{asn}}.DECODE; & \rightarrow & ASN \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
GLU \stackrel{\text{def}}{=} \overline{\mathtt{glu}}.DECODE; & \rightarrow & GLU \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
GLN \stackrel{\text{def}}{=} \overline{\mathtt{gln}}.DECODE; & \rightarrow & GLN \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
HIS \stackrel{\text{def}}{=} \overline{\mathtt{his}}.DECODE; & \rightarrow & HIS \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
LYS \stackrel{\text{def}}{=} \overline{\mathtt{lys}}.DECODE; & \rightarrow & LYS \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
SER \stackrel{\text{def}}{=} \overline{\mathtt{ser}}.DECODE; & \rightarrow & SER \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
THR \stackrel{\text{def}}{=} \overline{\mathtt{thr}}.DECODE; & \rightarrow & THR \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
TYR \stackrel{\text{def}}{=} \overline{\mathtt{tyr}}.DECODE; & \rightarrow & TYR \stackrel{\text{def}}{=} \overline{hlsc}.DECODE; \\
 & & \\
STP \stackrel{\text{def}}{=} \overline{stop}.0; & &
\end{array}$$

Figure 4.5 – Changes applied to the last part of the *TRANSLATION* process to highlight the hydrophobic or hydrophilic property of the side chain of each amino acid. For a complete description of the model of the translation process see the Supplementary Information.

Status	Time	Property	Verify
✓	101 ms	RIBOSOME ⊨ <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	▶
✓	101 ms	RIBOSOME ⊨ <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 4.6 – Verification performed by the CAAL web-based tool of the HML formula related to the expression of a portion of the sequence of the correct HBB gene (first row) and the GL6Val mutation(second row), described in terms of the hydrophobic properties of its amino acids. The checkmarks on the "Status" column indicate that the formulae are satisfied. The red squares highlight the difference between the normal codon and the mutated one.

$$\langle a\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle g\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle c\rangle\langle a\rangle\langle u\rangle\langle'hlsc\rangle\ \langle c\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle a\rangle\langle c\rangle\langle u\rangle\langle'hlsc\rangle\ \langle c\rangle\langle c\rangle\langle u\rangle\langle'hbsc\rangle$$
$$\langle g\rangle\langle a\rangle\langle g\rangle\langle'hlsc\rangle\ \langle g\rangle\langle a\rangle\langle g\rangle\langle'hlsc\rangle\ \langle a\rangle\langle a\rangle\langle g\rangle\langle'hlsc\rangle\ \mathbf{tt}$$

- For the Glu6Val mutated HBB gene:

  $F_m \equiv$
  $$\langle a\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle g\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle c\rangle\langle a\rangle\langle u\rangle\langle'hlsc\rangle\ \langle c\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle a\rangle\langle c\rangle\langle u\rangle\langle'hlsc\rangle\ \langle c\rangle\langle c\rangle\langle u\rangle\langle'hbsc\rangle$$
  $$\langle g\rangle\langle u\rangle\langle g\rangle\langle'hbsc\rangle\ \langle g\rangle\langle a\rangle\langle g\rangle\langle'hlsc\rangle\ \langle a\rangle\langle a\rangle\langle g\rangle\langle'hlsc\rangle\ \mathbf{tt}$$

They are both satisfied by the RIBOSOME process, as shown in Figure 4.6.

Therefore the portion of the aminoacidic sequence of the β-subunit of the haemoglobin molecule can be written in terms of the hydrophobic properties of each amino acid:

hbsc hbsc hlsc hbsc hlsc hbsc hlsc hlsc hlsc

for the normal HBB;

hbsc hbsc hlsc hbsc hlsc hbsc hbsc hlsc hlsc

in the case of the Glu6Val mutation.

Using the model of the protein folding described in the in Chapter 3, it is possible to show how the expression of a gene can affect the conformation of a protein also by defining the proper position of an hydrophobic or an hydrophilic amino acid.

First it is useful to summarise how the hydrophobic interactions has been described in the protein folding model:

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

$$
\begin{array}{lcl}
 & \ldots & \\
\mathcal{I}^e_{aa} & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^s_p + \overline{vdwi}.\mathcal{F}^s_p; \\
\mathcal{I}^h_{aa} & \stackrel{\text{def}}{=} & \mathtt{hlsc}.\mathtt{O_p} + \mathtt{hbsc}.\mathtt{I_p}; \\
\mathtt{O_p} & \stackrel{\text{def}}{=} & \overline{\mathtt{esc}}.\mathcal{F}^s_p; \\
\mathtt{I_p} & \stackrel{\text{def}}{=} & \overline{\mathtt{bsc}}.\mathcal{F}^s_p
\end{array}
$$

where aa indicates an amino acid molecule, ndg represents the negative ΔG value of the process, hlsc and hbsc stand for hydrophilic and hydrophobic side chain respectively, 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 (a complete description of the protein folding model can be found in the aforementioned chapter).

Now we can write a HML formula which describes the behaviour of the $\mathcal{F}^s_p$ (protein folding step) process when receives as input the redefined amino acid sequence (described above):

$\mathtt{F}_p \equiv$
$\langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle$
$\langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle$
$\langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \mathbf{tt}$

$\wedge$

$\langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle$
$\langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle \langle' \mathtt{bsc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle$
$\langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hlsc} \rangle \langle' \mathtt{esc} \rangle \mathbf{tt}$

The formula, representing both the correct and the mutated sequences, is satisfied by the $\mathcal{F}^s_p$ process, as shown in Figure 4.7 (where it is transliterated as *PFS* process).

Because the hydrophobic properties of each amino acid determine its position in the inside or on the outside of the protein, the alteration of a single nucleotide (i.e. a point mutation) can cause a missense mutation which leads to a modified positioning of the affected amino acid in space and hence an alteration of the three-dimensional structure of the protein.

This is the case of the sickle-cell anaemia, in which the hydrophobic valine replaces the hydrophilic glutamic acid in the same position, forming an hydrophobic patch. The necessity

Status	Time	Property	Verify
✓	101 ms	PFS ⊨ <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><hlsc><'esc><aa><ndg><hlsc><'esc><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>tt	▶
✓	100 ms	RNAFS ⊨ <ub><ub><ndg><hbi><'bb>tt	▶

Figure 4.7 – Verification performed with the CAAL of the HML formulae associated to the folding process of the hemoglobin β-subunit (first row) and of the related mRNA (second row); they are described in terms of the hydrophobic interactions performed by amino acids and nucleotides respectively. The checkmarks on the "Status" column indicate that the formulae are satisfied.

of an hydrophobic amino acid to be shielded from water (buried), pushes the valine to bind into the hydrophobic pocket of another haemoglobin molecule, forming in this way the fibrous precipitates which characterise the sickle-cell disease.

In contrast, in the folding of the mRNA of the HBB we can observe a different behaviour, because each nucleotide interact in the same way with water.

Summarising the CCS specification of the $\mathcal{I}_b^h$ (base hydrophobic interaction) process of the RNA folding model (defined in Chapter 3), it is possible to note that each unpaired base is always buried and stacked parallel to another one.

$$
\begin{array}{lcl}
\mathcal{F}^s_{rna} & \stackrel{\text{def}}{=} & \mathtt{ub}.\mathcal{I}1_n + \mathtt{ub}.\mathcal{I}2_n + \mathtt{srsr}.\mathcal{I}1_n + \\
 & & \mathtt{drdr}.\mathcal{I}1_n + \mathtt{srdr}.\mathcal{I}1_n + \mathtt{tpb}.\mathcal{I}1_n; \\
\mathcal{I}1_n & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta G_{\mathcal{I}^e_b} + \mathtt{srsr}.\Delta G_{\mathcal{I}^e_b} + \mathtt{drdr}.\Delta G_{\mathcal{I}^e_b} + \\
 & & \mathtt{srdr}.\Delta G_{\mathcal{I}^e_b} + \mathtt{tpb}.\Delta G_{\mathcal{I}^e_b}; \\
\mathcal{I}2_n & \stackrel{\text{def}}{=} & \mathtt{ub}.\Delta G_{\mathcal{P}_{b2}} + \mathtt{ub}.\Delta G_{\mathcal{I}^h_b} + \mathtt{srsr}.\Delta G_{\mathcal{P}_{b3}} + \\
 & & \mathtt{drdr}.\Delta G_{\mathcal{P}_{b3}} + \mathtt{srdr}.\Delta G_{\mathcal{P}_{b3}}; \\
\Delta G_{\mathcal{I}^e_b} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^e_b; \\
\Delta G_{\mathcal{I}^h_b} & \stackrel{\text{def}}{=} & \mathtt{ndg}.\mathcal{I}^h_b; \\
 & & \\
 & \ldots & \\
 & & \\
\mathcal{I}^e_b & \stackrel{\text{def}}{=} & \overline{ii}.\mathcal{F}^s_{rna} + \overline{vdwi}.\mathcal{F}^s_{rna}; \\
\mathcal{I}^h_b & \stackrel{\text{def}}{=} & hbi.\mathtt{I}_{rna}; \\
\mathtt{I}_{rna} & \stackrel{\text{def}}{=} & \overline{\mathtt{bb}}.\mathcal{S}; \\
\mathcal{S} & \stackrel{\text{def}}{=} & \overline{\mathtt{sb}}.\mathcal{F}^s_{rna}
\end{array}
$$

where 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 with another one.

Therefore, the formula that describes the behaviour of the hydrophobic interaction in RNA folding is the following

$$\mathtt{F}_{rna} \equiv \langle \mathtt{ub} \rangle \langle \mathtt{ub} \rangle \langle \mathtt{ndg} \rangle \langle hbi \rangle \langle' \mathtt{bb} \rangle$$

for every couple of unpaired bases and *it is satisfied by the* $\mathcal{F}^s_{rna}$ process (as shown in Figure 4.7, where it is transliterated as *RNAFS* process).

This means that, form the point of view of the hydrophobic interactions, a point mutation can't affect significantly the folding of the RNA.

4.4 Conclusions

In this chapter a pathology which affect the folding process has been treated as emerging behaviour of the component elements of proteins (amino acids) and RNAs (nucleotides).

With the aid of formal models, we have shown that even the mutation of a single nucleotide of the gene that codes for a protein can alter the pathway of the protein folding process.

Our study has been focussed on the hydrophobic interactions, which are critical in proteins because each amino acid is specifically hydrophobic or hydrophilic.

This property is not equally true for RNAs, in which the nucleotides interact almost in the same way with water; even if different base stacking can have dissimilar energetic values, this difference can't affect significantly the folding of the RNA.

In Chapter 3, we proved that the structures of proteins and the distinctive interactions between their component units (the amino acids) allow them to fold up in three-dimensional conformations more complex if compared to the folding of RNAs.

Due to the tight relation between structure and function which exists in biological systems, these three-dimensional conformations allow proteins to perform more effectively the functions carried out by RNAs and also to deal with more complex tasks.

However, as proved with the models proposed in this chapter, the greater complexity of proteins has the drawback to expose them to some pathologies which instead do not affect the simpler structure of RNAs.

Further studies in this direction will involve the analysis of other pathologies associated to protein misfolding [32], in particular the ones responsible for ageing-related disease (like Alzheimer and Parkinson [27]). To model the folding processes related to such diseases, we will complement the formal approaches described in this chapter with other algebraic and computational methods, like topological data analysis and graph grammars [52, 54, 73, 75].

Algebraic Characterisation of Non-coding RNA

5.1 Introduction

The relation between structures and functions is a relevant topic in biology, whose investigation received a significant contribution by different computational approaches, from process calculi to topological data analysis [9, 15, 51, 54, 68].

In particular, formal languages and graph grammars have been successfully applied in modelling the properties that correlate the functions expressible by RNA molecules and specific substructures involved in their folding - the process that allows a linear biopolymer to reach a three-dimensional conformation by forming hydrogen bonds between non-consecutive monomers [52, 73].

In Chapter 3, 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 the others in the same sequence. This result has been obtained by comparing the RNA folding with that performed by proteins, in order to identify an abstraction level at which these two classes of molecules show the same structural and functional complexity [50]. We refer to this level as *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 the present chapter, we want to hypothesise the functions that characterise the *congruence level* introduced and further explore the applicability of process algebras in modelling the related biological processes.

The resulting models will form the basis of a multiagent simulation [43]. In an agent-based simulation, agents are discrete software elements whose interactions correspond to those performed by the components of the modelled system, quite faithfully to the actual behaviour of a

biological process [56]. 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 multiagent systems.

However, biological processes are complex systems whose emerging behaviour is not always possible to predict, due to the incompleteness of observed data.To incorporate this property in an agent-based model of a biological system, agents' interactions should have an aleatory nature or the simulation environment should be non-predictable (this implies that each run of the simulation is affected by statistical uncertainty). For that reason, a further step is needed to provide an effective specification of the environment, hopefully by referring to interactive computation modelling [55].

The multiagent simulator referred in this work is developed to study the molecular interactions characterising metabolic pathways, and analyse the emergence of global properties from local interactions [13, 69]. We simulated a complete enzymatic reaction by modelling the molecules involved (enzymes, metabolites and complexes) as autonomous and interactive agents. We will extensively discuss this agent-based method for simulating bimolecular interactions in the Part II of this manuscript. For this reason, the present chapter can be considered a bridge between the two modelling approaches exploited in this work.

The RNA models we propose in this chapter are algebraic specifications of new functionalities that will enrich the simulator. We expect that, similarly to the results we obtained regarding metabolic reactions [69], analysing the behaviour emerging from agents' interactions will yield additional information on the biological properties of RNAs.

5.2 Results

At the abstraction level we are exploring, the behavioural equivalence between RNA and protein has been reached by reducing the complexity of the protein folding (limiting the number of amino acids that can interact through hydrogen bonds). This limitation also reduces the complexity of the structures, and hence of the functions, that can be expressed by the folding process. Therefore, 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 Section 1 characterises the congruence relation that defines the ncRNA congruence class, whose complete formalisation will be provided in a future work.

In this work, 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 [44, 78].

5.2.1 Ligand Binding Function

Ribozymes can bind, through specific binding-sites, 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 [24, 91].

In our models, the ligand binding function consists in gaining a ligand, through a binding site of the RNA molecule, in order 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, the binding of the ligand is not possible.

The model of this functional role is provided by the *Ligand Binding* process ($\mathscr{B}$), 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 (bs_*) as output; otherwise the binding site remains 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 in order to store the bound ligand;
2. the ligand is released;
3. a second function is activated or interrupted.

Basing on the above description, we provide the following CCS specification of the process which allows checking if a ligand can be stored, producing as output an occupied binding site

(bs_*):

$$
\begin{aligned}
RNA &\stackrel{\text{def}}{=} bs.\mathscr{B};\\
\mathscr{B} &\stackrel{\text{def}}{=} l.(SC_{\vee} + EC_{\vee});\\
SC_{\vee} &\stackrel{\text{def}}{=} sc.SC + sn.SN;\\
SC &\stackrel{\text{def}}{=} ec.BS_* + en.EN;\\
EC_{\vee} &\stackrel{\text{def}}{=} ec.EC + en.EN;\\
EC &\stackrel{\text{def}}{=} sc.BS_* + sn.SN;\\
SN &\stackrel{\text{def}}{=} \overline{bs}.\mathscr{B};\\
EN &\stackrel{\text{def}}{=} \overline{bs}.\mathscr{B};\\
BS_* &\stackrel{\text{def}}{=} \overline{bs_*}.0
\end{aligned}
\tag{5.1}
$$

The ncRNA is represented here and in the following specifications and formulas as the general process RNA. For a complete explanation of the symbols used in our models, refer to Table 5.1.

5.2.2 Enzymatic Function

Ribozymes perform a variety of enzymatic activities in cells, for which several analogies have been found with those carried out by proteins [19].

Since the present work is intended to outline a model of the functions characterising the *congruence level* that relates RNAs and proteins [50], we can generalise the enzymatic activity of ribozymes as the catalysis of a reaction.

Formalising this process requires first to provide 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* ($\mathscr{R}_{fd}$): starting from a substrate, generates one or more products;
- *Backward Reaction Direction* ($\mathscr{R}_{bd}$): starting from the products, generate the original substrate.

The choice between $\mathscr{R}_{fd}$ and $\mathscr{R}_{bd}$ 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 $\mathscr{R}_{fd}$ and $\mathscr{R}_{bd}$ in parallel composition with the ΔG process; it produces the three

Table 5.1 – ncRNA processes, states and action labels

Process	**Description**
$\mathscr{B}$	ligand binding
$\mathscr{C}$	catalysis
ΔG	free energy variation
ΔG_{fd}	free energy variation in the forward reaction direction
ΔG_{bd}	free energy variation in the backward reaction direction
$\mathscr{E}$	enzymatic activity
$\mathscr{R}$	reaction
$\mathscr{R}_{fd}$	forward reaction direction (from substrate to product)
$\mathscr{R}_{bd}$	backward reaction direction (from product to substrate)
State	**Description**
BS_*	binding site occupied
EC	electrostatic complementarity
$EC_\vee$	electrostatic complementarity check
EN	electrostatic non-complementarity
ES	enzyme-substrate complex
P_{fd}	product in the forward reaction direction
P_{bd}	product in the backward reaction direction
S_{fd}	substrate in the forward reaction direction
S_{bd}	substrate in the backward reaction direction
SC	steric complementarity
$SC_\vee$	steric complementarity check
SN	steric non-complementarity
TS_{fd}	transition state of the forward reaction direction
TS_{bd}	transition state of the backward reaction direction
Label	**Description**
aer	activation energy reduction
as	active site free
bs	binding site free
bs_*	binding site occupied
dgn	negative free energy variation
dgp	positive free energy variation
dgz	null free energy variation
ec	electrostatic complementarity
en	electrostatic non-complementarity
es	enzyme-substrate complex
l	ligand
p	product
s	substrate
sc	steric complementarity
sn	steric non-complementarity
ts	transition state

possible outputs representing the types of values that the free energy variation can assume: negative, positive or zero (dgn, dgp and dgz respectively).

A *Reaction* process ($\mathscr{R}$) can be specified in CCS as follows:

$$
\begin{aligned}
\mathscr{R} &\stackrel{\text{def}}{=} (\mathscr{R}_{fd}|\Delta G)\backslash\{dgn, dgp, dgz\} \\
&\quad\ +(\mathscr{R}_{bd}|\Delta G)\backslash\{dgn, dgp, dgz\}; \\
\Delta G &\stackrel{\text{def}}{=} \overline{dgn}.\Delta G + \overline{dgp}.\Delta G + \overline{dgz}.\Delta G; \\
\mathscr{R}_{fd} &\stackrel{\text{def}}{=} s.S_{fd}; \\
S_{fd} &\stackrel{\text{def}}{=} p.\Delta G_{fd}; \\
\Delta G_{fd} &\stackrel{\text{def}}{=} dgn.P_{fd}; \\
P_{fd} &\stackrel{\text{def}}{=} \overline{ts}.TS_{fd}; \\
TS_{fd} &\stackrel{\text{def}}{=} \overline{p}.\mathscr{R}; \\
\mathscr{R}_{bd} &\stackrel{\text{def}}{=} p.P_{bd}; \\
P_{bd} &\stackrel{\text{def}}{=} s.\Delta G_{bd}; \\
\Delta G_{bd} &\stackrel{\text{def}}{=} dgn.S_{bd}; \\
S_{bd} &\stackrel{\text{def}}{=} \overline{ts}.TS_{bd}; \\
TS_{bd} &\stackrel{\text{def}}{=} \overline{s}.\mathscr{R};
\end{aligned}
\tag{5.2}
$$

We want to point out that the modelled reaction (and eventually the corresponding multiagent simulation) is driven by the free energy reduction. The ΔG_{fd} and ΔG_{bd} processes check if the ΔG of the related reaction direction is negative.

Before producing its final output (p for $\mathscr{R}_{fd}$ and s for $\mathscr{R}_{bd}$), each reaction direction has an intermediate output, the transition state (ts).

The *Enzymatic Activty* process ($\mathscr{E}$) takes this transition state as input to catalyse the reaction, along with an active site (as). The latter is a catalytic binding site, therefore, similarly of what described for the ligand binding function, it must show steric and electrostatic complementarity with the transition state, in order for the $\mathscr{E}$ process to proceed.

If these constraints are satisfied, the $\mathscr{E}$ process makes 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 $\mathscr{B}$ process, this verification as been modelled as a non-deterministic choice.

On the ES complex acts the binding energy of the enzyme to perform the *catalysis*, modelled with

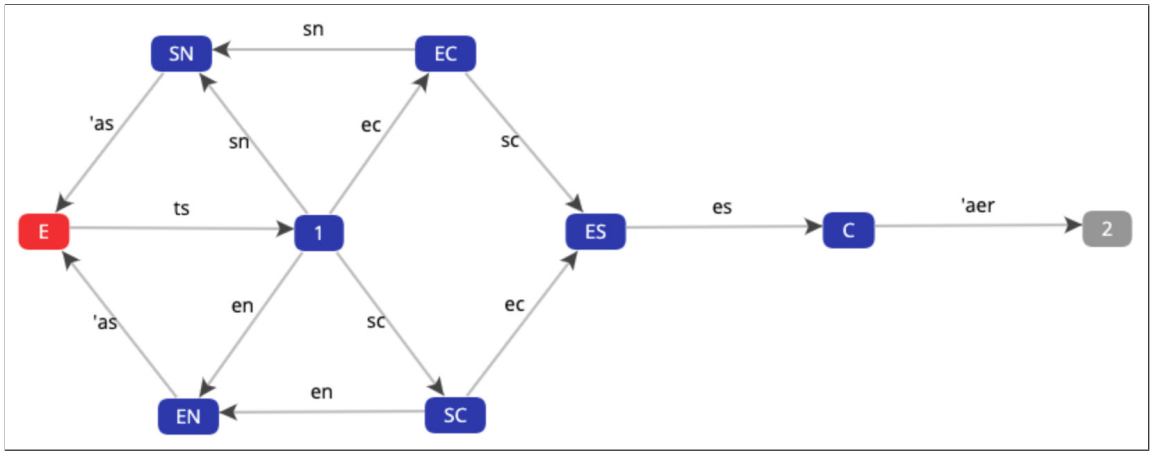


Figure 5.1 – Labelled Transition System (LTS) of the $\mathscr{E}$ process. In an LTS, each transition $P \xrightarrow{a} P'$ means that the process P can become the process P' by performing the action a. Each state has been transliterated from the CCS model, while action labels are left unchanged; output actions are indicated with a quotation mark. The state 1 represents the $SC_v + EC_v$ choice, while the state 2 corresponds to $TS_{fw} + TS_{bw}$.

the process $\mathscr{C}$, which causes the reduction of the activation energy of the reaction (*aer*), in order to obtain the ouput of one of the two reaction directions.

Here we propose a simplified specification for the model of the $\mathscr{E}$ process:

$$
\begin{aligned}
RNA &\stackrel{\text{def}}{=} as.\mathscr{E}; \\
\mathscr{E} &\stackrel{\text{def}}{=} ts.(SC_v + EC_v); \\
SC_v &\stackrel{\text{def}}{=} sc.SC + sn.SN; \\
SC &\stackrel{\text{def}}{=} ec.ES + en.EN; \\
EC_v &\stackrel{\text{def}}{=} ec.EC + en.EN; \\
EC &\stackrel{\text{def}}{=} sc.ES + sn.SN; \\
SN &\stackrel{\text{def}}{=} \overline{as}.\mathscr{E}; \\
EN &\stackrel{\text{def}}{=} \overline{as}.\mathscr{E}; \\
ES &\stackrel{\text{def}}{=} es.\mathscr{C}; \\
\mathscr{C} &\stackrel{\text{def}}{=} \overline{aer}.(TS_{fd} + TS_{bd});
\end{aligned}
\tag{5.3}
$$

To further clarify how this process works, Figure 5.1 shows its Labelled Transition System (LTS) specification, automatically generated with the aid of the web-based tool CAAL [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 5.2, the subsequent step is represented by the *model verification*. We will discuss this step in the next section, so that this chapter can 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.

5.2.3 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 verify that all the reactions are driven by the free energy reduction. Such biochemical properties are expressed as Hennessy-Milner Logic (HML) formulas so that we can ensure, via model checking, that they are satisfied [47].

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

$$RNA \models \langle bs \rangle \langle l \rangle \langle sc \rangle \langle en \rangle [\overline{bs_*}] ff \tag{5.4}$$

- If the free active site of a 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 - as):

$$RNA \models \langle as \rangle \langle ts \rangle \langle ec \rangle \langle sn \rangle \langle \overline{as} \rangle tt \tag{5.5}$$

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

$$\mathscr{R}_{fd} \models \langle s \rangle \langle p \rangle [dgn] \langle \overline{ts} \rangle tt \tag{5.6}$$

The verification that these formulas has been made with the aid of the model checking function of the web-based tool CAAL [3]. The results are shown in Figure 5.3.

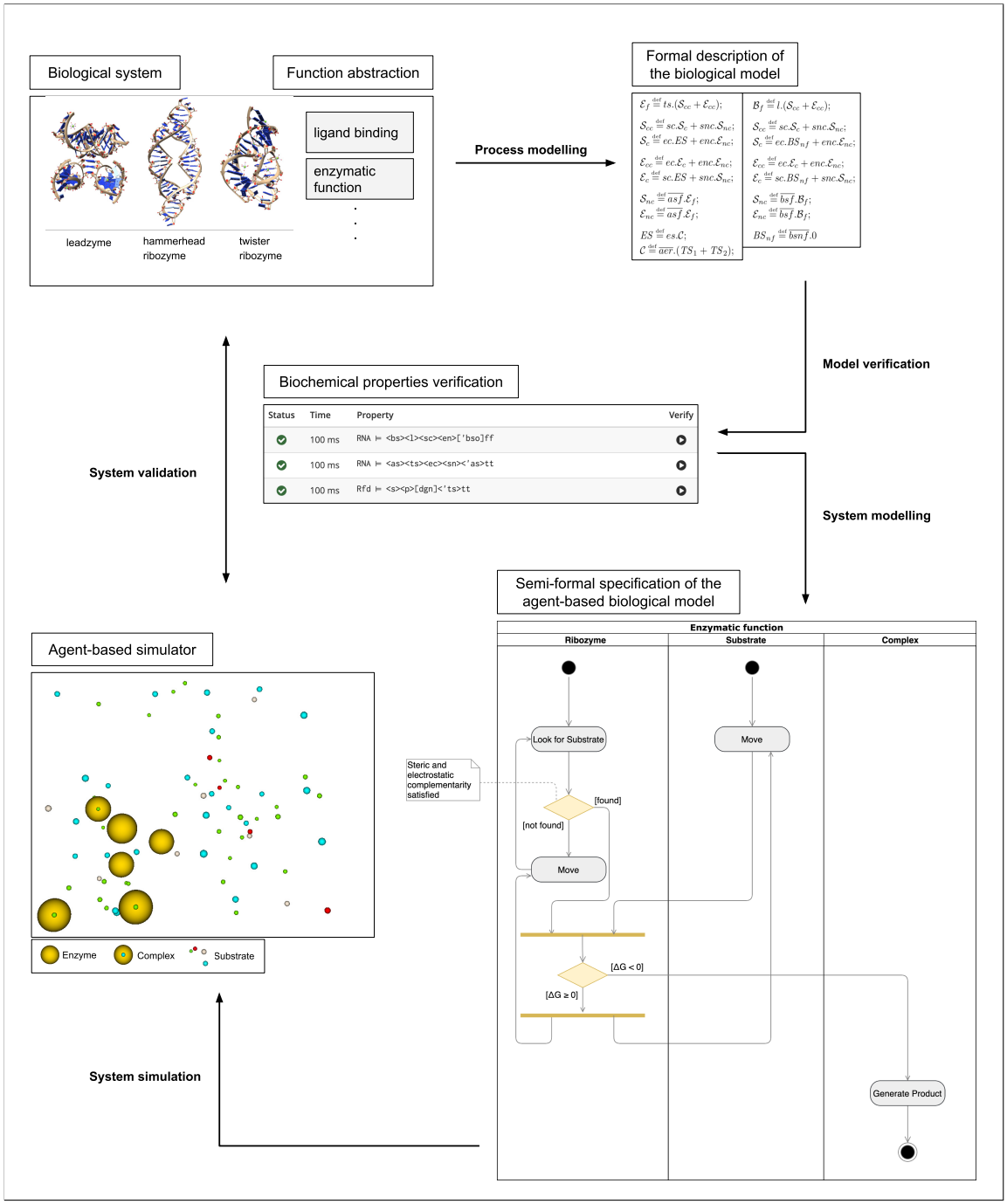


Figure 5.2 – Engineering life cycle for the simulation of ribozyme functions. We can identify five steps enclosed in two phases (represented through different formalisms): process modelling and verification; system modelling, simulation and validation. The starting point is the actual biological system [49], from which we derive an abstraction of the functions we aim to model and simulate. These functions are then formally modelled using process algebras (CCS in our case), and the properties of the models obtained verified through the best fitting method for model checking (for our models, we chose the Hennessy-Milner Logic). This phase is the one explored in the present chapter; the second phase will be defined upon the multiagent simulator described in the second part of this manuscript. It will involve the definition of a low-level specification, the generation of the actual agent-based simulation and the validation of the results obtained, intended to make the agent-based model more faithful to biological system. For the first step of this phase, we provide in this figure a semi-formal example using a UML activity diagram.

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

Figure 5.3 – Verification of some biochemical properties of the ribozyme functions, expressed as HML formulas. It has been performed through the CAAL web-based tool [3]; the checkmarks on the "Status" column indicate that all the formulas are satisfied. Output actions are represented with a quotation mark; the "bs_*" action label has been transliterated as "bso".

5.3 Conclusions

In this chapter, we provide a formal description of the functions that can be performed by RNA molecules at the abstraction level where thy have the same complexity of proteins [50]. We show how CCS, thanks to its expressiveness, can handle the complexity of modelling non-coding RNA functions, and specifically those performed by ribozymes. These functions characterise the congruence classes defined by the RNA catalytic activity. The validity of these models has been tested using the Hennessy-Milner Logic, to perform the model checking, and confirmed through an automated tool.

These results are solid basis upon which a multiagent simulator of molecular interactions can be enriched by implementing the functions of non-coding RNAs [69]. 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 5.2). Considering the results we obtained on metabolic reactions, we are optimistic that the analysis of the behaviour emerging from agents' interactions will bring new knowledge on the properties of ribozymes.

This molecules, beyond their biological function, have been applied in the treatment of 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 [28, 66, 82]. The simulations based on the models we propose in this chapter, might allow providing *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 is being increasingly acknowledged the relevance of these molecules in fundamental cellular processes, as well as their involvement in several diseases, such as in tumour progressions, where they carry out either the oncogenic or the tumour-suppressive role [74, 76]. We think that applying formal models in the study of non-coding RNA functions can provide the

perspective needed to fully understand the behaviour of this class of molecules and therefore contribute with a concrete support to handle the pathologies in which they are involved.

Part II

Agent-based Simulation of Metabolic pathways

Background and Methods for the Part II

In this chapter, we describe an agent-based modelling and simulation approach that we defined for studying the molecular interactions occurring in metabolic pathways. We choose, as a case study, the glycolysis process taking place in a species of yeast, the saccharomyces cerevisiae; for this reason, in Section 6.1, we firstly introduce some basic knowledge about the oxidation of glucose in living cells. In Section 6.2, 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 so as it can represent the input of a multiagent simulator. All these informations are necessary for a better understanding of the studies we will propose in the subsequent chapters of this second part of the manuscript.

6.1 Introduction to Yeasts' Glycolysis

The information provided in this section are intended as an overview of the reactions occurring in the glycolytic pathway. The reader that already has knowledge of these concepts can skip directly to section 6.2 on page 89, where we will describe the novel modelling and simulation methods that we defined to study such reactions.

Glycolysis is the process that degrade a molecule of glucose through a series of enzyme-catalysed reactions to yield two molecules of pyruvate. During the sequential reactions of glycolysis, some of the free energy released from glucose is conserved in the form of ATP and NADH.

When the degradation of glucose or other organic nutrients happens in absence of oxygen (anaerobic conditions) it is called fermentation, and is typical of some microorganisms, such as yeasts.

In the course of evolution, the chemistry of this reaction sequence has been completely conserved; the glycolytic enzymes of vertebrates are closely similar, in amino acid sequence and three-dimensional structure, to their homologs in yeast and spinach. Glycolysis differs among

species only in the details of its regulation and in the subsequent metabolic fate of the pyruvate formed. The thermodynamic principles and the types of regulatory mechanisms that govern glycolysis are common to all pathways of cell metabolism.

The breakdown of the six-carbon glucose into two molecules of the three-carbon pyruvate occurs in *ten steps*. We are going to describe these steps and provide the name and the acronym of the related molecular species as we will refer to them in the rest of this manuscript.

The first five steps constitute the *preparatory phase*. In the *first step*, glucose is phosphorylated to form glucose 6-phosphate (G6P), subsequently converted, as a *second step*, to fructose 6-phosphate (F6P), which, in the *third step*, is again phosphorylated to yield fructose 1,6-bisphosphate (F16bP). For both phosphorylations, ATP is the phosphoryl group donor.

In *step four*, the fructose 1,6-bisphosphate is split to yield two three-carbon molecules, dihydroxyacetone phosphate (DHAP) and glyceraldehyde 3-phosphate (GAP); this is the "lysis" step that gives the pathway its name. During the *fifth step*, the dihydroxyacetone phosphate is isomerised to a second molecule of glyceraldehyde 3-phosphate, ending the first phase of glycolysis.

The energy gain comes in the *payoff phase* of glycolysis. In the *sixth step*, each molecule of glyceraldehyde 3-phosphate is oxidised and phosphorylated to form 1,3-bisphosphoglycerate (BPG). Energy is then released as the two molecules of 1,3-bisphosphoglycerate are converted, *from the seventh to the tenth step*, to two molecules of pyruvate (PYR). Much of this energy is conserved by the coupled phosphorylation of four molecules of ADP to ATP. The net yield is two molecules of ATP per molecule of glucose used, because two molecules of ATP were invested in the preparatory phase. Energy is also conserved in the payoff phase in the formation of two molecules of NADH per molecule of glucose.

In the sequential reactions of glycolysis, three types of chemical transformations are particularly noteworthy:

1. degradation of the carbon skeleton of glucose to yield pyruvate;
2. phosphorylation of ADP to ATP by high-energy phosphate compounds formed during glycolysis;
3. transfer of a hydride ion to NAD^+, forming NADH.

In some plant tissues and in certain invertebrates, protists, and microorganisms such as brewer's yeast, pyruvate is further converted, under hypoxic or anaerobic conditions, into ethanol (EtOH) and CO_2, a process called *ethanol (alcohol) fermentation*.

Many carbohydrates besides glucose meet their catabolic fate in glycolysis, after being transformed into one of the glycolytic intermediates. The only one we will take into account in our models is glucose 1-phosphate (G1P), produced by glycogen phosphorylase and converted to glucose 6-phosphate.

Alongside the main steps described above, the breakdown of glucose can also enter one of the glycolysis branches, which led to the formation of end products such as trehalose (TRH), glycogen (not modelled), glycerol (GLY) and succinate (SUC).

A schematic representation of the steps and branches taken into account in our studies is provided on Figure 6.1 on the next page.

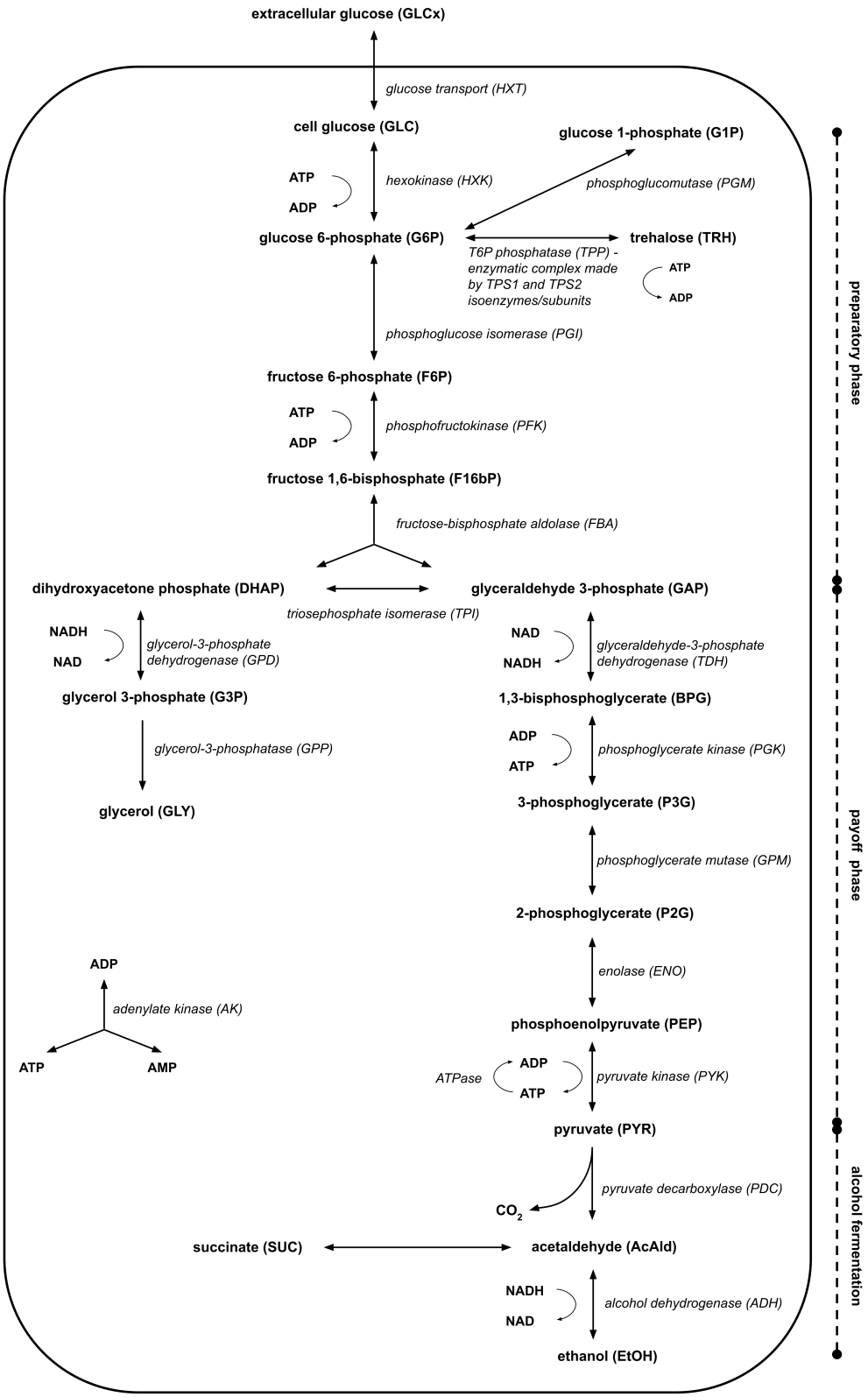


Figure 6.1 – Schematic representation of the glycolysis steps and branches taken into account in the subsequent chapters. On the right side of the image, we highlighted the three phases identifiable in yeast glycolysis. For each metabolite and enzyme involved, we reported both the name and the acronym adopted in this manuscript (in bold for the metabolites, in italics for the enzymes).

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

6.2.1 Agent-based Simulator for Metabolic Pathways

The study proposed in this manuscript has been carried out with the aid of OrionV2, a spatial simulator for metabolic pathways. It has been developed in Java starting from Orion, a prototype project carried out at the University of Camerino [6, 26, 53]; we fixed and refined the original software to let it capable of dealing with a large amount of molecules and highlighting their interactions.

OrionV2 is a multiagent simulator, this means that the molecules involved in the pathway are represented by agents, software peaces able to perceive changes in the environment and react to them.

To provide some basic formalism, a *reactive agent* is efined by a 6-tuple $\langle E, Per, Ac, see, do, action \rangle$ 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: $E \rightarrow Per$
- $action$: $Per \rightarrow Ac$
- do: $Ac \times E \rightarrow E$

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

The simulations are performed in the three-dimensional space and each molecule is modelled as a sphere, whose radius is estimated from its molecular weight and the average value of the molar specific volume of a protein in solution [36, 77, 90].

The simulator allows to set space unit and time scale as per requirement; in our studies we considered the angstrom (10^{-10} m) for space and 10^{-4} second for time (corresponding to one tick of the simulation clock). A cube of 1 femtolitre (having a side of 1000 Å) represents the best option for the aim of our study and meets the computational demand described above.

Every molecule is able to freely move inside the simulation volume. The movement of a molecule is given by a vector applied to the centre of its sphere; its module is calculated from the ambient

diffusion coefficient D via the following Equations 6.1 and 6.2, while its direction is calculated randomly basing on polar coordinates.

$$D = \frac{k_B T}{6\pi\eta r} \tag{6.1}$$

where, k_B is the Boltzmann constant, T is the temperature, η the viscosity of the environment and r is the radius of the molecule.

$$< x^2 > = 2Dt \tag{6.2}$$

where, assuming Brownian motion, $< x^2 >$ is the average value of the square of the distance covered in a time t.

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

The model at the basis of the simulator classifies molecules in three types: *enzymes*, *complexes* and *metabolites*. The property that distinguish enzymes and complexes from metabolites is that the latter are just able to move while the first two classes of molecules can act on the environment.

Interactions between enzymes and metabolites are modelled through a *perception paradigm*: each enzyme is able to identify the cognate metabolites in its proximity thanks to a perception-sphere that it projects on the environment (see Figure 6.2 for a representation of this sphere in the form of the potential interactions that an enzyme can perform).

As better explained in Chapter 7, such an approach is the simulator key-feature that allows us to study the effects of the 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 length of perceptions and the related molecular behaviours.

6.2.2 From a Kinetic Model to a Multiagent Simulation

To study in silico the effects of molecular interactions in a metabolic pathway, we need a kinetic model that contains the sequence of reactions characterising the pathway we want to study; form this model, we also gain some quantitative data, like the initial concentrations of the species involved (as it will better explained later in this section). In this perspective, a kinetic model represents just the source of our study and a reference to interpret our results. It provides a static representation of the system by describing its global properties through differential equations; however, we aim to show if kinetic data actually underlie processes related to the ability of molecules to perceive each other, even from a long distance. For that reason, a multiagent model of molecular interactions provides a better baseline over which carrying out in silico studies on molecular perception and long-distance interactions (see Chapters 7 and 8).

A multiagent approach, indeed, allows describing interactions at a local level; however, it also maintains compositionality, that is the capability of recursively applying the rules characterising agents interactions to progressively define higher abstraction levels. In this way, we are able to hide the unnecessary details of a specific level but, at the same time, to observe its global behaviour [11, 16].

Considering the specific case of a metabolic pathway, a kinetic model treats enzymatic reactions as mathematical functions that relate the concentrations of reactants to those of products and hide the role carried out by each molecular interaction. Conversely, in our agent-based model, each molecule is represented by an agent able to perceive the environment and the cognate partners with which it can interact. A similar approach may also be adopted by defining a molecular dynamics model; however, this kind of method places the analysis at an atomistic level and the related simulations have a high computational load. The compositionality of MAS models, instead, permits to conduct the study at a macromolecular level, without loosing in accuracy and performing light-weighted simulations.

As better described in Chapter 7, we exploited the capability of agents to perceive each other, to allow enzymes interacting with cognate metabolites placed at various distances, and simulate in this way the effect of long- and short-range forces on the system.

Enzymatic Reaction Automaton

In Chapter 5 we defined a reaction, such as $S \rightleftharpoons P$, as formed by two complementary *reaction directions*, represented by the following processes:

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

By analysing the dynamics of a biochemical reaction in the a metabolic pathway, we can now model the following molecular entities:

- *Free enzymes*, seeking a substrate to interact with.
- *Dual-complexes*, formed when an enzyme binds a cognate metabolite; they are unstable since they need a third metabolite to be saturated and generate the final product of the reaction.
- *Saturated enzymes*, representing the final complex of the reaction; they are formed by an enzyme linked to one or two metabolites, stably for a time interval given by the k_{cat} value of the reaction (when it runs out, the enzyme returns free and the reaction product is released in the environment).

An enzymatic reaction cycles through these three states; however, they can be modelled only by taking into account the local properties of the molecular interactions. Conversely, a kinetic approach describes the pathway globally to analyse the time-dependence of the metabolite concentrations through a set of ordinary differential equations. These equations threats enzymes just as functions from the substrate to the product (and viceversa), while to represent properly the local interactions we need to represent this type of molecules as autonomous entities.

Therefore, we model the cyclical pattern of an enzymatic reaction through the definition of an *automaton* based on the molecular entities described above.

Using the CCS process algebra, we can define, generalising the forward and backward reactions, the process $\mathscr{R}$ such that:

$$
\begin{aligned}
&\mathscr{R} \stackrel{\text{def}}{=} e.E_{m1} + e.E_{m2};\\
&E_{m1} \stackrel{\text{def}}{=} m1.DC1 + m1.ES;\\
&E_{m2} \stackrel{\text{def}}{=} m2.DC2;\\
&DC1 \stackrel{\text{def}}{=} m2.DC1_{m2};\\
&DC2 \stackrel{\text{def}}{=} m1.DC2_{m1};\\
&DC1_{m2} \stackrel{\text{def}}{=} m2.ES;\\
&DC2_{m1} \stackrel{\text{def}}{=} m1.ES;\\
&ES \stackrel{\text{def}}{=} \overline{pm}.\mathscr{R};
\end{aligned}
$$

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 like ATP or NADH;
- pm generalises the products of the reaction (one or more);
- E_{m1} and E_{m2} are the states representing the enzyme perceiving a cognate metabolite;
- $DC1$ and $DC2$ are the dual-complexes of the enzyme with m1 and m2 respectively;
- $DC1_{m2}$ and $DC2_{m1}$ are the states in which the dual complexes perceive the metabolite needed to saturate the enzyme;
- ES is a saturated enzyme.

In Figure 6.3, we provide the labelled Labeled Transition System (LTS) of the algebraic definition of the reaction automaton.

Since OrionV2 simulates the molecules as spheres (see Section 6.2.1), we were able to implement this model by allowing the formation of larger spheres as the result of the interaction between two cognate molecules. The volume of the sphere representing a molecular complex is calculated from the sum of the originating molecules' weight. Figure 6.4 provides a schematic representation of the automaton for the case in which the enzyme interact with two metabolites.

6.2.3 Choosing a Reference Kinetic Model

A multiagent model of molecular interactions requires that each molecule can be represented as an agent. For that reason, we need to gain the data necessary for the simulation from a model that provides enzymatic concentrations.

Moreover, not all the kinetic parameters can be excluded: in order for a saturated enzyme to generate the products of the reaction, it is fundamental that it respects the correct timing of the metabolic pathway by waiting an amount of time obtained from the k_{cat} value (or turnover number); this represents the number of molecules transformed by an enzyme in one second.

The K_m, which measures the affinity of an enzyme for a specific substrate, is also needed, since an enzyme can form a complex with an encountered metabolite randomly or on the basis of a list constructed over the k_{cat}/K_m ratio (specificity constant). This possibility can be established in the initial setup of the simulator trough its input XML file.

Basing on this requirement, we identified in the "Smallbone2013 - Iteration 18" [79] a model particularly suitable for the aims of our study, since it contains a complete set of experimental data on the isoenzymes involved in a well-studied process, the glycolysis of Saccharomyces cerevisiae.

The Smallbone2013 model of glycolysis provides a detailed description of the chain of reactions that generate energy from glucose by braking it into two molecules of pyruvate. In addition to the main branch of glycolysis, the Smallbone2013 model includes the glycerol, glycogen and trehalose branches and also considers the alcoholic fermentation steps, which lead to the formation of ethanol (see Figure 6.1).

6.2.4 Defining the Input for the Simulation

The input of the simulator is an SBML (Systems Biology Markup Language) model, retrieved from the literature and filled with experimental data [42]; 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 SBML file. As we said in the previous section, for the study described in this manuscript we choose the Smallbone2013 model, provided in the SBML form accessible at `http://identifiers.org/biomodels.db/MODEL1303260018`.

A dedicated interface of the simulator converts the SBML model in an XML file specifically formatted to be interpreted by the simulator itself, but also to be human-readable.

Therefore, its main function is to translate the kinetic representation of the metabolic reactions into our agent-based model. To do this, for each reaction in the SBML model, it gets the reactants and products and generates a XML code for each of its interactions, basing on the algebraic definition provided in section 6.2.2. It also associates to the defined reaction its k_{cat} value and the Km values of all its interactions.

As an example, we consider a reaction catalysed by the enzyme `E`, with two substrate metabolites (`M1` and `M2`) and two products (`P1` and `P2`).

Starting from the generalised SMBL:

```
<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="P2"/>
    <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="km_value"  units="mM"/>
      <parameter metaid="metaid_value" id="Km2" value="km_value"  units="mM"/>
    </listOfParameters>
```

the converter generates the following XML code:

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

```
        <Km unit="mM">Km_value</Km>
      </interaction>
      <interaction>
        <reactants>
          <reactant id="E+M2"/>
          <reactant id="M1"/>
        </reactants>
        <products>
          <product id="E+M1+M2"/>
        </products>
        <Km unit="mM">Km_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 algebraic model), while `E+M1+M2` represents the saturated enzyme. It is important to notice that this representation sets just the information needed to start the simulation. The Km of the last interaction is always 0, since it represents just the release of the products of the reaction.

In addition, the model conversion interface retrieves from molecular databases, specifically ChEBI [38] and UniProt [86], the molecular weights, needed for the simulation but missing in the SBML model.

The enzyme perception paradigm has been embodied in the simulator itself and will be examined more in depth in the following chapters.

6.2.5 Simulation Output and Visualisation

The output of the simulator is a set of CSV files reporting the type and number of molecules contained in the simulated environment, along with their position, at each instant of simulation.

An option lets the user to generate, starting from these files, the plots of the concentration changes over time (Figure 6.5).

A 3D interface allows observing the evolution of the system (see Figure 6.6 for a screenshot of the simulation environment at the beginning of a simulation).

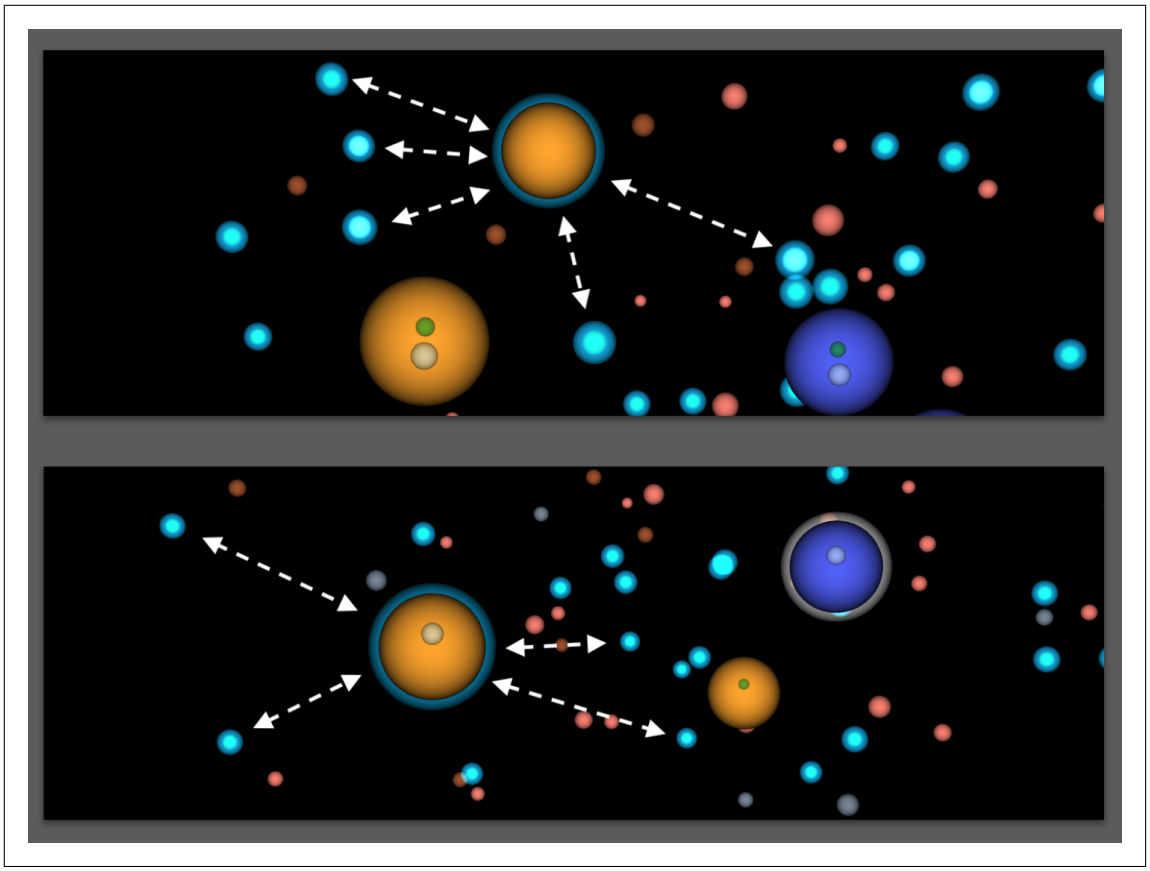


Figure 6.2 – Representation of the *perception paradigm* underling the molecular interactions in the simulated environment. All the molecules are represented as spheres, from which the larger ones correspond to the enzymes; when they are bound to one or two metabolites, the smaller molecules are shown as attached to the sphere of the enzyme. To maintain the clarity of the illustration, the perception spheres are not explicitly represented; instead, a perceiving enzyme and the metabolites in its perception volume are highlighted in blue. In the top figure, we show the potential interactions of a free enzyme; in the figure below, a similar situation is depicted for a complex made by an enzyme with a bound metabolite. The white arrows point out that each interaction in the multiagent system is 2-body.

h!t

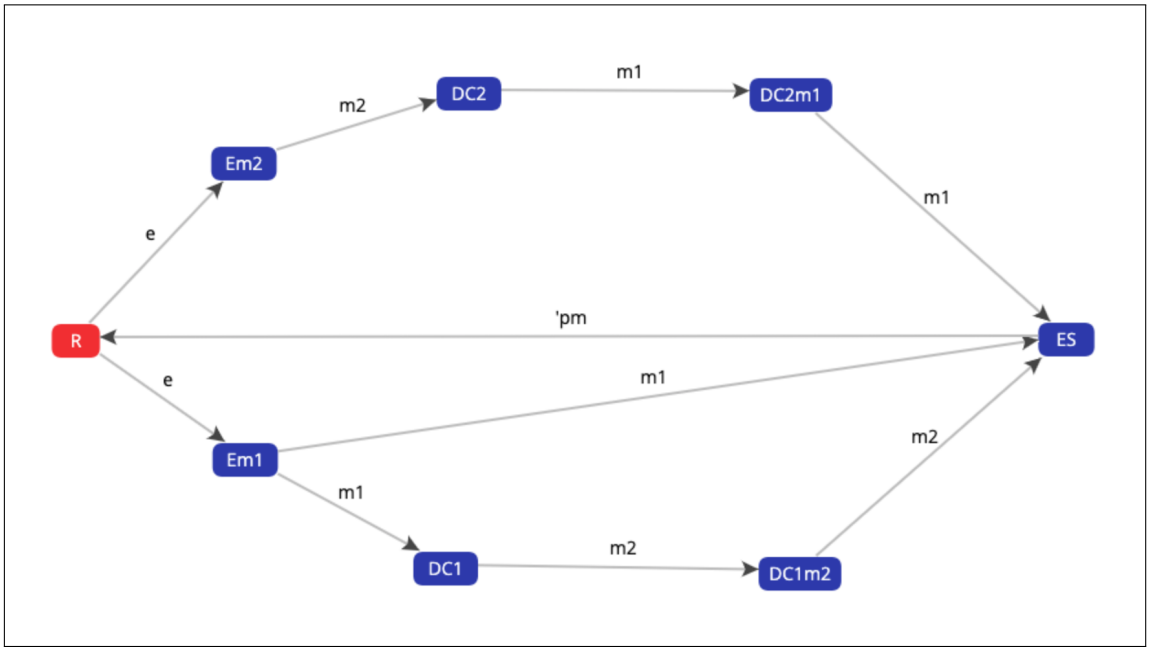


Figure 6.3 – Labeled Transition System of the automaton representing an enzymatic reaction in our agent-based model. It has been generated from the algebraic definition provided in this chapter through the web-based tool CAAL [3]. The names of the states are transliterations of the names provided in the CCS formula.

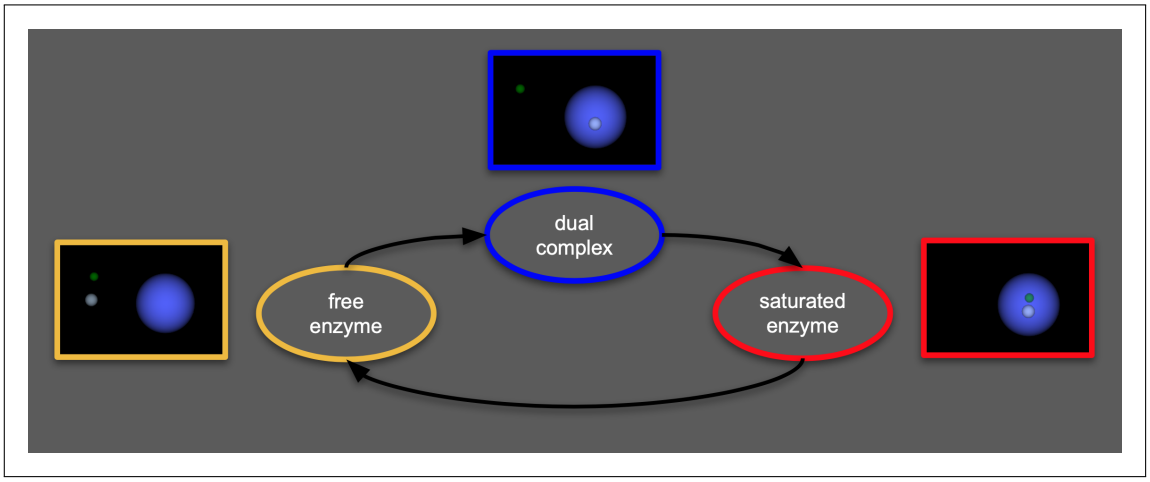


Figure 6.4 – The three states of the enzymatic reaction automaton in which the enzyme interacts with two metabolites. Each state has been associated with a representation of the related molecular entities in the agent-based model. For better show the molecules involved in the formation of a molecular 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.

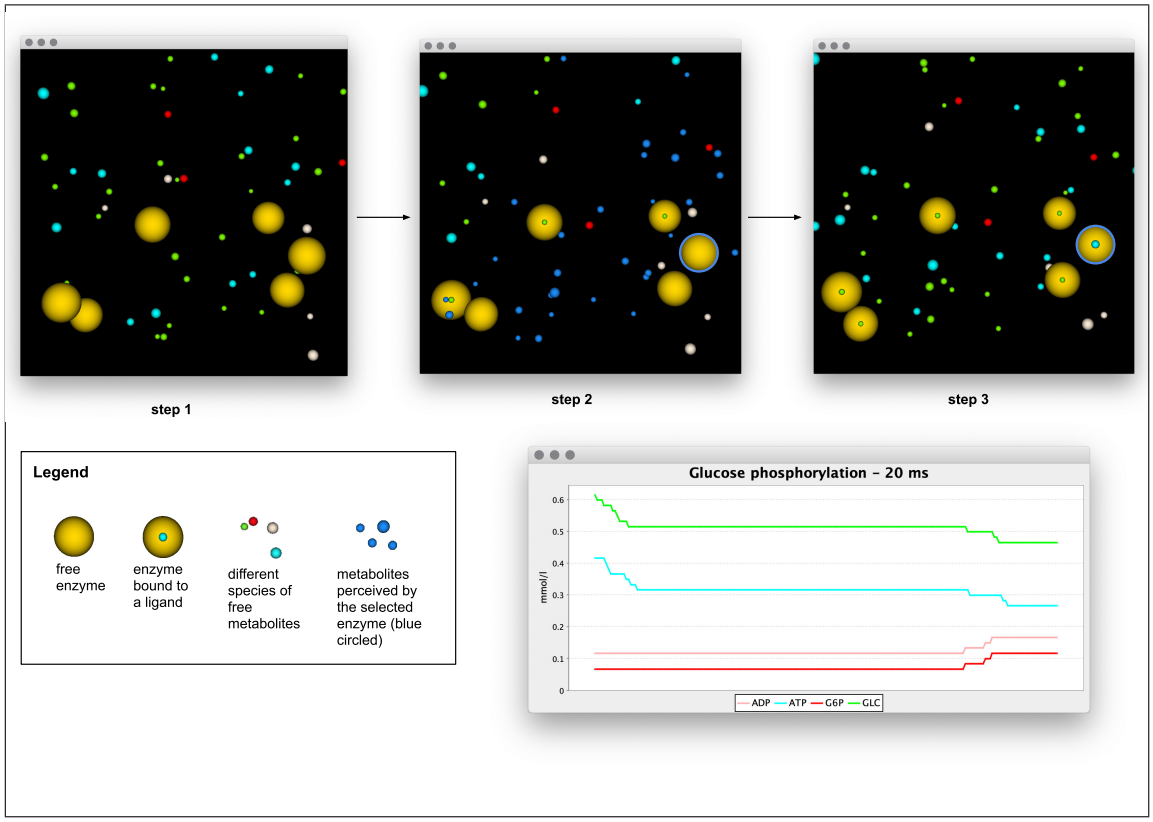


Figure 6.5 – Agent-based simulation of the molecular interactions involved in an enzymatic reaction. In the upper part of the figure, we show the steps needed for an enzyme to bind to a cognate ligand in a simplified run of the simulation: in the *first step*, all the enzymes (yellow spheres) and metabolites (small spheres of various colours) are freely immerse in the three-dimensional environment; the *second step* shows how a selected enzyme (highlighted by a blue circle) perceives all the affine metabolites it can reach (coloured in blue); in the *third step*, the enzyme binds one of the identified metabolites. The plot shown on the lower right is the output of an actual short simulation (20 ms) of the reaction catalysed by hexokinase.

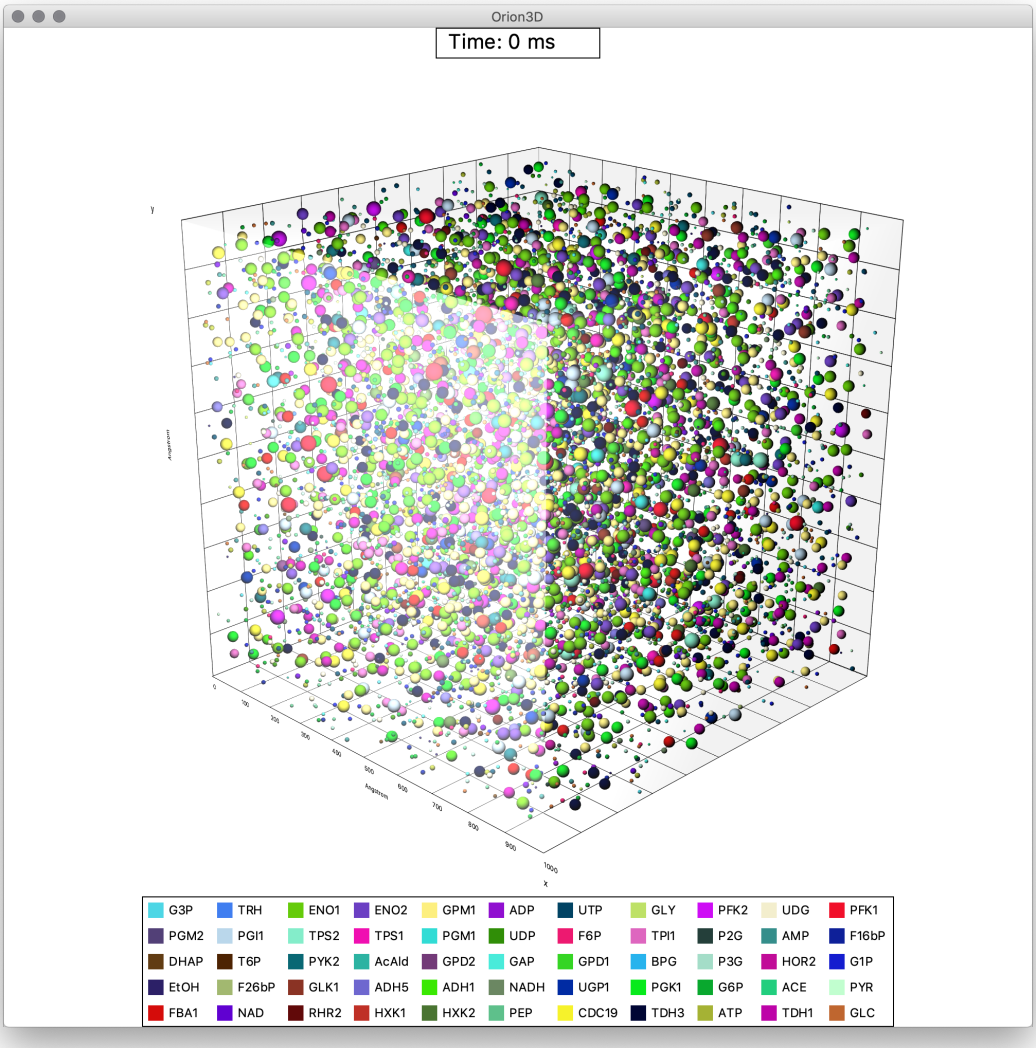


Figure 6.6 – Light-themed 3D interface of OrionV2. The cube representing the volume of simulation has side of 1000 Å. The interface shows the position of every molecule instant by instant. It is also possible to highlight the metabolites perceived by each enzyme in a specific moment of the simulation. At the bottom of the interface, a legend associate each molecule with its respective colour.

Testing in Silico the Long-distance Electrodynamic Interactions among Biomolecules

7.1 Introduction

Interactions larger than the Debye screening length ($\simeq$ 10Å) between cognate partners biomolecules, intended as bulk diffusion, are not well investigated. However, while long-distance electrostatic interactions have been considered unlikely, electrodynamic interactions, occurring between oscillating electric dipoles, have a long-range nature. Experimental evidence for the existence of collective excitations in biological macromolecules is available in the Raman and far-infrared spectroscopic domains [25, 65]. As shown by Nardecchia et al. [62], the overall interaction potential is generically composed of a resonant long-range term r^{-3} and a short-range term r^{-6}, where r is the intermolecular distance; therefore, an attractive (resonant) potential $U(\overrightarrow{r}) \sim r^{-3}$ should be added to the random Brownian force.

To provide an efficient simulation environment to study such long-distance interactions, we adopted a many-body approach, implemented in the form of a multiagent simulator, as described in Chapter 6. We deepen the perception paradigm at its basis and exploit this property to simulate the ability of the enzymes to identify distant cognate metabolites. For this investigation, we set the molecular interactions to be completely random, without establishing any priority on the metabolite perceived by an enzyme. We aim in this way to analyse the effects the long-range forces as system properties emerging from the local bimolecular interactions.

We opt for a multiagent environment instead of a simulator based on Molecular Dynamics because our approach need a compositionality that an agent-based simulation can handle more effectively than a model relying solely on differential equations; besides the higher computational cost that the latter would have by describing the system at an atomistic level.

We think that a network of mutually conditioned reactions may better bring out the global effect of the long-distance electrodynamic interactions on a biological system; for that reason, we decided to model a well known metabolic pathway, the glycolysis, for which a large amount of data is available in the literature.

7.2 Integrative Methods for this Chapter

7.2.1 Long-distance Electrodynamic Interactions

Charge oscillating at high frequency (in the range of $10^{10} - 10^{11}$ Hz) does not suffer Debye screening effect by the ions of the medium and a macromolecule behaves like an oscillating dipole; long-range forces may be activated between two resonant dipolar systems. Considering two dipolar molecules A and B, vibrating at frequencies ω_A and ω_B respectively, for $\omega_A \gg \omega_B$ the intermolecular interaction is a short-range $U(\overrightarrow{r}) \sim r^{-6}$, where r is the intermolecular distance, while the creation of dipoles activates a long-range interaction $U(\overrightarrow{r}) \sim r^{-3}$ between the two molecules (provided that $\omega_A \simeq \omega_B$, that is at resonance) [62].

7.2.2 Modelling the Whole Glycolytic Pathway

In the previous chapter, we introduced the kinetic model chosen to gain the data necessary for our simulations. It is the "Smallbone2013 - Glycolysis in S.cerevisiae - Iteration 18" [80], whose SBML is accessible at `https://www.ebi.ac.uk/biomodels/MODEL1303260018`. We opted for this model because it contains a complete set of experimental data (including enzymatic concentrations) about the glycolytic pathway of the well-studied organism Saccharomyces cerevisiae.

By importing the reactions of the SMBL file as the input of our agent-based simulations, we excluded all those for which the Smallbone2013 model does not provide enzymatic concentrations. Our simulator can actually handle this kind of reactions, since we can model them in terms of their bulk effects; however, for the aim of observing the emergent behaviour of the the long-distance interactions, introducing any bulk reaction would perturb the environment and hide the absence of actual interactions among the molecules modelled as agents.

Basing 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 cytosol and the extracellular environment). 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 [34]. However, as we will detail in the remainder of this section, the length of the simulations makes the allosteric regulation and the whole energy charge effects negligible.

Nonetheless, according to most of the literature, we modelled the reactions catalysed by hexokinase, phosphofructokinase and pyruvate kinase as irreversible [8, 17, 45], since they function as control points of the whole glycolysis process, despite in the Smallbone model they are considered reversible.

Our agent-based model is intended as the basis 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 (indeed, the succinate branch is already turned off in the Samlbone2013 model).

The resulting subset of reactions characterising the model at the basis of our simulations can be found in table 7.2. For further details on the process needed to convert a kinetic model into an agent-based model, refer to Chapter 6.

7.2.3 Simulating a Large Number of Molecules

Although agent-based simulations have a fairly light computational load, simulating a metabolic pathway involves thousands of molecules, and therefore as many agents running concurrently. The resultant 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 8.1, we report the initial concentrations of all the simulated species. The total number of molecules in the environment at the beginning of the simulation is 6955.

We run our simulations on a cloud-based virtual machine, powered by an Intel Skylake CPU with 8 vCPUs, and 32 GB of memory. Despite these hardware resources, a run of the simulation progresses of 1 instant of simulated time (10^{-4} seconds) about every 10 seconds in real time. This means that simulating 0.1 seconds requires a run of roughly 24 hours. For this reason, we choose this interval of 0.1 seconds as the standard value for our studies. Even if it could be too short for observing some biological fenomena, such as the effects of enzyme activations and inhibitions, it revealed to be sufficient to highlight the impact of the long-distance electrodynamic interactions on the glycolytic pathway.

Metabolites		Enzymes	
Name	Initial Conc. (mmol/L)	Name	Initial Conc. (mmol/L)
ADP	0.129	CDC19	0.205
ATP	0.429	ENO1	0.686
BPG	0.007	ENO2	0.197
DHAP	0.116	FBA1	0.134
F16bP	0.458	GLK1	0.045
F6P	0.235	GPD1	0.068
G1P	0.539	GPD2	0.008
G3P	0.274	GPM1	0.730
G6P	0.772	HOR2	0.055
GAP	0.316	HXK1	0.017
GLC	0.628	HXK2	0.061
NAD	0.150	PFK1	0.047
P2G	0.068	PFK2	0.039
P3G	0.470	PGI1	0.138
PEP	0.610	PGK1	0.258
PYR	0.211	PGM1	0.033
T6P	0.020	PGM2	0.013
UDP	0.282	PYK2	0.061
UTP	0.649	RHR2	0.051
AMP	0.440	TDH1	0.351
NADH	0.087	TDH2	0.000
UDG	0.467	TDH3	0.420
F26bP	0.030	TPI1	0.294
GLCx	0.740	TPS1	0.034
GLY	0.150	TPS2	0.027
TRH	0.015	UGP1	0.062

Table 7.1 – Initial concentrations of the molecular species simulated in our studies. The original values of the Smallbone2013 have been scaled to fit the computational demand of the multiagent simulations.

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

Table 7.2 – Table of the reactions gained from the Smallbone2103 model to define the multiagent model at the basis of our simulations. As explained in the Introduction, 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. The reactions are shown in alphabetic order; the related k_{cat} values are also reported.

7.3 Results

To simulate the effect of molecular long-distance interactions, we constructed an agent-based model characterised by a *perception-based paradigm* specifically defined for this purpose. Its core property lies on the definition of a *perception sphere* that surrounds each active molecule (enzymes and complexes, as better explained in Chapter 6). By setting the radius of the perception sphere (*perception radius*), we can model different distances at which enzymes and complexes are able to find their cognate metabolites.

Each perception radius is obtained by summing the radius of the enzyme to the *perception distance* at which we want that the enzyme could 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 a metabolite diminishes; for this reason, each perception sphere is characterised by different interaction probabilities, depending on its size.

Since our aim is to compare the effects of long- and short-distance interactions, we focus this study on three perception distances: 5 Å, 10 Å and 300 Å.

- A perception sphere with radius of 5 angstroms sets the space on which the Van der Waals forces act; this means that, when a metabolite enters this sphere (of a cognate enzyme), there is a probability $p = 1$ that the interaction will happen.

- A 10 angstroms radius models a distance affected by the Debye screening and restricts the interactions to just those allowed by short-range Coulomb forces; in this case, the probability of the interaction reduces from 1 to 1/2 when the metabolite is detected at a distance d, such that $5 < d \leq 10$ angstroms.

- A radius of 300 angstroms has been chosen as the average length to simulate the existence of long-range forces among biomolecules (also considering that the size of the simulation volume of our study is 1000 cubic angstroms). A perception sphere of this size, is modelled with four different interaction probability intervals. Specifically, let p be the probability of interaction, d_m be the distance of the metabolite from the centre of the sphere representing the perceiving enzyme, r the radius of such a sphere and d_p the perception distance (all the lengths expressed in angstroms):

  - if $d_m \leq r + 5$, then $p = 1$

  - if $r + 5 < d_m \leq d_p/4$, then $p = 3/4$

  - if $d_p/4 < d_m \leq 3/4\, d_p$, then $p = 1/2$

  - if $3/4\, d_p < d_m$, then $p = 1/4$

As mentioned above, we consider this modelling choice 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 7.1, we provide a graphical representation of how the perception radii project on the environment.

By setting the local rules that determine movements and interactions of the molecules involved in the yeast glycolysis, the global behaviour of the pathway emerges in the form of variations in the species concentrations. The outputs of the agent-based simulator described in this chapter allow us to analyse these variations in the form of plot of concentration changes (mmol/L) over time (ms).

Comparing the results obtained by simulating the glycolytic pathway for an interval of 0.1 seconds, we observe interesting behaviours. In particular, as shown in Figure 7.2a, the simulations performed with perception radii of 300 angstroms has the highest reactivity; however, differently from our initial expectations, they are fairly close to the concentration changes obtained by simulating perception radii of 10 angstroms. In both cases, the simulations generate significant amounts of the glycolysis end-products, that is pyruvate, ATP and NADH.

This means that our in silico experiments suggest that the Debye screening may not be sufficient to limit the progression of the pathway in short intervals of time and that, in the first analysis, the long-range forces might not have a significant impact on the oxidation of glucose.

A remarkably different result is obtained in the case of simulations based on perception radii of 5 angstroms, which model a system affected only by short-range Van der Waals forces. Despite the 100% probability of a metabolite to be "captured" by an enzyme when it enters so small interaction spheres, at the end of the simulation we can observe a negligible increases in the concentration of the the pathway end-products.

Nevertheless, not all the metabolite species show the same low-reactivity; intermediate products, such as G6P, F6P or GAP, have concentration changes close to those observable in the simulations with perception radii of 10 or 300 angstroms (see Appendix C for a comparison of the concentration changes for all the species simulated).

Moreover, the concentration of glucose decreases similarly in all the three types of simulation; this hints at the possibility that excluding the long-range forces from the pathway affects just the efficiency of the glycolytic process, by reducing the production rate of pyruvate, ATP and NADH, but not its overall effectiveness.

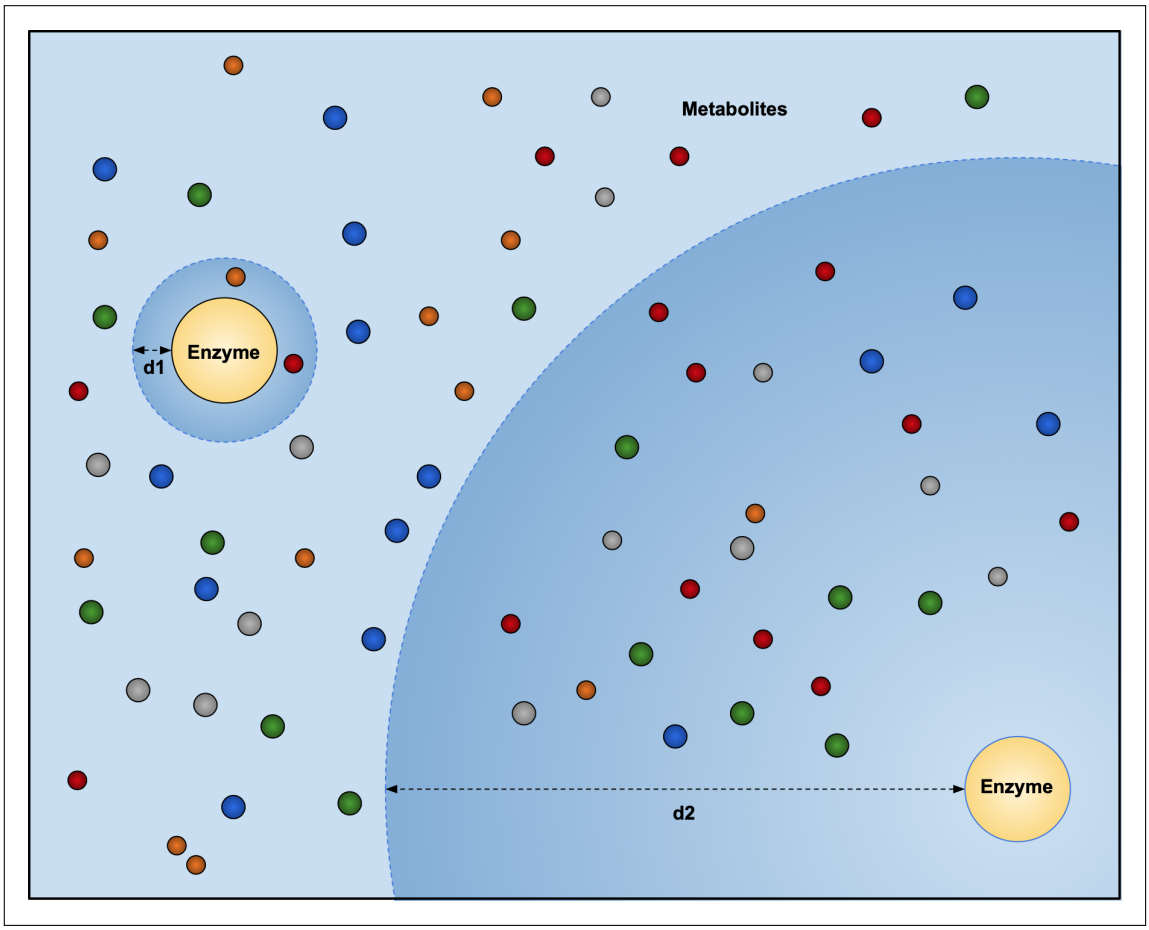


Figure 7.1 – Graphical representation of the enzyme's perception radii. The radius of the **d1** type limits the enzyme interactions to those allowed by short-range forces, while a **d2** type radius models the effect of long-distance interactions.

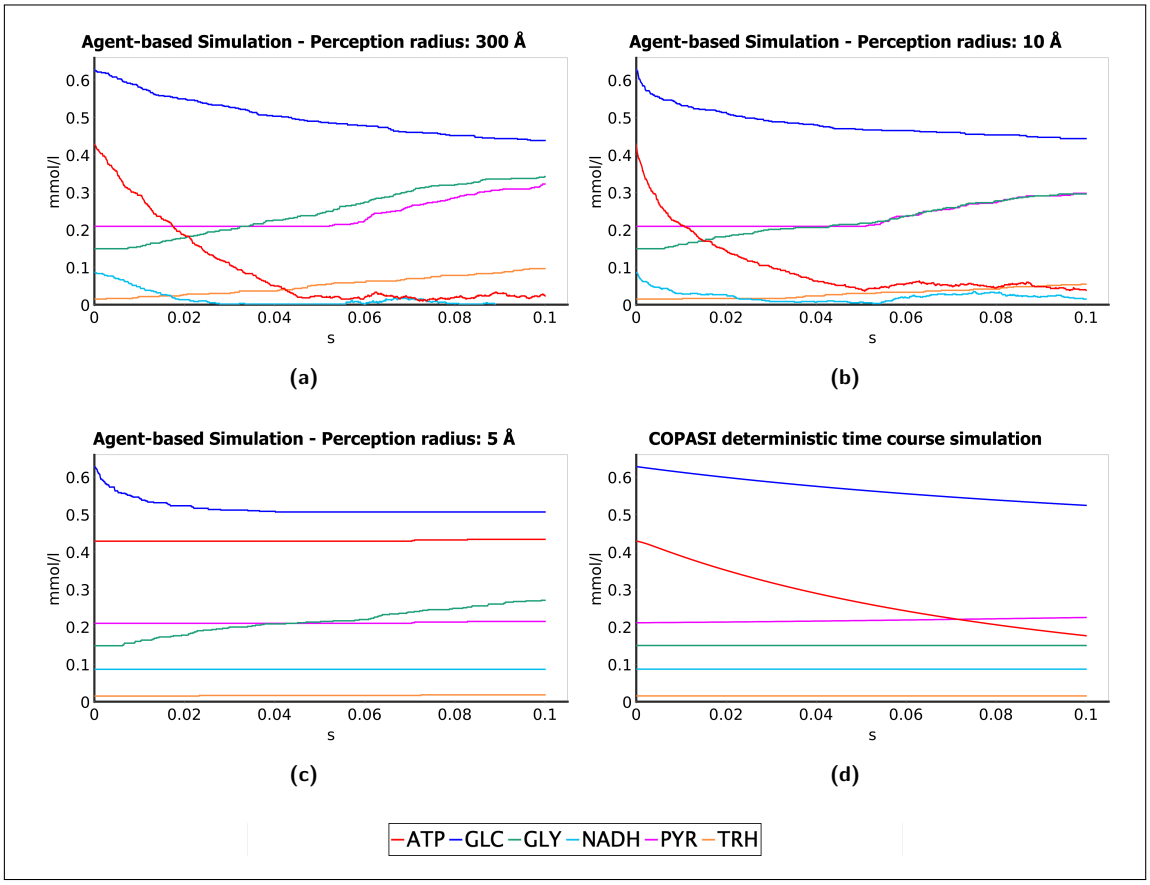


Figure 7.2 – Concentration changes of significant metabolite species in different simulations of 0.1 seconds. Through this figure we provide a comparison of the plots generated by three multiagent simulations – with perception radii set to 300 Å **(a)**, 10 Å **(b)** and 5 Å **(c)** respectively – and by a deterministic time course simulation based on the Samllbone2013 kinetic model **(d)**. The selected metabolite species are glucose (GLC) – the source of the glycolytic pathway – as well as pyruvate (PYR), trehalose (TRH), glycerol (GLY), NADH and ATP – which represents the end products of glycolysis and of the branches we considered in our agent-based model (trehalose and glycerol). It is possible to notice how the simulation that takes into account long-range forces **(a)** also shows a higher reactivity and a noticeable increase in the amounts of the pathway end products. Conversely, the simulation which limits the interactions to those allowed by Van der Waals forces **(c)**, generates a negligible amount of end products, even if it consumes a similar quantity of glucose. We interpret these results as a hint that the long-distance interactions affect the efficiency but not the effectiveness of the glycolytic pathway. The plot resulting from the deterministic simulation of the Smallbone2013 model **(c)** has been obtained via the software Copasi [41]. The discrepancy between this output and our analysis on the effects of the long-distance interactions turns out to be a possible argument in favour of the validity of the agent-based approach over the standard kinetic modelling; indeed, comparing the pathway reactivity tendency of these plots with the results of in vivo experiment shown Figure 7.3, it is possible to observe that the latter are closer to the reactivity tendency characterising the the simulation based on larger perception radii.

7.4 Discussion

Exploiting an agent-based simulator instead of relying on molecular dynamics gave us the possibility of implementing simple local rules that requires a light computational load to be executed; in this way, we were able to carry out preliminary studies that do not need to define computationally high-weighted sets of differential equations and to take into account the effect of the whole system on each interaction.

Our in silico experiments gave us a hint that the long-range forces might not be fundamental for the step-wise progression of the whole glycolytic pathway. Even limiting the interactions to those performed by Van der Waals forces (5 Å perception radius), we can still observe a reduction in the glucose concentration and evident changes in most of the intermediate products. However, in this case, the end products show variations too small to be relevant, suggesting that long-distance interactions may have a significant impact the efficiency of the process, and therefore may be crucial when the energy demand of the cell increases.

As we said Chapter 6, we referred to the Smallbone2013 model primarily to gain the initial concentrations of the molecular species and the turnover numbers of the enzymes; however, by comparing our results with the time-course deterministic simulation based on the kinetic equations provided by this model, we are able to propose further considerations on our findings.

Indeed, our analysis on the effects of the long-range forces on metabolic pathways goes against the results of the deterministic simulation based on the Smallbone kinetic model; as observable in the plots of Figure 7.2, the pathway behaviour it describes is closer to the output of our simulations based on 5 Å perception radii than to those obtained with 300 Å radii. This means that, at least for short intervals of time, the kinetic model sustains a limited effect of the long-distance interactions not only on the effectiveness, but also on the efficiency of the glycolytic pathway.

However, Teusink et al. questioned that the in vitro kinetics could be able to accurately describe an in vivo behaviour and Westerhoff et al. observed the non-robustness of a silicon cell for yeast glycolysis [84, 88].

To verify the reliability of our simulations, we compared our results with the in vivo studies conducted by Teusink et al. on the glycolysis of a yeast intact cell [84]. They extends over time intervals significantly larger than those we took into account in our studies, therefore we cannot perform a detailed comparison; nonetheless, the reactivity tendency of these experimental findings matches more the behaviour emerging from the effect of long-range forces then the trend observable from the deterministic simulation of the kinetic model. In Figure 7.3, this tendency can be noticed for glucose, glycerol, pyruvate and trehalose.

The models constructed bottom-up, through the definition of local interactions, enable the emer-

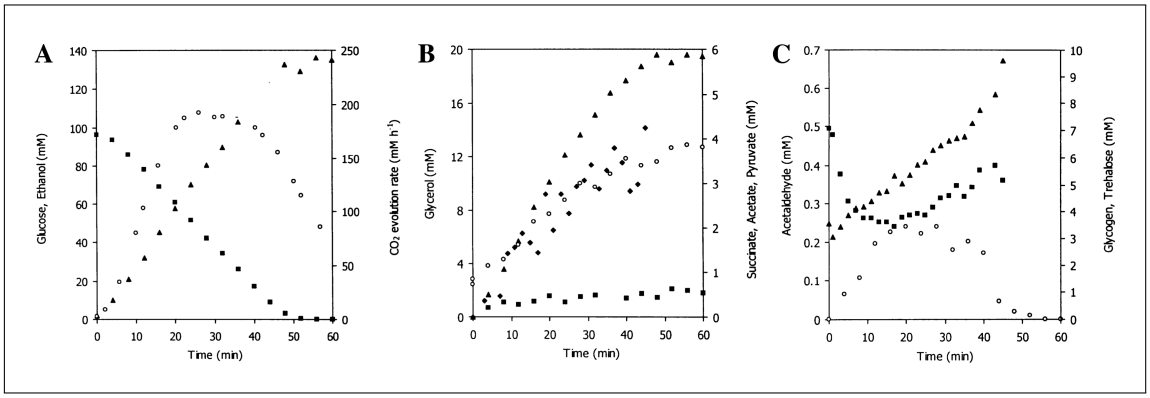


Figure 7.3 – The pathway reactivity tendency determined in vivo shows highest similarities with our agent-based simulations of long-range forces (Figure 7.2a) than with the deterministic simulation of the Smallbone2013 kinetic model (Figure 7.2d).
Note: External concentrations and CO2 flux during anaerobic glucose fermentation in resting yeast. **(A)** glucose (■), ethanol (▲) and CO2 evolution rate (○). **(B)** glycerol (▲), succinate (○), acetate (■) and pyruvate (◆). **(C)** glycogen (▲), trehalose (■) and acetaldehyde (○). Glycogen and trehalose are expressed in units of glucose. Adapted from Teusink, B. et al. "Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry." European Journal of Biochemistry 267, 5313-5329 (2000). Copyright by FEBS Press.

gence of global properties of a metabolic pathway; the results we obtained, even if preliminary, push us to speculate that these properties might be more faithful to the behaviours detectable in living cells than the system described a priori by a kinetic model.

7.5 Conclusions

In this work we provide a novel interpretation of the agent-based perception paradigm that allows us to study the long-distance interactions among biomolecules. Although other agent-based approaches have been proposed in the past for the study of molecular interactions, our model is specifically designed to analyse the effects of long and short range forces on the evolution of metabolic pathways.

The results provided in this chapter are preliminary and in some way contradictory with the results gained in vitro on the effect of long-range forces on biomolecules [62]. This is particularly true for the concentration plots we obtained simulating a system affected by the Debye screening, when compared with the same plots generated by simulating the long-distance interactions. Differently from our expectations, the two kinds of simulation produce similar outputs, suggesting a limited impact of the long-range forces on the glycolytic pathway. Only by reducing

the intermolecular interactions to Van der Waals distances, the metabolic pathway is unable to generate a significant amount of end products.

The simulation based on this model requires subsequent validation through a dedicated experimental study; at present, the investigation we carried out starting from the data obtained through the literature provided an interesting outcome, which show the potential of OrionV2 for carrying out in silico studies on molecular interactions.

Highlighting the importance of the long-range forces on the efficiency of the glycolytic process paves the way for future studies on pathologies and aging-related dysfunction affected by the rate of glucose oxidation [18, 59, 64, 81].

Further improvements of our approach may comprehend implementing the enzyme activation and inhibition, as well as running the simulations on a distributed computational environment; the latter would allow us to use real concentrations and extend the duration of the simulated intervals.

We might finally integrate our agent-based model with a more physically accurate description of the local interactions through systems of differential equations.

Modelling Metabolic Reactions on the Basis of the Interaction-as-perception Paradigm

8.1 Introduction

The core idea of this chapter is to analyse 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 [29, 57, 70, 71, 89]. This technique consists in building simplicial complexes, i.e. finite collections of objects, each of which could be seen as a n-bodies relation, and selecting the most meaningful one.

Weight Rank Clique Filtration and Persistent Homology are the two computational methods used to *map* simulation data into simplicial complexes, and to *visualise* the significant topological structures in the specific domain of metabolic reactions [14, 22, 67, 92].

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. In this way we can derive the graph of perceptions at a given step; on those graphs, we apply the topological data analysis to capture the 3-body interactions through the interpretation of 2-simplices as observable structures, which are convex hulls of three points. We use the 2-simplex formation as a valid semantic to represent the global dynamics of the system.

8.2 Integrative Methods for this Chapter

8.2.1 Multi-agent Modelling and Simulation

The investigation we are going to present is based on the spatial simulator for metabolic pathways we discussed in the previous chapters, taking the already mentioned "Smallbone2013 - Glycolysis in S.cerevisiae - Iteration 18" [80] model as source for the species concentrations and kinetic values.

The only reaction simulated for the aim of this study is the phosphorylation of glucose catalysed by hexokinase, which produces glucose 6-phosphate and ADP; the Smallbone2013 model takes into account the contribution of isoenzymes; therefore we considered the following three reactions:

$$\mathrm{GLC} + \mathrm{ATP} \xrightarrow{\mathrm{HKX1}} \mathrm{G6P} + \mathrm{ADP}$$
$$\mathrm{GLC} + \mathrm{ATP} \xrightarrow{\mathrm{HKX2}} \mathrm{G6P} + \mathrm{ADP}$$
$$\mathrm{GLC} + \mathrm{ATP} \xrightarrow{\mathrm{GLK1}} \mathrm{G6P} + \mathrm{ADP}$$

For such reactions, the Smallbone2013 model provides the experimental data summarised in Table 8.1.

8.2.2 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 some quantities, called *topological invariants*, that identify the space. In particular, those invariants can be thought as $n-$dimesional holes. Given a set of points (our data), a topological space is built over these points, whose elements are equipped with a notion of proximity that characterises a coordinate-free metric.

As we work in a discrete domain, the focus is on a topological spaces called simplicial complexes.

Simplicial complexes are made up by 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. Performing a filtration can be seen as wearing lenses for examining the dataset: different lenses consent to extract different kinds of information from the topological space; different filtrations give rise to different conversions

of the data points into simplicial complexes. In this paper, we use the Weight Rank Clique Filtration.

Weight Rank Clique Filtration

Weight Rank Clique Filtration (WRCF) is a particular filtration that is designed for operating on graphs: it allows us to build a simplicial complex starting 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 that can be in any dimension. To perform the Weight Rank Clique Filtration and the visualisation, we used a tool that is currently under development at the Bioshape and Data Science Lab of the University of Camerino. This tool exploits the Javaplex library for the computation of homology and the GraphSharp library for visualisation [83].

8.2.3 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.

Turning to the details, we generated, along with the standard output of the simulator (as described in Section 8.2.1), additional information about every interaction performed at each time step. In particular, we gain the identifier of all the molecules involved in such an interaction and the value of the related k_{cat}/K_m ratio. Basing 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 docked metabolite (so as 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 docked to the same enzyme.

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

- A *free enzyme* can make no perception (if there is no other compatible molecule in its surroundings) or just direct unstable perceptions.
- A *dual-complex* (formed when an enzyme binds one of the perceived metabolites) always carries out an inner fixed perception - of the enzyme for the docked 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 docked 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 k_{cat} value of the reaction (when it runs out, the enzyme returns free and two new metabolites are released in the simulation environment).

These three different behaviours identify the states of the automaton describing the cyclical pattern of an enzymatic reaction (see Chapter 6). As shown in Section 8.3, the iteration of this cycle drives the evolution of the reaction through phases of higher/lower stability, a property that we highlight through a quantitative analysis the topological representation (2-simplex) of intermolecular perceptions (see Figure 8.3.a). The 2-simplex topological structures provide a higher order global representation of interaction compared to that of a classical agent-based model. In the latter, each molecular interaction is 2-body, defined according to the biochemical reactions (like those shown in Section 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 8.1).

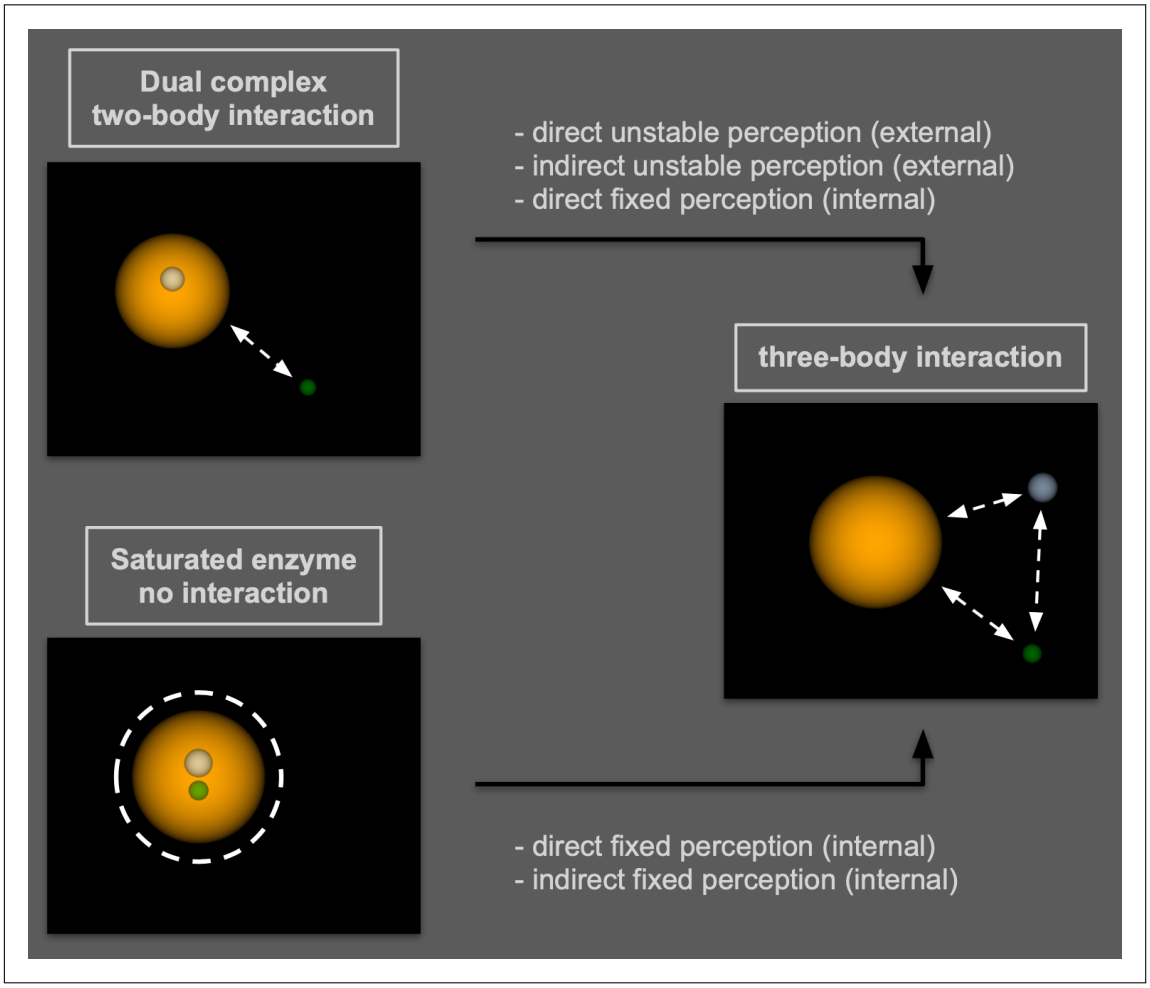


Figure 8.1 – Representation of the interaction-as-perception paradigm. In the classical agent-based model the interaction between a dual-complex and a complementary metabolite is 2-body; a saturated enzyme has no interactions at all. Conversely, through the interaction-as-perception paradigm they can both be interpreted as 3-body, since we take into account the potential interactions. However, to illustrate this paradigm basing 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.

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				
ID	$Conc.$	k_{cat}	K_{GLC}	K_{ATP}
GLC	6.28	/	/	/
ATP	4.29	/	/	/
ADP	1.29	/	/	/
G6P	0.77	/	/	/

Table 8.1 – Initial concentrations and kinetics parameters from Smallbone2013 model [80].

8.3 Results

By applying our multiagent simulation to study the metabolic reactions catalysed by hexokinase isomers (see Section 8.2.1 for details), we can observe how the molecules in the simulated environment move and interact at each time instant.

To analyse the dynamic evolution of each reaction from a topological point of view, we need to abstract from the standard spatial simulation output, basing on an interaction-as-perception paradigm. According to such an approach, an enzyme perceives a cognate metabolite whether a metabolite enters its interaction volumes or a docking between the two molecules actually happens.

The network of intermolecular perceptions (modelled on the basis of the above-described approach) can be interpreted in terms of simplicial complexes formation, where, every time an enzyme perceives a cognate metabolite, an edge among the two molecules is defined.

Changes in topological 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 do not perceive nor interact; therefore the topological environment is filled with sparse nodes (see Figure 8.2.a);
- 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 8.2.b), made by a central hub (the enzyme) connected to multiple nodes (metabolites);
- dockings between an enzyme and a single metabolite are caught in our representation by the formation of stable isolated 1-simplices composed by 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 by 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 8.2.c). Every triangle of this type is a potential stable link connecting the central complex and the opposite vertex;
- the potential condition described above is resolved when a fully saturated enzyme forms and can be identified by a stable 2-simplex; each final complex lingers in the simulation volume for a time given by the experimental value of the related k_{cat}, therefore, after such a delay, three new isolated nodes appear in place of a 2-simplex (Figure 8.2.d), i.e. the ones represented the enzyme and the products of the catalysed reaction.

Table 8.2 – Correlation between interaction-as-perception paradigm and topological 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

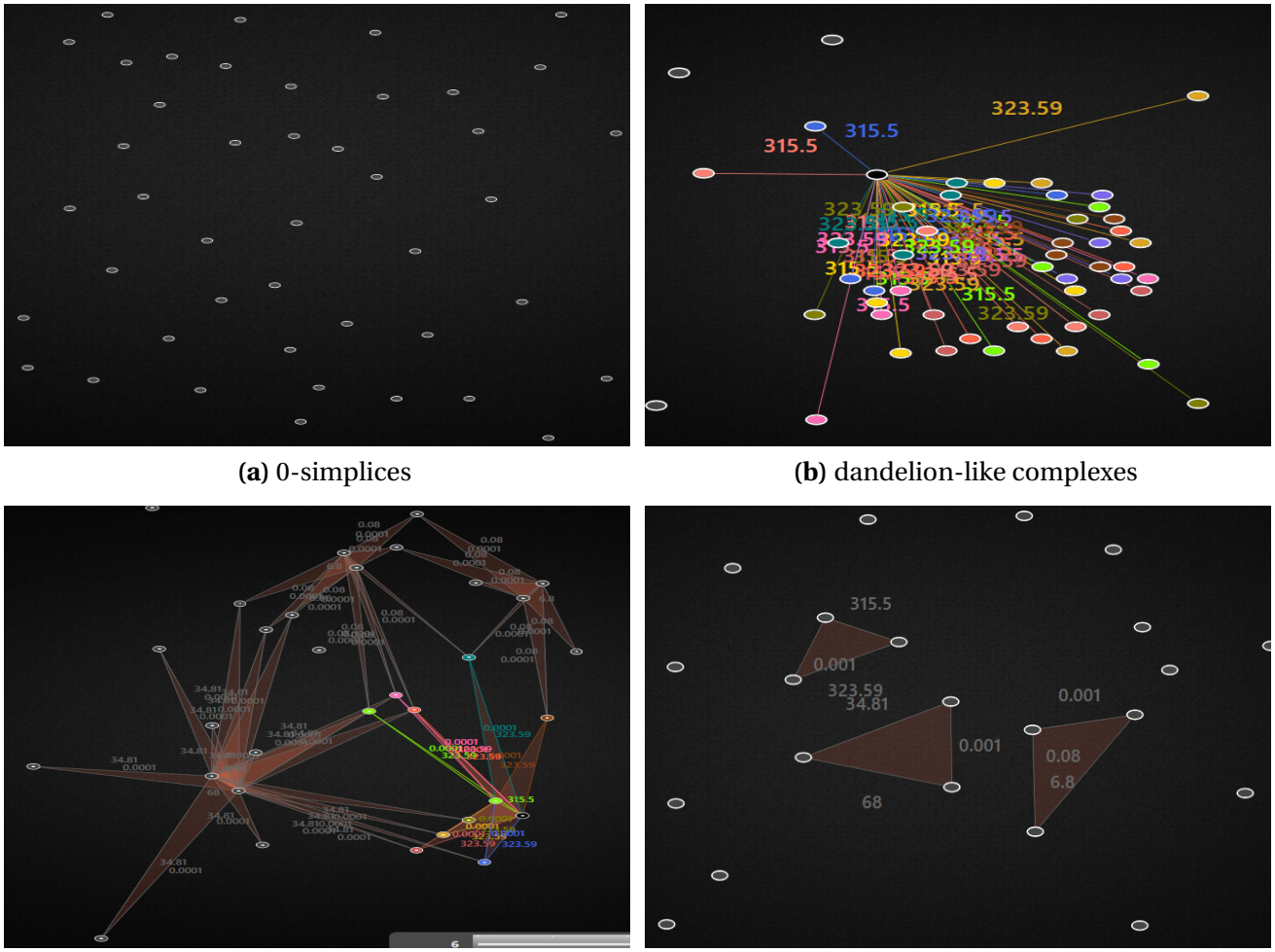
All the simplicial complexes we can observe during the time evolution of the simulation have a direct correlation with the perception-based structures described in Section 8.2. In Table 8.2 we summarise such relations by tracing each topological structure, identified in the previous description, back to the interaction-as-perception paradigm.

Representing through the above-described simplicial approach the dynamics of the multiagent simulation allows us to highlight some fundamental properties of metabolic reactions progression over time. Specifically, we can observe that changes in 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 instability of the system; therefore, we observe high concentrations of these structures during the reactive phases. As shown in Figure 8.3, considering a long temporal horizon, blocks of reactive phases are clearly distinguishable from the ones 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 a reactivity phase to another, in most cases identifiable by two opposite and overlapped spikes of the graph. 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 overall amount correlated to just one unit increase of stable 2-simplices.

As we can observe in Figure 8.3, such a behaviour determines a progressive decrease in 2-simplex



(a) 0-simplices

(b) dandelion-like complexes

(c) Four booklet-like complexes

(d) 2-simplices

Figure 8.2 – This figure shows the most significant structures we can identify through our topological analysis of the simulation. **(a)** 0-simplices representing all the molecules at the beginning of the simulation; **(b)** a "dandelion-like" structure made by a central node (enzyme) linked to the nodes representing the compatible substrate in its neighbourhood; **(c)** "booklet-like" structure composed by a central hub made by two linked nodes (corresponding to a "dual-complex" enzyme-metabolite) 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 identifiable 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 the arch itself.

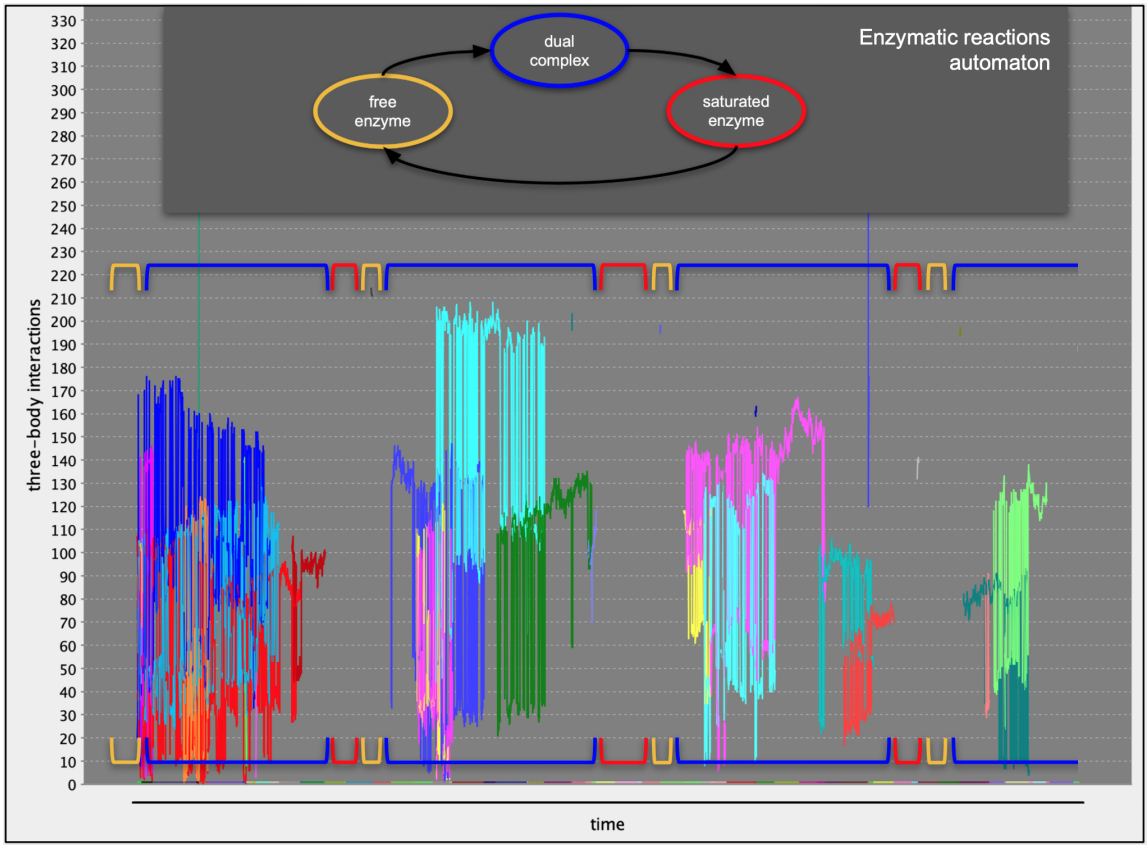


Figure 8.3 – Changes over time of the number of 2-simplices associated to each edge representing a dual complex; they are plotted along with the number of the stable 2-simplices (correlated to saturated enzymes). The aim of this plot is to provide a global view of how, on a long temporal horizon, highly 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.

stars amount, and therefore in block's reactivity, over time.

We also highlight that a transition between a stable phase and 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 delay elapsed, the product is released, and the enzyme starts to look for a new substrate, pushing the system towards a new reactive block.

In Section 8.2.3, 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 8.3.a).

8.4 Discussion

In the present work, we use a multiagent simulation to generate the dynamics of a complex system, while the Weight Rank Clique Filtration and Persistent Homology 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 topological 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 some insights, in terms of topological invariants, of what happens in the simulated systems.

Even if we do not claim to infer from these results any direct biological meaning, we hypothesise that both the above-mentioned patterns 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 the ones obtained by applying our visualisation method to other well-proven modelling approaches (e.g. based on PDE or SDE) or even directly to experimental data; it might allow us to identify how the simulated process differs from the one chosen as benchmark, and consequently make the necessary adjustments to make them fit.

8.5 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 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 emerging behaviour of a multiagent simulation with simplicial data analysis we have advanced the visualisation capabilities of the Orion simulator. The visualisation allowed us to identify the simplicial structures associated with the reaction space over time. This result might reveal to be a useful validation tool for the multiagent simulation itself. Indeed, it opens to the possibility of performing the same simplicial data analysis on empirically retrieved data, so as to verify the faithfulness of the simulation to the actual biological process [54].

At the same time, identifying patterns in the reactivity associated with molecular interactions graph might provide computational support for studying therapies based on drug targeting and enzyme inhibition [10, 12, 33, 85]. However, we want to point out that at the current stage of the study we are still validating and testing the proposed simulator.

As further developments, we are working on other validation approaches that could be combined with those mentioned above, and in particular, the ones involving innovative applications of formal methods in the analysis of biological processes [7, 50].

Conclusions

In the engineering life cycle for the simulation of a biological process we can identify two phases:

- process modelling and verification;
- system modelling, simulation and validation.

The starting point is the actual biological system [49], from which we derive an abstraction of the functions we aim to model and simulate. These functions are then formally modelled using process algebras (CCS in our case), and the properties of the models obtained verified through the best fitting method for model checking (for our models, we chose the Hennessy-Milner Logic). This phase is the one explored in the first part of this manuscript.

The successful use of process calculi to specify behavioural models allowed us to compare RNA and protein folding processes from a new perspective. In Chapter 3, we modelled the folding processes as behaviours resulting from the interactions that nucleotides and amino acids (the elementary units that compose RNAs and proteins respectively) perform on their linear sequences. This approach has been intended to provide new knowledge about the studied systems without strictly relying on empirical data. By applying Milner's CCS process algebra to highlight the distinguishing features of the two folding processes, we discovered an abstraction level at which they show behavioural equivalences. We believe that this result could be interpreted as a clue in favour of the highly-debated RNA World theory, according to which, in the early stages of cell evolution, RNA molecules played most of the functional and structural roles carried out today by proteins.

We also provided an algebraic approach for modelling the process that leads to the formation of misfolded proteins. In Chapter 4, a class of pathologies that affects these molecules has been treated as the resulting behaviour of their structural components, and used to study their dissimilar response to an alteration of the correct folding pathway. Our study started from the formal description of how such pathologies originate as an error of the genetic code (a mutation,

in biological terms) and can propagate through each step of the gene expression, affecting both RNA and protein structures. The Glu6Val mutation (which causes the sickle-cell anaemia disease) has been used as a case study and represented as a property of the folding models; the verification of its validity allowed us to describe, from an algebraic point of view, how the protein folding can be significantly affected by the alteration of even a single amino acid of the polypeptide sequence.

The algebraic approaches described so far have initially not be intended as a simulation-based tool, but a theoretical way to acquire new knowledge about the studied systems. We defined a new methodology to understand biological behaviours by analysing the complexity of the interactions characterising living systems. However, upon this models we were be able to define an algebraic specification for an actual simulation.

Indeed, in Chapter 5, we went one step further, by exploring the expressiveness of process algebras in modelling the functions representing the behaviour of non-coding RNA molecules. Basing on these results, we proposed a methodology suitable to generate an algebraic specification of a multiagent simulation. This approach was designed not only for theoretical purposes, but mostly to support the study of cellular processes and pathologies involving non-coding RNAs, by constructing agent-based models and validating hypotheses through model simulation. It might equally promote the development of future applications of non-coding RNA mediated inhibition of influenza infections.

Has a first step in the implementation of this specifications, we made preliminary studies to identify an agent-based approach fitting the requirements for simulating the molecular interactions. We found in Orion, a spatial simulator for metabolic pathways developed at the University of Camerino, the best solution for our purpose. However, since it was a prototypical project, it needed to be largely changed and improved in its functionalities.

This second phase of the engineering life cycle has been described in the Part II of this manuscript. It involved the definition of a low-level specification, the generation of the actual agent-based simulation and the validation of the results obtained, intended to make the agent-based model more faithful to biological system.

In Chapter 7, as a preliminary study in this direction, we adapted this agent-based simulator to study the effect of the long-distance electrodynamic interactions among biomolecules. We tested our approach by simulating the glycolytic pathway to observe the collective behaviour of the molecules involved in mutually connected reactions. The global properties emergent from the local molecular interactions, provided interesting, although controversial, outcomes. In contrast to the results of in vitro experiments [31, 62], we obtained similar results on the simulation of long-range forces and of those limited by the Debye screening effect. Only the simulation constrained by the Wan der Waals forces show a significant impact on the concentration changes of the metabolites. These observations push us to speculate that the long-distance electrodynamic interactions affect just the efficiency but not the effectiveness of the glycolytic process. However,

they also proved the capabilities of our agent-based simulator to effectively deal with molecular interactions in complex biological systems; we think it can be a suitable platform over which we may implement the algebraic models defined in the first part of the manuscript.

To validate the metabolic simulations made by a large number of molecules, in Chapter 8 we investigated the potentiality of the interaction-as-perception at the basis of the multiagent system. It tackles the complexity of visualising the emerging behaviour of a glycolytic pathway. We performed the topological data analysis of the molecular perceptions graphs gained during the formation of the enzymatic complexes to visualise the set of emerging patterns. Identifying specific patterns in terms of simplicial structures, allow us to characterise the reactions space over time and conceivably reveal the simulation reactivity trend.

This visualisation approach allowed us to identify the simplicial structures associated with the reaction space over time. A result that might reveal to be a useful validation tool for the multiagent simulation itself. Indeed, it opens to the possibility of performing the same simplicial data analysis on empirically retrieved data, so as to verify the faithfulness of the simulation to the actual biological process [54].

Appendices

Supplementary Information to Chapter 2

Process calculi may reveal the equivalence underlying RNA and proteins

A.1 Symbols and their transliteration

The following tables explain the symbols used to describe processes and actions of the proposed models; the transliterations of process names are necessary to construct the LTS representations as well as to perform the model checking and the bisimulation games through the automated tool CAAL - Concurrency Workbench, Alborg Edition [3].

Table 1.1: Processes

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

Table 1.2: Processes of the higher abstraction level

State\Process	Transliteration	Description
$\mathtt{C}$	C	amino acid carboxyl group
$\mathtt{B_{dr}}$	DR	double-ring base (purine)
$\Delta G_{\mathscr{F}^s}$	FSDG	folding step delta G
$\mathtt{N}$	N	amino acid amino group
$\mathcal{I}_n^h$	NHI	nucleotide hydrophobic interaction
$\Delta G_{\mathcal{I}_n^h}$	NHIDG	nucleotide hydrophobic interaction delta G
$\mathscr{F}_p^s$	PFS	protein folding step
$\mathscr{F}_{rna}^s$	RNAFS	RNA folding step
$\mathtt{O_{rna}}$	RNAO	RNA outside
$\mathtt{B_{sr}}$	SR	single-ring base (pyrimidine)
$\mathcal{P}_{aa3}$	TAAP	triple amino acid pairing
$\Delta G_{\mathcal{P}_{aa3}}$	TAAPDG	triple amino acid pairing delta G
$\mathtt{U_{aa3}}$	TAAU	triple amino acids unit
$\mathtt{U_{b3}}$	TBU	triple base unit

Table 1.3: Actions

Action	Description
`aa`	amino acid
`aa1fco`	first amino acid free carboxyl group
`aa1fnh`	first amino acid free amino group
`aa2fco`	second amino acid free carboxyl group
`aa2fnh`	second amino acid free amino group
`bb`	buriede 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`	negative delta G
`paa`	paired amino acids
`pdg`	positive delta G
`sb`	stacked bases
`sr`	single-ring base (pyrimidine)
`tpb`	three paried bases
`ub`	unpaired base
vdwi	van der Waals interaction
`zdg`	zero delta G

Table 1.4: Actions of the higher abstraction level

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

A.2 Models Construction

In the models of the folding process that we have defined, the weak interactions are classified in three main categories:

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

The hydrogen bond could be defined as an electrostatic interaction, but due to its distinctive properties and the fundamental role it carries out in the folding process, it has been represented separately.

All the weak interactions listed above have been modelled to formally describe the whole folding process. Each folding process always starts from a linear sequence (of nucleotides in RNAs and of amino acids in proteins) and is driven by the reduction in free energy between two different folded configurations.

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 weak interaction between two units of the sequence (equally for RNAs and proteins). In order for a folding step to take place, the weak interaction must cause a reduction in the free energy of the system, which means that the folding step must have a negative ΔG. The ΔG variation during folding is represented as a process that can produce three possible outputs: negative, a positive or zero ΔG.

A.2.1 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-conventional 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 a triple base pairing (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) has been modelled generalising that process as an interaction between a purine (adenine or guanine - labelled `dr`, since they are **d**ouble-**r**ing bases) and a pyrimidine (uracil and cytosine - **s**ingle-**r**ing bases and hence labelled `sr`) or between a two paired bases and a third base (also in this case, a generic purine or pyrimidine). The base pairing is symmetric, thus: `srdr` = `drsr`.

To remove some details not necessary for the aim of the model, it has also been opted for another generalisation, not explicitly representing all the possible interaction between a couple of paired

bases and a third base, but indicating this process as a "triple base-pairing" ($\mathcal{P}_{b3}$) 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 allowed in a base pair, in our models they must be at least two and at most three; the number of hydrogen bonds that link an unpaired base to a group of two already paired bases must be from one to three. It has been decided to limit the minimum number of hydrogen bonds in a base pair (to the number of two) because base pairs with a single hydrogen bond can be classified as a variant of the primary types and because the whole number of hydrogen bonds found in a base triplet is at least three.

Moreover, because up to now the only known base pair that involves three hydrogen bonds is the one between cytosine and guanine, only the `srdr` base pair is allowed in the model to form a triple hydrogen bond; this means that also AU, GU and CA base pairs could potentially form a triple hydrogen bond, which is a stretch of the current knowledge on hydrogen bonding. Since this property is important for the stability of the RNA molecules, we want to better justify the proposed abstraction: if we want to capture the constraint of limiting the formation of three hydrogen bonds only to the GC base pair, we should represent explicitly every bases and their combination instead of the convention adopted; this would reduce the readability of our models to capture a property that not affect the main purpose for which they were created.

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

$\mathcal{P}_{b2}$ is a sub-process of a general $\mathcal{F}^{s}_{rna}$ (*RNA Folding Step*) process, from which it receives its input (the $\mathcal{F}^{s}_{rna}$ process will be described later in this section); it is one of the possible sub-processes that give to each folding step its specificity. As explained in the article, 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 ($\mathcal{P}_{b3}$) takes as input a couple of bases, paired by the $\mathcal{P}_{b2}$ process, and a third unpaired base (`ub`) and provides as output a group of three paired bases (`tpb`). The number of hydrogen bonds that can be generated in this process is at least one and at most three.

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

The following is the specification of the $\mathcal{P}_{b2}$ and the $\mathcal{P}_{b3}$ processes using Milner's CCS (in the subsection A.2.4 on page 144 they will be contextualised in the complete description of the $\mathcal{F}^{s}_{rna}$ process):

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

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

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

In *proteins*, an 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. In the rest of this document, 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 hydrogen bond to each other 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 an hydrogen bond).

The *Amino Acids Pairing* process ($\mathcal{P}_{aa}$) is a subprocess of the general $\mathcal{F}^{s}_{p}$ (protein folding step), as $\mathcal{P}_{b2}$ is a subprocess of $\mathcal{F}^{s}_{rna}$.

$\mathcal{P}_{aa}$ takes two amino acids (`aa`) as input, makes an 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`); then, provides a group of two paired amino acids (`paa` - paired amino acids) 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 an hydrogen bond through one of its functional groups;
2. when the $\mathcal{P}_{aa}$ process receives two amino acids as input, we have the certainty that an hydrogen bond will form, because the negative ΔG of the interaction has already been checked in the early phases of the $\mathcal{F}_p^s$ process.

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

$$
\begin{aligned}
\mathcal{P}_{aa} &\stackrel{\text{def}}{=} \texttt{aa1fnh}.\texttt{NH}_{\texttt{aa1}} + \texttt{aa1fco}.\texttt{CO}_{\texttt{aa1}};\\
\texttt{NH}_{\texttt{aa1}} &\stackrel{\text{def}}{=} \texttt{aa2fco}.\texttt{CO}_{\texttt{aa2}};\\
\texttt{CO}_{\texttt{aa1}} &\stackrel{\text{def}}{=} \texttt{aa2fnh}.\texttt{NH}_{\texttt{aa2}};\\
\texttt{CO}_{\texttt{aa2}} &\stackrel{\text{def}}{=} \texttt{hb}.\mathcal{B}_{aa};\\
\texttt{NH}_{\texttt{aa2}} &\stackrel{\text{def}}{=} \texttt{hb}.\mathcal{B}_{aa};\\
\mathcal{B}_{aa} &\stackrel{\text{def}}{=} \overline{\texttt{paa}}.\mathcal{F}_p^s.
\end{aligned}
$$

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

A.2.2 Electrostatic interactions

Two particles electrically charged can interact according to the Coulomb's law. The model of the folding process does not investigate the interactions at atomic level, therefore the details of this law will not be covered. What we need to know is that two elementary units of either an RNA or a protein sequence, can interact in a folding step if they are both charged and if the ΔG of such interaction is negative. The main purpose of this kind of interactions 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 weak bond between two ions of opposite charge; the van der Waals interactions occur between two molecules oppositely polarised.

The modelling of these interactions is basically 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 a ionic interaction (*ii*) or a van der Waals interaction (*vdwi*), which are produced as output.

The *Bases Electrostatic Interaction* process ($\mathcal{I}_b^e$) specifies the electrostatic interactions in the RNA folding model:

$$\mathcal{I}_b^e \stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}_{rna}^s + \overline{vdwi}.\mathcal{F}_{rna}^s.$$

The *Amino Acids Electrostatic Interaction* process ($\mathcal{I}_{aa}^e$) specifies the electrostatic interactions in the protein folding model:

$$\mathcal{I}_{aa}^e \stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}_p^s + \overline{vdwi}.\mathcal{F}_p^s;$$

$\mathcal{I}_b^e$ is a subprocess of $\mathcal{F}_{rna}^s$; $\mathcal{I}_{aa}^e$ is a subprocess of $\mathcal{F}_p^s$.

A.2.3 Hydrophobic interactions

Water is a polar solvent, this 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 purine and pyrimidine bases are hydrophobic and relatively insoluble in water, while the backbone of alternating ribose and phosphate groups is hydrophilic.

To minimize contact of the bases with water and stabilizing the three-dimensional structure of the RNA, during the folding process, the backbone is placed on the outside of the molecule, facing the surrounding water, while the bases are positioned inside, stacked with the planes of their rings parallel to each other (a process called hydrophobic stacking interaction).

In the RNA folding model, the *Bases Hydrophobic Interaction* process ($\mathcal{I}_b^h$) takes two bases as input, produces an 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 $\mathcal{I}_b^h$ is a subprocess of $\mathcal{F}_{rna}^s$, the fact that the ΔG of the interaction is negative has already been checked in the earlier phases of the latter process.

The CSS specification of the $\mathcal{I}_b^h$ process is:

$$
\begin{aligned}
\mathcal{I}_b^h &\stackrel{\text{def}}{=} hbi.\mathtt{I}_{rna};\\
\mathtt{I}_{rna} &\stackrel{\text{def}}{=} \overline{\mathtt{bb}}.\mathcal{S};\\
\mathcal{S} &\stackrel{\text{def}}{=} \overline{\mathtt{sb}}.\mathcal{F}_{rna}^s.
\end{aligned}
$$

In *proteins*, the specific characteristics of an amino acid are determined by the properties of its R group; the polarity of that group varies widely, from non-polar 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. Folding of a polypeptide chain thus creates an "inside" and an "outside" and generates buried and exposed amino acid side chains. The interior of a protein is generally a densely packed core of hydrophobic amino acid side chains.

The hydrophobic interactions in proteins do not exhibit the stacking phenomenon, therefore the *Amino Acids Hydrophobic Interaction* process ($\mathcal{I}_{aa}^h$) takes only one amino acid as input. Then, if the amino acid side chain is hydrophilic (`hlsc`), it is exposed outside the protein (`esc`), if the side chain is hydrophobic (`hbsc`), it is buried inside the protein (`bsc`).

The inside and the outside of the protein are identified by the states $\mathtt{I}_\mathtt{p}$ and $\mathtt{O}_\mathtt{p}$ respectively. $\mathcal{I}_{aa}^h$ is a subprocess of $\mathcal{F}_p^s$.

The following is the CCS specification of the $\mathcal{I}_b^h$ process:

$$
\begin{aligned}
\mathcal{I}_{aa}^h &\stackrel{\text{def}}{=} \mathtt{hlsc}.\mathtt{O}_\mathtt{p} + \mathtt{hbsc}.\mathtt{I}_\mathtt{p};\\
\mathtt{O}_\mathtt{p} &\stackrel{\text{def}}{=} \overline{\mathtt{esc}}.\mathcal{F}_p^s;\\
\mathtt{I}_\mathtt{p} &\stackrel{\text{def}}{=} \overline{\mathtt{bsc}}.\mathcal{F}_p^s.
\end{aligned}
$$

A.2.4 Folding step

Now that we have described the model of each weak interaction in both RNA and protein, it is possible to contextualise these models in the folding step they belong to ($\mathcal{F}^s_{rna}$ or $\mathcal{F}^s_p$). Each step represents an iteration which allows the nondeterministic choice of one of the possible weak interaction subprocess. $\mathcal{F}^s_{rna}$ and $\mathcal{F}^s_p$ ensure that each subprocess complies with the specific restrictions on its input (according to the descriptions made above in this section) and that the interaction has a negative ΔG (and hence can be carried out).

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

$$\mathcal{F}^s_{rna} \stackrel{\text{def}}{=} \mathtt{ub}.\mathcal{I}1_n + \mathtt{ub}.\mathcal{I}2_n + \mathtt{srsr}.\mathcal{I}1_n + \mathtt{drdr}.\mathcal{I}1_n + \mathtt{srdr}.\mathcal{I}1_n + \mathtt{tpb}.\mathcal{I}1_n;$$

$$\mathcal{I}1_n \stackrel{\text{def}}{=} \mathtt{ub}.\Delta G_{\mathcal{I}^e_b} + \mathtt{srsr}.\Delta G_{\mathcal{I}^e_b} + \mathtt{drdr}.\Delta G_{\mathcal{I}^e_b} + \mathtt{srdr}.\Delta G_{\mathcal{I}^e_b} + \mathtt{tpb}.\Delta G_{\mathcal{I}^e_b};$$

$$\mathcal{I}2_n \stackrel{\text{def}}{=} \mathtt{ub}.\Delta G_{\mathcal{P}_{b2}} + \mathtt{ub}.\Delta G_{\mathcal{I}^h_b} + \mathtt{srsr}.\Delta G_{\mathcal{P}_{b3}} + \mathtt{drdr}.\Delta G_{\mathcal{P}_{b3}} + \mathtt{srdr}.\Delta G_{\mathcal{P}_{b3}};$$

$$\Delta G_{\mathcal{I}^e_b} \stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^e_b;$$

$$\Delta G_{\mathcal{I}^h_b} \stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^h_b;$$

$$\Delta G_{\mathcal{P}_{b2}} \stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{b2};$$

$$\Delta G_{\mathcal{P}_{b3}} \stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{b3};$$

$$\mathcal{P}_{b2} \stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}1_{b2};$$

$$\mathcal{B}1_{b2} \stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}2_{b2};$$

$$\mathcal{B}2_{b2} \stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}3_{b2} + \overline{\mathtt{srsr}}.\mathcal{F}^s_{rna} + \overline{\mathtt{drdr}}.\mathcal{F}^s_{rna} + \overline{\mathtt{srdr}}.\mathcal{F}^s_{rna};$$

$$\mathcal{B}3_{b2} \stackrel{\text{def}}{=} \overline{\mathtt{srdr}}.\mathcal{F}^s_{rna};$$

$$\mathcal{P}_{b3} \stackrel{\text{def}}{=} \mathtt{hb}.\mathcal{B}1_{b3};$$

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

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

$$\mathcal{B}3_{b3} \stackrel{\text{def}}{=} \overline{\mathtt{tpb}}.\mathcal{F}^s_{rna};$$

$$\mathcal{I}^e_b \stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}^s_{rna} + \overline{vdwi}.\mathcal{F}^s_{rna};$$

$$\mathcal{I}^h_b \stackrel{\text{def}}{=} hbi.\mathtt{I}_{rna};$$

$$\mathtt{I}_{rna} \stackrel{\text{def}}{=} \overline{\mathtt{bb}}.\mathcal{S};$$

$$\mathcal{S} \stackrel{\text{def}}{=} \overline{\mathtt{sb}}.\mathcal{F}^s_{rna}.$$

$\mathcal{I}1_n$ and $\mathcal{I}2_n$ (nucleotide interaction) are states that allow the selection the right subprocess on the basis of its permitted inputs.

$\Delta G_{\mathcal{P}_{b2}}$ (base pairing delta G), $\Delta G_{\mathcal{P}_{b3}}$ (triple base pairing delta G), $\Delta G_{\mathcal{I}^e_b}$ (bases electrostatic interaction delta G) and $\Delta G_{\mathcal{I}^h_b}$ (bases hydrophobic interaction delta G) processes check that the ΔG of the related interaction is negative.

The CCS specification of the whole $\mathcal{F}^s_p$ (Protein folding step) process is:

$$
\begin{aligned}
\mathcal{F}^s_p &\stackrel{\text{def}}{=} \mathtt{aa}.\mathcal{I}1_{aa} + \mathtt{aa}.\Delta G_{\mathcal{I}^h_{aa}};\\
\mathcal{I}1_{aa} &\stackrel{\text{def}}{=} \mathtt{aa}.\Delta G_{\mathcal{I}^e_{aa}} + \mathtt{aa}.\Delta G_{\mathcal{P}_{aa}};\\
\Delta G_{\mathcal{I}^e_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^e_{aa};\\
\Delta G_{\mathcal{I}^h_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^h_{aa};\\
\Delta G_{\mathcal{P}_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{aa};
\end{aligned}
$$

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

$$
\begin{aligned}
\mathcal{I}^e_{aa} &\stackrel{\text{def}}{=} \overline{ii}.\mathcal{F}^s_p + \overline{vdwi}.\mathcal{F}^s_p;\\
\mathcal{I}^h_{aa} &\stackrel{\text{def}}{=} \mathtt{hlsc}.\mathtt{O_p} + \mathtt{hbsc}.\mathtt{I_p};\\
\mathtt{O_p} &\stackrel{\text{def}}{=} \overline{\mathtt{ec}}.\mathcal{F}^s_p;\\
\mathtt{I_p} &\stackrel{\text{def}}{=} \overline{\mathtt{bc}}.\mathcal{F}^s_p.
\end{aligned}
$$

$\mathcal{I}1_{aa}$ is a state that allows the selection of the subprocesses that take two amino acids as input. $\Delta G_{\mathcal{P}_{aa}}$ (amino acids pairing delta G), $\Delta G_{\mathcal{I}^e_{aa}}$ (amino acids electrostatic interaction delta G) and $\Delta G_{\mathcal{I}^h_{aa}}$ (amino acids hydrophobic interaction delta G) processes check that the ΔG of the related interaction is negative.

A.2.5 RNA folding and protein folding

In order to meet the requirement that each interaction must have a negative ΔG, both the $\mathcal{F}^s_{rna}$ and $\mathcal{F}^s_p$ processes are placed in parallel composition with the $\Delta G_{\mathcal{F}^s}$ (folding step delta G) process, defining in this way the overall folding process ($\mathcal{F}_{rna}$ and $\mathcal{F}_p$ respectively).

$$\Delta G_{\mathcal{F}^s} \stackrel{\text{def}}{=} \overline{\mathtt{pdg}}.\Delta G_{\mathcal{F}^s} + \overline{\mathtt{ndg}}.\Delta G_{\mathcal{F}^s} + \overline{\mathtt{zdg}}.\Delta G_{\mathcal{F}^s};$$

$$\mathcal{F}_{rna} \stackrel{\text{def}}{=} (\mathcal{F}^s_{rna} | \Delta G_{\mathcal{F}^s}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\};$$

$$\mathcal{F}_p \stackrel{\text{def}}{=} (\mathcal{F}^s_p | \Delta G_{\mathcal{F}^s}) \setminus \{\mathtt{ndg}, \mathtt{pdg}, \mathtt{zdg}\}.$$

A.2.6 Model checking

It is possible to verify that the biochemical properties of the folding processes are satisfied by the above-described model. We propose here four examples, expressing some properties as HML formulas and establish if they are satisfied performing the model checking.

1. Two unpaired bases (ub) can form an hydrogen bond (hb) if the ΔG of the interaction is negative (ndg):
$$\mathcal{F}^s_{rna} \models \langle \mathtt{ub} \rangle \langle \mathtt{ub} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hb} \rangle tt;$$

2. with a single hydrogen bond it is not possible to form a base pair (srsr, drdr, srdr):
$$\mathcal{P}_{b2} \models \langle \mathtt{hb} \rangle ([\overline{\mathtt{srsr}}]\mathrm{ff} \text{ and } [\overline{\mathtt{srdr}}]\mathrm{ff} \text{ and } [\overline{\mathtt{drdr}}]ff);$$

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:
$$\mathcal{F}^s_{rna} \models \langle \mathtt{ub} \rangle \langle \mathtt{srsr} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hb} \rangle \langle \overline{\mathtt{tpb}} \rangle tt;$$

4. if an amino acid has an hydrophobic side chain (hbsc), it has to be buried inside (bsc) and not exposed outside (esc) the protein:

$$\mathcal{F}_p^s \models \langle \mathtt{aa} \rangle \langle \mathtt{ndg} \rangle \langle \mathtt{hbsc} \rangle (\langle \overline{\mathtt{bsc}} \rangle \mathrm{tt} \text{ and } [\overline{\mathtt{esc}}] f f);$$

The verification that these formulas are satisfied was made with the aid of the model checking function of the web-based tool CAAL. The results are shown in Figure A.1.

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

Figure A.1 – Verification of some biochemical properties, expressed as HML formulas, performed by the CAAL web-based tool. The checkmarks on the "Status" column indicate that all the formulas are satisfied.

A.2.7 Higher 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 it will be proved in this article, this level of abstraction can actually be defined. Its construction, however, requires a generalisation of the weak-interaction processes and the imposition of some limitations to the "expressiveness" of the protein folding process.

The first of the two aforementioned modification 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 property of each weak interaction, therefore the model is still faithful to the biological process. However they are not sufficient to obtain a behavioural equivalence between the folding processes of RNAs and proteins.

What we still need to do is limiting 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.

Base pairing process

$\mathcal{P}_{b2}$ takes two unpaired units (uu) as input (from the $\mathscr{F}^{s}_{rna}$ process) and produces a paired unit (pu) as output. *The label hb not indicates a single hydrogen bond, but stands for the overall interaction based on hydrogen bonding.*

$$\begin{aligned}
\mathcal{P}_{b2} &\stackrel{\text{def}}{=} hb.\mathtt{B_{sr}B_{sr}} + hb.\mathtt{B_{dr}B_{dr}} + hb.\mathtt{B_{sr}B_{dr}};\\
\mathtt{B_{sr}B_{sr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^{s}_{rna};\\
\mathtt{B_{dr}B_{dr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^{s}_{rna};\\
\mathtt{B_{sr}B_{dr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^{s}_{rna}.
\end{aligned}$$

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

Triple base pairing process

The $\mathcal{P}_{b3}$ process takes an unpaired unit (uu) and a paired unit (pu) as input (from the $\mathscr{F}^{s}_{rna}$ process) and produces a triple unit (tpu) as output.

$$\begin{aligned}
\mathcal{P}_{b3} &\stackrel{\text{def}}{=} hb.\mathtt{U_{b3}};\\
\mathtt{U_{b3}} &\stackrel{\text{def}}{=} \overline{\mathtt{tpu}}.\mathscr{F}^{s}_{rna}.
\end{aligned}$$

The state $\mathtt{U_{b3}}$ (triple base unit) indicates that an hydrogen bonding interaction (possibly made by more than one hydrogen bond) has taken place.

Amino acid pairing process

The $\mathcal{P}_{aa}$ process takes two unpaired units (uu) as input (from the $\mathscr{F}_p^s$ process) and produces a paired unit (pu) as output. As the same for the base pairing process, *the label hb not indicates a single hydrogen bond.*

$$\begin{aligned}
\mathcal{P}_{aa} &\stackrel{\text{def}}{=} hb.\mathtt{NC} + hb.\mathtt{CN};\\
\mathtt{NC} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}_p^s;\\
\mathtt{CN} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}_p^s.
\end{aligned}$$

The states NC and CN (where N and C stand for amino group and carboxyl group respectively) allow the preservation of the right complementarity of the hydrogen bond interaction between amino acids.

Triple amino acid pairing process

This is a new process (not present in the previous model); it is necessary to limit the capabilities of amino acids to hydrogen-bond with each other; as for the base pairing, at this level of abstraction at most three amino acids can be connected by the same hydrogen bonding interaction (not to be confused with a single hydrogen bond).

The $\mathcal{P}_{aa3}$ process takes an unpaired unit (uu) and a paired unit as input and produces a triple unit (tpu) as output.

$$\begin{aligned}
\mathcal{P}_{aa3} &\stackrel{\text{def}}{=} hb.\mathtt{U_{aa3}};\\
\mathtt{U_{aa3}} &\stackrel{\text{def}}{=} \overline{\mathtt{tpu}}.\mathscr{F}_p^s.
\end{aligned}$$

Electrostatic interaction

The base electrostatic interaction ($\mathcal{I}_b^e$) and the amino acid electrostatic interaction ($\mathcal{I}_{aa}^e$) processes are unchanged compared with the previous model (see Section A.2.2 on page 141).

Nucleotide hydrophobic interaction

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

The process, renamed $\mathcal{I}_n^h$, takes one unpaired unit as input and buries inside the RNA its hydrophobic component ($\mathtt{hbc}-\mathtt{bc}$) while exposes outside the RNA its hydrophilic component ($\mathtt{hlc}-\mathtt{ec}$).

$$\begin{aligned}
\mathcal{I}_n^h &\stackrel{\text{def}}{=} \mathtt{hlc}.\mathtt{O}_{\mathtt{rna}} + \mathtt{hbc}.\mathtt{I}_{rna} \\
\mathtt{O}_{\mathtt{rna}} &\stackrel{\text{def}}{=} \overline{\mathtt{ec}}.\mathscr{F}_{rna}^s; \\
\mathtt{I}_{rna} &\stackrel{\text{def}}{=} \overline{\mathtt{bc}}.\mathscr{F}_{rna}^s.
\end{aligned}$$

Amino acid hydrophobic interaction

The $\mathcal{I}_{aa}^h$ process takes one unpaired unit as input and buries inside the protein its hydrophobic component ($\mathtt{hbc}-\mathtt{bc}$) while exposes outside the protein its hydrophilic component ($\mathtt{hlc}-\mathtt{ec}$). In this case the "component" is a generalisation of the side chain, this means that each unpaired unit taken as input can have an hydrophobic or an hydrophilic component (but not both).

$$\begin{aligned}
\mathcal{I}_{aa}^h &\stackrel{\text{def}}{=} \mathtt{hlc}.\mathtt{O}_{\mathtt{p}} + \mathtt{hbc}.\mathtt{I}_{\mathtt{p}}; \\
\mathtt{O}_{\mathtt{p}} &\stackrel{\text{def}}{=} \overline{\mathtt{ec}}.\mathscr{F}_p^s; \\
\mathtt{I}_{\mathtt{p}} &\stackrel{\text{def}}{=} \overline{\mathtt{bc}}.\mathscr{F}_p^s.
\end{aligned}$$

Folding step

The $\mathscr{F}_{rna}^s$ and $\mathscr{F}_p^s$ perform the same tasks as in the previous model (see Section A.2.4 on page 144).

The CCS specification of the whole modified $\mathscr{F}^s_{rna}$ process is:

$$
\begin{aligned}
\mathscr{F}^s_{rna} &\stackrel{\text{def}}{=} \mathtt{uu}.\mathcal{I}1_n + \mathtt{pu}.\mathcal{I}1_n + \mathtt{uu}.\Delta G_{\mathcal{I}^h_n} + \mathtt{uu}.\mathcal{I}2_n + \mathtt{tpu}.\mathcal{I}1_n;\\
\mathcal{I}1_n &\stackrel{\text{def}}{=} \mathtt{uu}.\Delta G_{\mathcal{I}^e_b} + \mathtt{pu}.\Delta G_{\mathcal{I}^e_b} + \mathtt{tpu}.\Delta G_{\mathcal{I}^e_b};\\
\mathcal{I}2_n &\stackrel{\text{def}}{=} \mathtt{uu}.\Delta G_{\mathcal{P}_{b2}} + \mathtt{pu}.\Delta G_{\mathcal{P}_{b3}};\\
\Delta G_{\mathcal{I}^e_b} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^e_b;\\
\Delta G_{\mathcal{I}^h_n} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^h_n;\\
\Delta G_{\mathcal{P}_{b2}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{b2};\\
\Delta G_{\mathcal{P}_{b3}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{b3};
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}_{b2} &\stackrel{\text{def}}{=} hb.\mathtt{B_{sr}B_{sr}} + hb.\mathtt{B_{dr}B_{dr}} + hb.\mathtt{B_{sr}B_{dr}};\\
\mathtt{B_{sr}B_{sr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^s_{rna};\\
\mathtt{B_{dr}B_{dr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^s_{rna};\\
\mathtt{B_{sr}B_{dr}} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}^s_{rna};
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}_{aa3} &\stackrel{\text{def}}{=} hb.\mathtt{U_{aa3}};\\
\mathtt{U_{aa3}} &\stackrel{\text{def}}{=} \overline{\mathtt{tpu}}.\mathscr{F}^s_p;
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{I}^e_b &\stackrel{\text{def}}{=} \overline{ii}.\mathscr{F}^s_{rna} + \overline{vdwi}.\mathscr{F}^s_{rna};\\
\mathcal{I}^h_n &\stackrel{\text{def}}{=} \mathtt{hlc}.\mathtt{O_{rna}} + \mathtt{hbc}.\mathtt{I}_{rna};\\
\mathtt{O_{rna}} &\stackrel{\text{def}}{=} \overline{\mathtt{ec}}.\mathscr{F}^s_{rna};\\
\mathtt{I}_{rna} &\stackrel{\text{def}}{=} \overline{\mathtt{bc}}.\mathscr{F}^s_{rna}.
\end{aligned}
$$

$\mathcal{I}1_n$ and $\mathcal{I}2_n$ (nucleotide interaction) are states that allow the selection the right subprocess on the basis of its permitted inputs.

$\Delta G_{\mathcal{P}_{b2}}$ (base pairing delta G), $\Delta G_{\mathcal{P}_{b3}}$ (triple base pairing delta G), $\Delta G_{\mathcal{I}_b^e}$ (bases electrostatic interaction delta G) and $\Delta G_{\mathcal{I}_n^h}$ (nucleotide hydrophobic interaction delta G) processes check if the ΔG of the related interaction is negative.

The CCS specification of the whole modified $\mathscr{F}_p^s$ process is the following:

$$
\begin{aligned}
\mathscr{F}_p^s &\stackrel{\text{def}}{=} \mathtt{uu}.\mathcal{I}1_{aa} + \mathtt{pu}.\mathcal{I}1_{aa} + \mathtt{uu}.\Delta G_{\mathcal{I}^h_{aa}} + \mathtt{uu}.\mathcal{I}2_{aa} + \mathtt{tpu}.\mathcal{I}1_{aa};\\
\mathcal{I}1_{aa} &\stackrel{\text{def}}{=} \mathtt{uu}.\Delta G_{\mathcal{I}^e_{aa}} + \mathtt{pu}.\Delta G_{\mathcal{I}^e_{aa}} + \mathtt{tpu}.\Delta G_{\mathcal{I}^e_{aa}};\\
\mathcal{I}2_{aa} &\stackrel{\text{def}}{=} \mathtt{uu}.\Delta G_{\mathcal{P}_{aa}} + \mathtt{pu}.\Delta G_{\mathcal{P}_{aa3}};\\
\Delta G_{\mathcal{I}^e_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^e_{aa};\\
\Delta G_{\mathcal{I}^h_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{I}^h_{aa};\\
\Delta G_{\mathcal{P}_{aa}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{aa};\\
\Delta G_{\mathcal{P}_{aa3}} &\stackrel{\text{def}}{=} \mathtt{ndg}.\mathcal{P}_{aa3};\\
\\
\mathcal{P}_{aa} &\stackrel{\text{def}}{=} hb.\mathtt{NC} + hb.\mathtt{CN};\\
\mathtt{NC} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}_p^s;\\
\mathtt{CN} &\stackrel{\text{def}}{=} \overline{\mathtt{pu}}.\mathscr{F}_p^s;\\
\\
\mathcal{P}_{aa3} &\stackrel{\text{def}}{=} hb.\mathtt{U_{aa3}};\\
\mathtt{U_{aa3}} &\stackrel{\text{def}}{=} \overline{\mathtt{tpu}}.\mathscr{F}_p^s;\\
\\
\mathcal{I}^e_{aa} &\stackrel{\text{def}}{=} \overline{ii}.\mathscr{F}_p^s + \overline{vdwi}.\mathscr{F}_p^s;\\
\mathcal{I}^h_{aa} &\stackrel{\text{def}}{=} \mathtt{hlc}.\mathtt{O_p} + \mathtt{hbc}.\mathtt{I_p};\\
\mathtt{O_p} &\stackrel{\text{def}}{=} \overline{\mathtt{ec}}.\mathscr{F}_p^s;\\
\mathtt{I_p} &\stackrel{\text{def}}{=} \overline{\mathtt{bc}}.\mathscr{F}_p^s.
\end{aligned}
$$

$\mathcal{I}1_{aa}$ and $\mathcal{I}2_{aa}$ are states that allow the selection of the right subprocess on the basis of its permitted inputs. $\Delta G_{\mathcal{P}_{aa}}$ (amino acids pairing delta G), $\Delta G_{\mathcal{P}_{aa3}}$ (triple amino acids pairing delta

G), $\Delta G_{J^e_{aa}}$ (amino acids electrostatic interaction delta G) and $\Delta G_{J^h_{aa}}$ (amino acids hydrophobic interaction delta G) processes check if the ΔG of the related interaction is negative.

The folding processes are still defined as the parallel composition of the folding step process and the folding step ΔG (see Section A.2.5 on page 147).

Supplementary Information to Chapter 4

An Algebraic Approach to the Study of Protein Misfolding

B.1 Gene Expression Model

The first step in the description of the gene expression model is to define the set of nucleotides; they are the elementary units of both DNA and RNA (with some biochemical differences not relevant for the aim of this model) and are identified by the bases they contain.

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

where each letter stands for adenine, thymine, cytosine, guanine and uracil respectively.

DNA should not contain the uracil base, while RNA does not contain the thymine; therefore it is useful to define two subsets of $\mathcal{N}$ as follows:

$$\mathcal{N}_{dna} = \{\mathtt{a}, \mathtt{t}, \mathtt{c}, \mathtt{g}\}$$

$$\mathcal{N}_{rna} = \{\mathtt{a}, \mathtt{u}, \mathtt{c}, \mathtt{g}\}$$

The expression of the DNA sequence of a gene flows through three main processes: **transcription**, **RNA processing** and **translation**.

As shown in these Supplementary Information, it is possible to define DNAs and RNAs (and hence genes) as strings of nucleotides while proteins as strings of amino acids.

The three above-mentioned processes can therefore be imagined as functions on strings: the final product of the gene expression will be the result of the composition of these functions.

To outline this idea we can define

- the transcription as the *tsc* function, such that

$$RNA = tsc(gene);$$

- the RNA processing as the *prc* function, such that

$$mRNA = prc(RNA);$$

- the translation as the *tsl* function, such that

$$protein = tsl(mRNA).$$

This means that the overall gene expression process
($GeneExp$) can be defined as:

$$GeneExp(gene) = (tsl \circ prc \circ tsc)(gene) = protein$$

During the transcription process, the sequence of nucleotides taken as template to produce an RNA molecule (transcript) is read from the complementary strand of the actual coding strand (for a specific gene). This is due to the base pairing process, which characterises almost every step of the gene expression (and hence the transcription).

B.1.1 Transcription process

The transcription process uses the DNA sequence of a gene as template to produce an RNA molecule (the transcript). Each gene codes for a specific protein or a functional RNA molecule, therefore the transcript must contain a copy of such definite information. The process is mainly carried out by the RNA polymerase (RNApol).

As already said, the transcription relies on the complementary base pairing principle; this means that, to transcribe a gene placed in one of the two strands of a DNA double-helix (the coding strand), the RNA polymerase must take its complementary strand as template. From a strictly informational point of view, we can say that the RNA polymerase reads the product of a replication process.

However, differently from the DNA replication, the resulting sequence is not simply the complement of the source strand. Biochemically, an RNA strands differs in various aspects from a DNA strand, but the property to which we will focus is the presence of a uracil (`u`) instead of a thymine (`t`).

We can formally define an RNA sequence as a string of elements of the above-defined NUC set; therefore the set of all possible RNA sequences is the set $\mathscr{R}$ defined as follows:

$$\mathscr{R} = \{n_1 n_2 \dots n_k \mid n_i \in \mathscr{N}_{rna},\ i \in \{1, \dots, k\}\},$$

where n stands for *ribonucleotide*.

This means that, each time the RNA polymerase reads a `a` on the template strand, it will "write" a `u` on the transcript (but, excluding the case of a mispairing, `g` always pair with `c`).

The process starts from a *promoter* sequence and proceeds until it reaches a *terminator* sequence (working on one nucleotide at time). Therefore, being $\omega_p, \omega_t \in \mathscr{N}^+_{dna}$ the strings representing the *promoter* and the *terminator* sequences respectively, we can define the genes set ($\mathscr{G}$) as a subset of the $\mathscr{D}$ set such that:

$$\mathscr{G} = \{\omega_p\ \omega_g\ \omega_t \mid \omega_p\ , \omega_t \in \mathscr{N}^+_{dna},\ \omega_g \in \mathscr{N}^*_{dna}\}$$

The transcription process also provides a proofreading function, and, similarly to the one performed by the DNA polymerase, it can fail in the detection of a mispairing.

In the model, the RNA polymerase process (RNAPOL), takes as input the base of a nucleotide (`a`, `t`, `c` or `g`) and produces as output the association of such base with its complementary ribonucleotide (`au`, `ta`, `cg`, `gc`). When a mispairing occurs, a purine (`a`, `g`) is associated with the wrong pyrimidine (`u`, `c`) - or vice-versa (remembering that in the template strand is present a `t` instead of a `u`). The output in these cases will be one of the following base pairs: `ac`, `ca`, `gu`, `tg`.

The proofreading process (*PROOFREAD2* [1]) takes the base pairs produced as output by the RNAPOL process and provides the correct nucleotide that has to be added to the transcript. The proofread process can make mistakes and leave a mispairing uncorrected(see the activity diagram on Figure B.1 on the following page).

[1] The number is used to distinguish this proofread process from the one of the DNA polymerase.

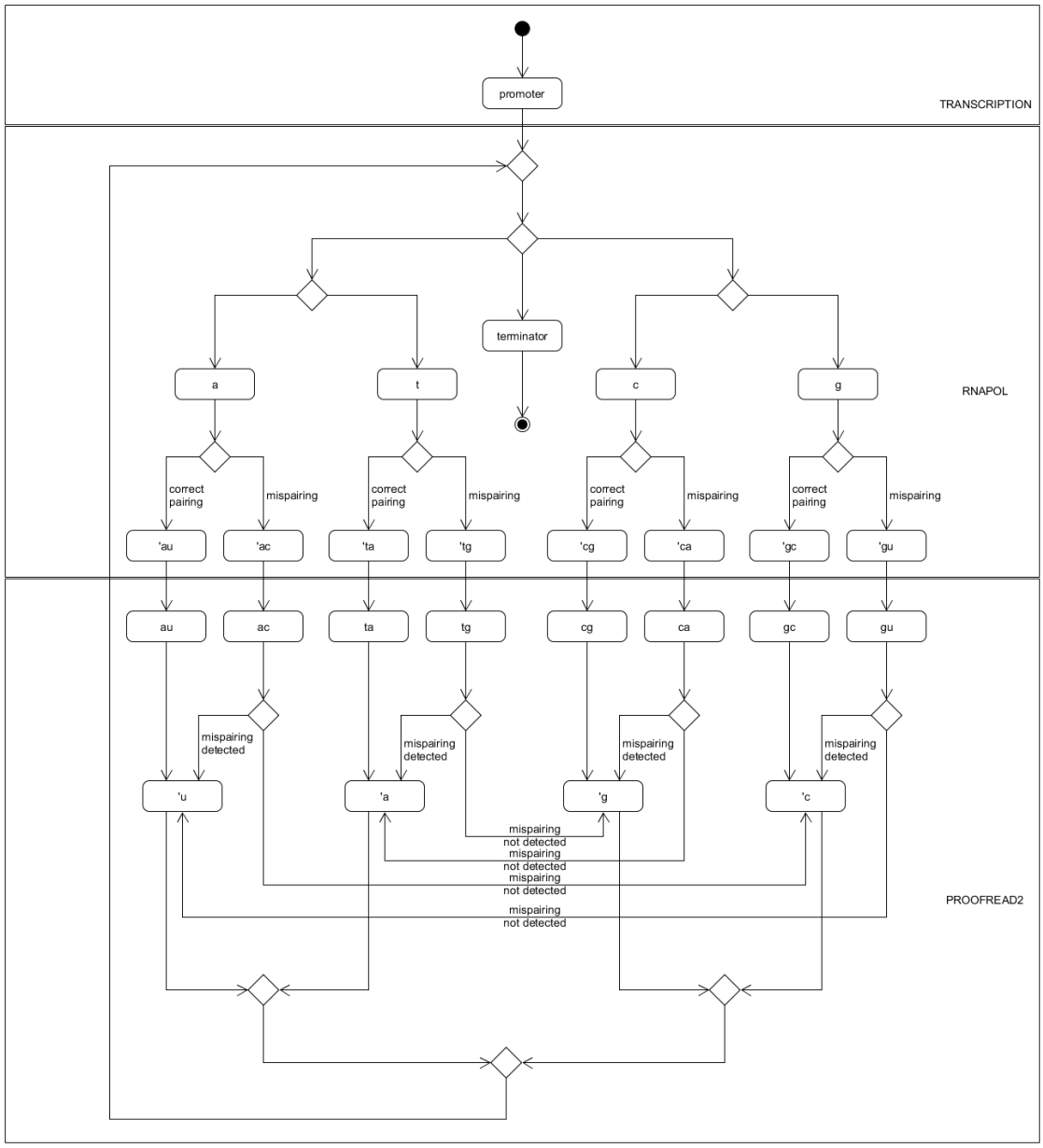


Figure B.1 – Activity diagram of the TRANSCRIPTION process (with the two subprocesses RNAPOL and PROOFREAD2).

The following is the specification of the TRANSCRIPTION process using Milner's CCS:

$$TRANSCRIPTION \stackrel{\text{def}}{=} promoter.RNAPOL;$$

$$RNAPOL \stackrel{\text{def}}{=} \mathtt{a}.A3 + \mathtt{t}.T3 + \mathtt{c}.C3 + \mathtt{g}.G3 + terminator.0;$$

$$A3 \stackrel{\text{def}}{=} \overline{\mathtt{au}}.PROOFREAD2 + \overline{\mathtt{ac}}.PROOFREAD2;$$

$$T3 \stackrel{\text{def}}{=} \overline{\mathtt{ta}}.PROOFREAD2 + \overline{\mathtt{tg}}.PROOFREAD2;$$

$$C3 \stackrel{\text{def}}{=} \overline{\mathtt{cg}}.PROOFREAD2 + \overline{\mathtt{ca}}.PROOFREAD2;$$

$$G3 \stackrel{\text{def}}{=} \overline{\mathtt{gc}}.PROOFREAD2 + \overline{\mathtt{gu}}.PROOFREAD2;$$

$$PROOFREAD2 \stackrel{\text{def}}{=} \mathtt{au}.U4 + \mathtt{ac}.U4 + \mathtt{ta}.A4 + \mathtt{tg}.A4 + \mathtt{cg}.G4 + \mathtt{ca}.G4 + \mathtt{gc}.C4 + \mathtt{gu}.C4 +$$
$$\mathtt{ac}.C4 + \mathtt{tg}.G4 + \mathtt{ca}.A4 + \mathtt{gu}.U4;$$

$$A4 \stackrel{\text{def}}{=} \overline{\mathtt{a}}.RNAPOL;$$

$$U4 \stackrel{\text{def}}{=} \overline{\mathtt{u}}.RNAPOL;$$

$$C4 \stackrel{\text{def}}{=} \overline{\mathtt{c}}.RNAPOL;$$

$$G4 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.RNAPOL;$$

Given *B* as one of the possible bases (*A, T, C, G* and *U*),

- each *B3* state describes the behaviour of the *RNAPOL* process when takes the corresponding base as input: this behaviour is defined by the nondeterministic choice between the correct and the wobble base pairing;

- each *B4* state describes which output the proofread process will provide basing on the choice made in the *PROOFREAD2* state; in the first row are indicated the correct choices (including the error corrections), while in the second row the wrong choices (hence the cases in which the proofreading process does not recognise a mispairing).

The LTS of the TRANSCRIPTION process can be seen in Figure B.2.

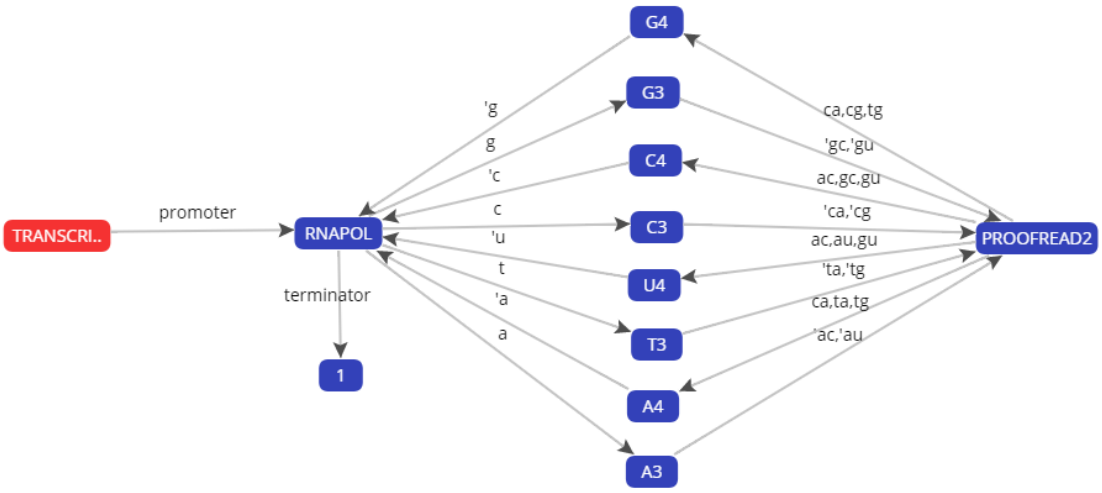


Figure B.2 – Labelled Transition System of the *TRANSCRIPTION* process generated with the CAAL web-based tool.

B.1.2 RNA Processing

The transcript can be an intermediary in the synthesis of a protein (in this case it is called mRNA - messger RNA) or be itself the final product of the gene expression (often called functional RNA or simply non-coding RNA).

However, before an RNA molecule can be considered "mature" (and hence carry out its purpose), it must undergo different **RNA processing steps**. The transcripts are processed in various ways, depending on their type.

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

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

A third step, common to all type of RNA, is called **RNA splicing** and performs the removal of the noncoding intervening sequence (**introns**) from the ribonucleotide chain of the transcript; as the result of this process, the transcript is converted in a uninterrupted sequence of **exons** (the coding portions of an eukaryotic gene).

The actual splicing process involves a complex molecular machinery called spliceosome, but

what we need to know to understand how the informational content changes in this process is only that a set of subsequences of the whole RNA sequence are subtracted from the latter. These subsequences are identified by a special opening and closing sequence of nucleotides (gu and ag respectively - but not all the gu and ag groups represent the starting and the ending point of an intron).

Therefore, the $\mathscr{I}$ set of all the introns is a subset of $\mathscr{R}$ such that:

$$\mathscr{I} = \{\mathtt{gu}\ n_1 n_2 \ldots n_k\ \mathtt{ag}\ | n_i \in \mathscr{N}_{rna},\ \ i \in \{1, \ldots, k\}\}$$

The model of the RNA processing (described with the process *PROCESSING*) allows to chose the set of subprocesses specific for the mRNA (hence *CAPPING* - *MRNASPLICING* - *POLIAD*) or the single *SPLICING* process (*SPLICING*), to which undergo the non-coding RNAs.

The *CAPPING* and *POLIAD* processes simply add a cap to the $5'$ end (*fpend*) and a poly-A tail (*polyatail*) to the $3'$ end (*tpend*) of the RNA sequence respectively.

The *SPLICING* and the *MRNASPLICING* processes have basically the same behaviour, with the only differences that the *MRNASPLICING* process comes necessarily after the *CAPPING* process and, when it reaches the $3'$ end, it is followed by the *POLIAD* porcess, while the *SPLICING* process ends the whole RNA processing when it stops at the $3'$ extremity of the RNA sequence.

To perform the removal of the introns, the *SPLICING* and the *MRNASPLICING* processes read the RNA sequence one nucleotide at time and produce it as output until they reaches a *gu* sequence that is the start point of an intron. From this point, they simply read the following nucleotides, producing no output, until they find the *ag* sequence that signals the end of the intron. These two phases are alternated until they get to the $3'$ end (*threepend*) of the RNA.

The CCS specification of the *PROCESSING* process is:

$$PROCESSING \stackrel{\text{def}}{=} fpend.SPLICING + fpend.CAPPING;$$

$$CAPPING \stackrel{\text{def}}{=} \overline{cap}.MRNASPLICING;$$

$$SPLICING \stackrel{\text{def}}{=} \mathtt{u}.U5 + \mathtt{c}.C5 + \mathtt{a}.A5 + \mathtt{g}.G5 + \mathtt{g}.INTRONSTART1 + tpend.0;$$

$$A5 \stackrel{\text{def}}{=} \overline{\mathtt{a}}.SPLICING;$$

$$U5 \stackrel{\text{def}}{=} \overline{\mathtt{u}}.SPLICING;$$

$$C5 \stackrel{\text{def}}{=} \overline{\mathtt{c}}.SPLICING;$$

$$G5 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.SPLICING;$$

$$INTRONSTART1 \stackrel{\text{def}}{=} \mathtt{u}.CUT1 + \mathtt{u}.U6 + \mathtt{c}.C6 + \mathtt{a}.A6 + \mathtt{g}.G6;$$

$$A6 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.A5;$$

$$U6 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.U5;$$

$$C6 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.C5;$$

$$G6 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.G5;$$

$$CUT1 \stackrel{\text{def}}{=} \mathtt{u}.CUT1 + \mathtt{c}.CUT1 + \mathtt{a}.CUT1 + \mathtt{a}.INTRONEND1 + \mathtt{g}.CUT1;$$

$$INTRONEND1 \stackrel{\text{def}}{=} \mathtt{g}.SPLICING + \mathtt{g}.CUT1 + \mathtt{u}.CUT1 + \mathtt{c}.CUT1 + \mathtt{a}.CUT1;$$

$$MRNASPLICING \stackrel{\text{def}}{=} \mathtt{u}.U7 + \mathtt{c}.C7 + \mathtt{a}.A7 + \mathtt{g}.G7 + \mathtt{g}.INTRONSTART2 + tpend.POLIAD;$$

$$A7 \stackrel{\text{def}}{=} \overline{\mathtt{a}}.MRNASPLICING;$$

$$U7 \stackrel{\text{def}}{=} \overline{\mathtt{u}}.MRNASPLICING;$$

$$C7 \stackrel{\text{def}}{=} \overline{\mathtt{c}}.MRNASPLICING;$$

$$G7 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.MRNASPLICING;$$

$$INTRONSTART2 \stackrel{\text{def}}{=} \mathtt{u}.CUT2 + \mathtt{u}.U8 + \mathtt{c}.C8 + \mathtt{a}.A8 + \mathtt{g}.G8;$$

$$A8 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.A7;$$

$$U8 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.U7;$$

$$C8 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.C7;$$

$$G8 \stackrel{\text{def}}{=} \overline{\mathtt{g}}.G7;$$

$$CUT2 \stackrel{\text{def}}{=} \mathtt{u}.CUT2 + \mathtt{c}.CUT2 + \mathtt{a}.CUT2 + \mathtt{a}.INTRONEND2 + \mathtt{g}.CUT2;$$

$$INTRONEND2 \stackrel{\text{def}}{=} \mathtt{g}.MRNASPLICING + \mathtt{g}.CUT2 + \mathtt{u}.CUT2 + \mathtt{c}.CUT2 + \mathtt{a}.CUT2;$$

$$POLIAD \stackrel{\text{def}}{=} \overline{polyatail}.0;$$

The states *INTRONSTART* check if a g (read in the RNA sequence) is followed by a u (and hence can be the possible start of an intron); the states *INTRONEND* check if a a is followed by a g (and therefore could signal the end of the intron).

The states *CUT* allow to read the nucleotides of an intron without producing them as output.

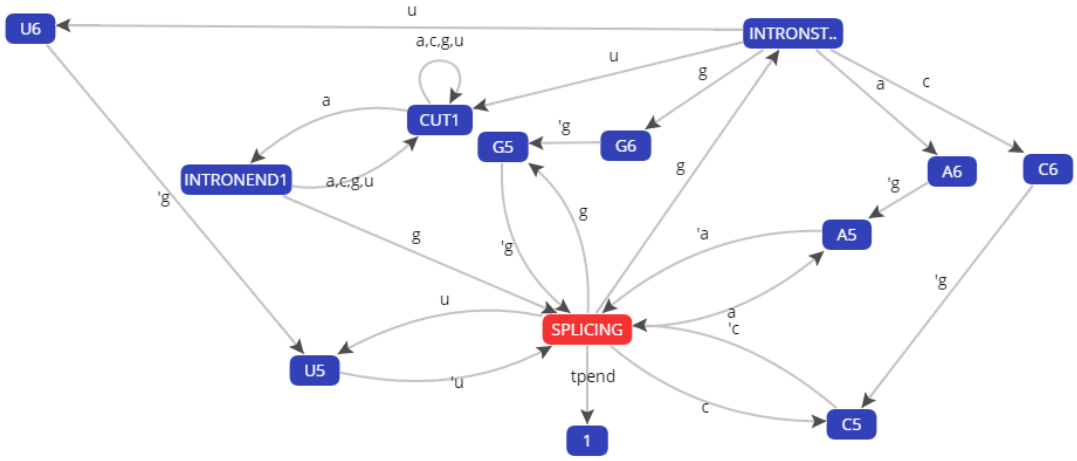


Figure B.3 – Labelled Transition System of the SPLICING process (and all its subprocesses) generated with the CAAL web-based tool.

Each of the above described states are followed by a 1 or a 2 number to distinguish if they are subprocesses of the SPLICING or the *MRNASPLICING* process, but processes with the same name perform the same task.

Given *B* as one of the possible bases (*A*, *C*, *G* and *U*),

- the states *B5* and *B7* indicate that the corresponding base will be provided as output by the *SPLICING* and the *MRNASPLICING* processes respectively;
- the states *B6* and *B8* allow to output a g when it is not followed by a *u* (and therefore it certainly isn't part of an intron).

The LTS of the *SPLICING* process is shown in Figure B.3.

A mutation can be removed during the splicing process only if it is placed, by chance, inside an intron; otherwise it can pass through this phase of the gene expression without alteration.

B.1.3 Translation process

The translation process converts the information contained in the nucleotide sequence of a mature mRNA into the amino acid sequence of a protein. This process is performed by a molecular complex called **ribosome**. As with the previous phases, we will focus only on the informational aspects of this process.

	aga									uua				agc						
	agg													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

Table B.1 – The genetic code table represents the association of each of the 20 amino acids with the related group of codons.

The **genetic code** consist in the association of each of the 20 amino acids with a sequence of three nucleotides (called **codon**). The number of triples over the set of four nucleotides (a, u, c, g) is 64, hence each amino acid can be codified by more than one codon. The association is made basing on the table in Table B.1.

The set of all the possible codons $\mathscr{C}$ is a subset of the triples in $\mathscr{R}$:

$$\mathscr{C} = \{\omega_c \mid \omega_c \in \mathscr{N}^+_{rna},\ |\omega_c| = 3\};$$
$$\mathscr{C} \subset \mathscr{R}.$$

We can also provide the set of all the possible amino acids that we can find in a protein:

$$\mathscr{A} = \{\texttt{ala}, \texttt{arg}, \texttt{asp}, \texttt{asn}, \texttt{cys}, \texttt{glu}, \texttt{gln}, \texttt{gly}, \texttt{his}, \texttt{ile},$$
$$\texttt{leu}, \texttt{lys}, \texttt{met}, \texttt{phe}, \texttt{pro}, \texttt{ser}, \texttt{thr}, \texttt{trp}, \texttt{tyr}, \texttt{val}\}$$

Therefore, it is possible define the proteins as the set $\mathscr{P}$ of the strings on the $\mathscr{A}$ set:

$$\mathscr{P} = \{a_1 a_2 \ldots a_k \mid a_i \in \mathscr{A},\ i \in \{1, \ldots, k\}\}$$

The translation process begins from a *start codon* (aug - which also codes for the Methionine amino acid) and terminates when the ribosome reaches one of the three possible *stop codons* (uaa, uag, uga).

In the model, the ribosome is represented as a process (*RIBOSOME*) which scans the mrna sequence one nucleotide at time, starting from the *cap* and looking for a aug codon. When it finds the start codon the ribosome begins to produce as output an amino acid for each codon it

reads until it reaches a stop codon.

The following is the Milner's CCS specification of the TRANSLATION process:

$$TRANSLATION \stackrel{\text{def}}{=} cap.RIBOSOME;$$

$$RIBOSOME \stackrel{\text{def}}{=} \mathrm{u}.RIBOSOME + \mathrm{c}.RIBOSOME + \mathrm{a}.STARTCODON1 + \mathrm{g}.RIBOSOME;$$

$$STARTCODON1 \stackrel{\text{def}}{=} \mathrm{u}.STARTCODON2 + \mathrm{c}.RIBOSOME + \mathrm{a}.RIBOSOME + \mathrm{g}.RIBOSOME;$$

$$STARTCODON2 \stackrel{\text{def}}{=} \mathrm{u}.RIBOSOME + \mathrm{c}.RIBOSOME + \mathrm{a}.RIBOSOME + \mathrm{g}.START;$$

$$START \stackrel{\text{def}}{=} \overline{met}.DECODE;$$

$$DECODE \stackrel{\text{def}}{=}$$

$$\mathrm{u}.(\mathrm{u}.(\mathrm{u}.PHE + \mathrm{c}.PHE + \mathrm{a}.LEU + \mathrm{g}.LEU) + \mathrm{c}.(\mathrm{u}.SER + \mathrm{c}.SER + \mathrm{a}.SER + \mathrm{g}.SER) +$$

$$\mathrm{a}.(\mathrm{u}.TYR + \mathrm{c}.TYR + \mathrm{a}.STP + \mathrm{g}.STP) + \mathrm{g}.(\mathrm{u}.CYS + \mathrm{c}.CYS + \mathrm{a}.STP + \mathrm{g}.TRP)) +$$

$$\mathrm{c}.(\mathrm{u}.(\mathrm{u}.LEU + \mathrm{c}.LEU + \mathrm{a}.LEU + \mathrm{g}.LEU) + \mathrm{c}.(\mathrm{u}.PRO + \mathrm{c}.PRO + \mathrm{a}.PRO + \mathrm{g}.PRO) +$$

$$\mathrm{a}.(\mathrm{u}.HIS + \mathrm{c}.HIS + \mathrm{a}.GLN + \mathrm{g}.GLN) + \mathrm{g}.(\mathrm{u}.ARG + \mathrm{c}.ARG + \mathrm{a}.ARG + \mathrm{g}.ARG)) +$$

$$\mathrm{a}.(\mathrm{u}.(\mathrm{u}.ILE + \mathrm{c}.ILE + \mathrm{a}.ILE + \mathrm{g}.MET) + \mathrm{c}.(\mathrm{u}.THR + \mathrm{c}.THR + \mathrm{a}.THR + \mathrm{g}.THR) +$$

$$\mathrm{a}.(\mathrm{u}.ASN + \mathrm{c}.ASN + \mathrm{a}.LYS + \mathrm{g}.LYS) + \mathrm{g}.(\mathrm{u}.SER + \mathrm{c}.SER + \mathrm{a}.ARG + \mathrm{g}.ARG)) +$$

$$\mathrm{g}.(\mathrm{u}.(\mathrm{u}.VAL + \mathrm{c}.VAL + \mathrm{a}.VAL + \mathrm{g}.VAL) + \mathrm{c}.(\mathrm{u}.ALA + \mathrm{c}.ALA + \mathrm{a}.ALA + \mathrm{g}.ALA) +$$

$$\mathrm{a}.(\mathrm{u}.ASP + \mathrm{c}.ASP + \mathrm{a}.GLU + \mathrm{g}.GLU) + \mathrm{g}.(\mathrm{u}.GLY + \mathrm{c}.GLY + \mathrm{a}.GLY + \mathrm{g}.GLY)) +$$

$$polyatail.0;$$

$$ALA \stackrel{\text{def}}{=} \overline{\texttt{ala}}.DECODE;\ GLY \stackrel{\text{def}}{=} \overline{\texttt{gly}}.DECODE;$$

$$VAL \stackrel{\text{def}}{=} \overline{\texttt{val}}.DECODE;\ LEU \stackrel{\text{def}}{=} \overline{\texttt{leu}}.DECODE;$$

$$ILE \stackrel{\text{def}}{=} \overline{\texttt{ile}}.DECODE;\ PRO \stackrel{\text{def}}{=} \overline{\texttt{pro}}.DECODE;$$

$$PHE \stackrel{\text{def}}{=} \overline{\texttt{phe}}.DECODE;\ MET \stackrel{\text{def}}{=} \overline{\texttt{met}}.DECODE;$$

$$TRP \stackrel{\text{def}}{=} \overline{\texttt{trp}}.DECODE;\ CYS \stackrel{\text{def}}{=} \overline{\texttt{cys}}.DECODE;$$

$$ARG \stackrel{\text{def}}{=} \overline{\texttt{arg}}.DECODE;\ ASP \stackrel{\text{def}}{=} \overline{\texttt{asp}}.DECODE;$$

$$ASN \stackrel{\text{def}}{=} \overline{\texttt{asn}}.DECODE;\ GLU \stackrel{\text{def}}{=} \overline{\texttt{glu}}.DECODE;$$

$$GLN \stackrel{\text{def}}{=} \overline{\texttt{gln}}.DECODE;\ HIS \stackrel{\text{def}}{=} \overline{\texttt{his}}.DECODE;$$

$$LYS \stackrel{\text{def}}{=} \overline{\texttt{lys}}.DECODE;\ SER \stackrel{\text{def}}{=} \overline{\texttt{ser}}.DECODE;$$

$$THR \stackrel{\text{def}}{=} \overline{\texttt{thr}}.DECODE;\ TYR \stackrel{\text{def}}{=} \overline{\texttt{tyr}}.DECODE;$$

$$STP \stackrel{\text{def}}{=} \overline{stop}.0;$$

The states *STARTCODON1* and *STARTCODON2* allow to identify an `aug` codon and then initiate the actual transcription, that is the *START* process, which produces as output the `met` (Methionine) associated with the start codon and is followed by the *DECODE* process.

The *DECODE* process performs the transcription of each codon to the related amino acid (which is produced as output in the corresponding state) until it reaches a stop codon (which lead to the *STP* process which stops the transcription) or a *polyatail* sequence.

Indeed, the model contemplates the possibility that the ribosome reaches the $3'$ end of the RNA molecule, signalled by the *polyatail* sequence. However, this event is caused by a class of mutations that are beyond the scope of this dissertation.

It is simple to understand that a mutation on a single nucleotide (point mutation) can change the amino acid produced in the translation process. The point mutations can be classified as:

1. **silent mutations**, which generate a codon that still codes for the original amino acid;

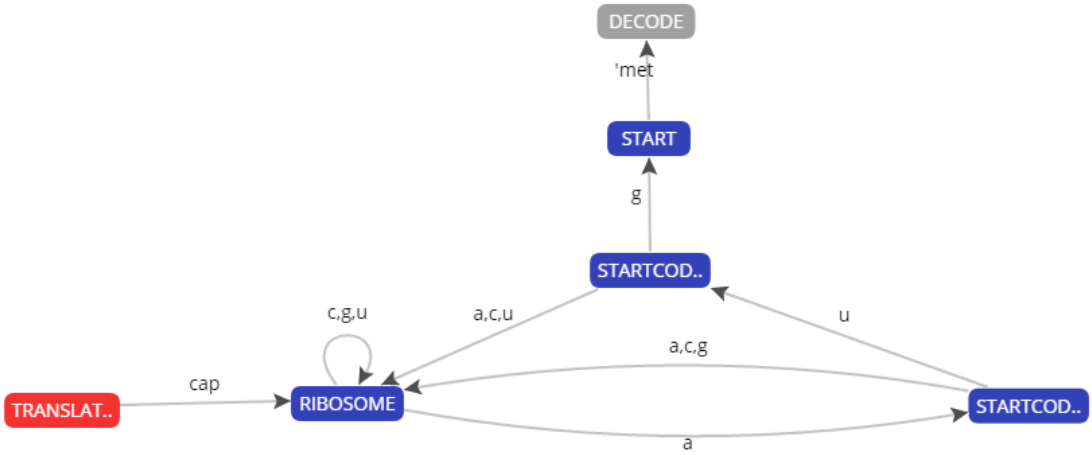


Figure B.4 – Labelled Transition System of the TRANSLATION process generated with the CAAL web-based tool.

2. **nonsense mutations**, which produce a stop codon that cause the truncation of the protein;
3. **missense mutations**, which cause the codon to code for a different amino acid.

In Figure B.4 and Figure B.5 are represented the LTS of the *TRANSLATION* process and of its subprocess *DECODE*.

An examples of missense mutation is the **Hemoglobin S disease** (HbS), also called sickle-cell anaemia, used the study, in the chapter related to these Supplementary Information, the behaviour of proteins and RNAs when their folding process is altered.

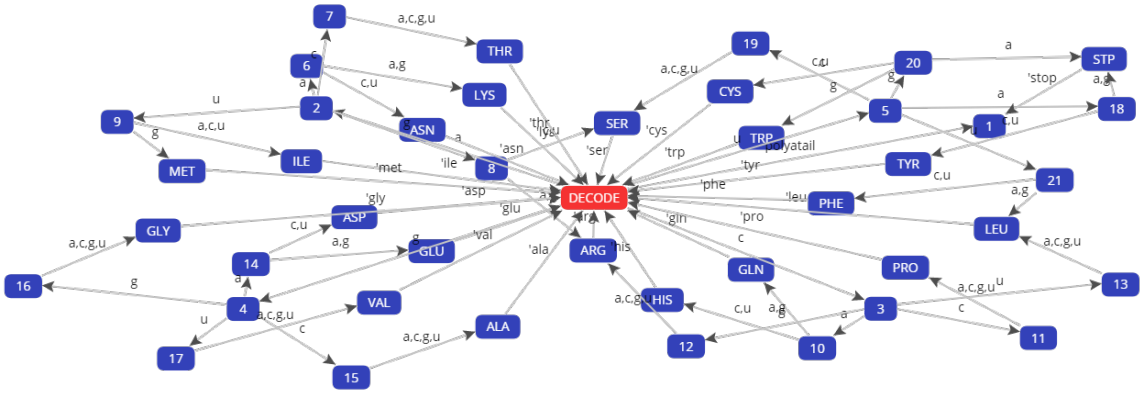


Figure B.5 – Labelled Transition System of the DECODE process (and all its subprocesses) generated with the CAAL web-based tool.

B.2 Formal description of HBB gene expression

The behaviour of the gene expression process is in this section described using HML (Hennessy-Milner Logic) formulas. More precisely it will be shown how the gene that code for one of the β subunits of the haemoglobin molecule (**HBB**) is expressed through each phase detailed in the previous section (replication, transcription, processing and translation).

Each of this step is associated with a HML formula which is satisfied by the related process.

The DNA sequence of the HBB gene (1742 bp long) has been retrieved from the NCBI (National Center for Biotechnology Information) site [63]; the gene contains three exons (coloured in green in their coding regions) and two introns (coloured in blue). In red is highlighted the codon that codes for the Glu 6 of the amino acid sequence of the HBB.

The HML formulas describing the behaviour of each step can be very long, therefore they are represented only as their beginning part (one or two rows), their middle part, where is present the codon of the Glu6, and their ending rows.

```
 cggctgtcatcacttagacctcaccctgtggagccacaccctagggttggccaatctactcccaggagcaggg
agggcaggagccagggctgggcataaaagtcagggcagagccatctattgcttacatttgcttctgacacaact
gtgttcactagcaacctcaaacagacaccatggtgcatctgactcctgaggagaagtctgccgttactgccctg
tggggcaaggtgaacgtggatgaagttggtggtgaggccctgggcaggttggtatcaaggttacaagacaggtt
taaggagaccaatagaaactgggcatgtggagacagagaagactcttgggtttctgataggcactgactctctc
tgcctattggtctattttcccacccttaggctgctggtggtctacccttggacccagaggttctttgagtcctt
tggggatctgtccactcctgatgctgttatgggcaaccctaaggtgaaggctcatggcaagaaagtgctcggtg
cctttagtgatggcctggctcacctggacaacctcaagggcacctttgccacactgagtgagctgcactgtgac
```

aagctgcacgtggatcctgagaacttcagggtgagtctatgggacgcttgatgttttctttccccttctttttct
atggttaagttcatgtcataggaaggggataagtaacagggtacagtttagaatgggaaacagacgaatgattg
catcagtgtggaagtctcaggatcgttttagtttcttttatttgctgttcataacaattgttttcttttgttta
attcttgctttctttttttttcttctccgcaatttttactattatacttaatgccttaacattgtgtataacaa
aaggaaatatctctgagatacattaagtaacttaaaaaaaaactttacacagtctgcctagtacattactattt
ggaatatatgtgtgcttatttgcatattcataatctccctactttattttcttttatttttaattgatacataa
tcattatacatatttatgggttaaagtgtaatgttttaatatgtgtacacatattgaccaaatcagggtaattt
tgcatttgtaattttaaaaaatgctttcttcttttaatatactttttgtttatcttatttctaatactttccc
taatctctttctttcagggcaataatgatacaatgtatcatgcctctttgcaccattctaaagaataacagtga
taatttctgggttaaggcaatagcaatatctctgcatataaatatttctgcatataaattgtaactgatgtaag
aggtttcatattgctaatagcagctacaatccagctaccattctgcttttattttatggttgggataaggctgg
attattctgagtccaagctaggccctttgctaatcatgttcatacctcttatcttcctcccacagctcctggg
caacgtgctggtctgtgtgctggcccatcactttggcaaagaattcaccccaccagtgcaggctgcctatcaga
aagtggtggctggtgtggctaatgccctggcccacaagtatcactaagctcgctttcttgctgtccaatttcta
ttaaaggttcctttgttccctaagtccaactactaaactgggggatattatgaagggccttgagcatctggatt
ctgcctaataaaaaacatttattttcattgcaatgatgtat

B.2.1 Replication

REPLICATION ⊨
⟨origin⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨g⟩⟨′gc⟩
⟨gc⟩⟨′c⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨a⟩⟨′at⟩⟨at⟩⟨′t⟩
⋮
⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩ ⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨a⟩⟨′at⟩⟨at⟩⟨′t⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩ ⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩
⋮
⟨a⟩⟨′at⟩⟨at⟩⟨′t⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨a⟩⟨′at⟩⟨at⟩⟨′t⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨t⟩
⟨′ta⟩⟨ta⟩⟨′a⟩⟨a⟩⟨′at⟩⟨at⟩⟨′t⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨terminus⟩**tt**;

Each formula of the type ⟨b1⟩⟨′b1b2⟩⟨b1b2⟩⟨′b2⟩ (where b1 and b2 are bases) represents the base read by the *DNAPOL* process, followed by the base pair provided as output, which in turn is taken as input by the *PROOFREAD1* process, which finally provides the (possibly) correct base that has to be added to the newly synthesised strand.

B.2.2 Mismatch repair

MMREPAIR ⊨
⟨origin⟩⟨c⟩⟨g⟩⟨′g⟩⟨g⟩⟨c⟩⟨′c⟩⟨g⟩⟨c⟩⟨′c⟩⟨c⟩⟨g⟩⟨′g⟩⟨t⟩⟨a⟩⟨′a⟩⟨g⟩⟨c⟩⟨′c⟩⟨t⟩⟨a⟩⟨′a⟩⟨c⟩⟨g⟩⟨′g⟩⟨a⟩⟨t⟩⟨′t⟩
⟨t⟩⟨a⟩⟨′a⟩⟨c⟩⟨g⟩⟨′g⟩⟨a⟩⟨t⟩⟨′t⟩⟨c⟩⟨g⟩⟨′g⟩⟨t⟩⟨a⟩⟨′a⟩⟨t⟩⟨a⟩⟨′a⟩⟨a⟩⟨t⟩⟨′t⟩⟨g⟩⟨c⟩⟨′c⟩⟨a⟩⟨t⟩⟨′t⟩

⋮

⟨t⟩⟨a⟩⟨′a⟩ ⟨g⟩⟨c⟩⟨′c⟩⟨a⟩⟨t⟩⟨′t⟩⟨g⟩⟨c⟩⟨′c⟩⟨g⟩ ⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩

⋮

⟨t⟩⟨a⟩⟨′a⟩⟨t⟩⟨a⟩⟨′a⟩⟨g⟩⟨c⟩⟨′c⟩⟨c⟩⟨g⟩⟨′g⟩⟨a⟩⟨t⟩⟨′t⟩⟨a⟩⟨t⟩⟨′t⟩⟨t⟩⟨a⟩⟨′a⟩⟨g⟩⟨c⟩⟨′c⟩⟨a⟩⟨t⟩⟨′t⟩⟨t⟩⟨a⟩⟨′a⟩
⟨g⟩⟨c⟩⟨′c⟩⟨t⟩⟨a⟩⟨′a⟩⟨a⟩⟨t⟩⟨′t⟩⟨t⟩⟨a⟩⟨′a⟩⟨terminus⟩**tt**;

Each formula of the type ⟨b1⟩⟨b2⟩⟨′b2⟩ (where b1 and b2 are bases) represents the base read by the *MMRPROTEINS* process on the template strand, the base read on the newly synthesised strand, followed by the (possibly) correct base that should be paired with the first one.

The following is the sequence obtained after the *REPLICATION* process has operated on the template DNA sequence:

```
 gccgacagtagtgaatctggagtgggacacctcggtgtgggatcccaaccggttagatgagggtcctcgtccc
tcccgtcctcggtcccgacccgtattttcagtcccgtctcggtagataacgaatgtaaacgaagactgtgttga
cacaagtgatcgttggagtttgtctgtggtaccacgtagactgaggactcctcttcagacggcaatgacgggac
accccgttccacttgcacctacttcaaccaccactccgggacccgtccaaccatagttccaatgttctgtccaa
attcctctggttatctttgacccgtacacctctgtctcttctgagaacccaaagactatccgtgactgagagag
acggataaccagataaaagggtgggaatccgacgaccaccagatgggaacctgggtctccaagaaactcaggaa
acccctagacaggtgaggactacgacaatacccgttgggattccacttccgagtaccgttctttcacgagccac
ggaaatcactaccggaccgagtggacctgttggagttcccgtggaaacggtgtgactcactcgacgtgacactg
ttcgacgtgcacctaggactcttgaagtcccactcagataccctgcgaactacaaaagaaaggggaagaaaaga
taccaattcaagtacagtatccttcccctattcattgtcccatgtcaaatcttaccctttgtctgcttactaac
gtagtcacaccttcagagtcctagcaaaatcaaagaaaataaacgacaagtattgttaacaaaagaaaacaaat
taagaacgaaagaaaaaaaaagaagaggcgttaaaaatgataatatgaattacggaattgtaacacatattgtt
ttcctttatagagactctatgtaattcattgaattttttttgaaatgtgtcagacggatcatgtaatgataaa
ccttatatacacacgaataaacgtataagtattagagggatgaaataaaagaaaataaaaattaactatgtatt
agtaatatgtataaatacccaatttcacattacaaaattatacacatgtgtataactggtttagtcccattaaa
acgtaaacattaaaatttttacgaaagaagaaaattatatgaaaaaacaaatagaataaagattatgaaaggg
attagagaaagaaagtcccgttattactatgttacatagtacggagaaacgtggtaagatttcttattgtcact
attaaagacccaattccgttatcgttatagagacgtatatttataaagacgtatatttaacattgactacattc
tccaaagtataacgattatcgtcgatgttaggtcgatggtaagacgaaaataaaataccaaccctattccgacc
taataagactcaggttcgatccgggaaaacgattagtacaagtatggagaatagaaggagggtgtcgaggaccc
gttgcacgaccagacacacgaccgggtagtgaaaccgtttcttaagtggggtggtcacgtccgacggatagtct
ttcaccaccgaccacaccgattacgggaccgggtgttcatagtgattcgagcgaaagaacgacaggttaaagat
aatttccaaggaaacaagggattcaggttgatgatttgaccccctataatacttcccggaactcgtagacctaa
gacggattatttttgtaaataaaagtaacgttactacata
```

B.2.3 Transcription

$TRANSCRIPTION \models$
⟨promoter⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨a⟩⟨′au⟩⟨au⟩⟨′u⟩⟨c⟩⟨′cg⟩
⟨cg⟩⟨′g⟩⟨a⟩⟨′au⟩⟨au⟩⟨′u⟩⟨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⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩ ⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩
⋮
⟨g⟩⟨′gc⟩⟨gc⟩⟨′c⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨a⟩⟨′au⟩⟨au⟩⟨′u⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨a⟩
⟨′au⟩⟨au⟩⟨′u⟩⟨c⟩⟨′cg⟩⟨cg⟩⟨′g⟩⟨a⟩⟨′au⟩⟨au⟩⟨′u⟩⟨t⟩⟨′ta⟩⟨ta⟩⟨′a⟩⟨a⟩⟨′au⟩⟨au⟩⟨′u⟩⟨terminator⟩**tt**;

Each formula of the type ⟨b1⟩⟨′b1b2⟩⟨b1b2⟩⟨′b2⟩ represents the base read by the *RNAPOL* process, followed by the base pair provided as output, which in turn is taken as input by the *PROOFREAD2* process, which finally provides the (possibly) correct base which has to be added to the RNA sequence.

The following is the sequence obtained after the *TRANSCRIPTION* process has taken place:

cggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccaggagcaggg
agggcaggagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugcuucugacacaacu
guguucacuagcaaccucaaacagacaccauggugcaucugacuccugaggagaagucugccguuacugcccug
uggggcaaggugaacguggaugaaguugguggugaggcccugggcagguugguaucaagguuacaagacagguu
uaaggagaccaauagaaacugggcauguggagacagagaagacucuuggguuucugauaggcacugacucucuc
ugccuauuggucuauuuucccacccuuaggcugcugguggucuacccuuggacccagagguucuuugaguccuu
uggggaucuguccacuccugaugcuguuaugggcaacccuaaggugaaggcucauggcaagaaagugcucggug
ccuuuagugauggccuggcucaccuggacaaccucaagggcaccuuugccacacugagugagcugcacugugac
aagcugcacguggauccugagaacuucagggugagucuaugggacgcuugauguuuucuuuccccuucuuuucu
augguuaaguucaugucauaggaaggggauaaguaacaggguacaguuuagaaugggaaacagacgaaugauug
caucaguguggaagucucaggaucguuuuaguuucuuuuauuugcuguucauaacaauuguuuucuuuuguuua
auucuugcuuucuuuuuuuuucuucuccgcaauuuuuacuauuauacuuaaugccuuaacauuguguauaacaa
aaggaaauaucucugagauacauuaaguaacuuaaaaaaaaacuuuacacagucugccuaguacauuacuauuu
ggaauauaugugugcuuauuugcauauucauaaucucccuacuuuauuuucuuuuauuuuuaauugauacauaa
ucauuauacauauuuaugggguuaaaguguaauguuuuaauauguguacacauauugaccaaaucaggguaauuu
ugcauuuguaauuuuaaaaaaugcuuucuucuuuuaauauacuuuuuuguuuaucuuauuucuaauacuuuccc
uaaucucuuucuuucagggcaauaaugauacaauguaucaugccucuuugcaccauucuaaagaauaacaguga
uaauuucuggguuaaggcaauagcaauaucucugcauauaaauauuucugcauauaaauuguaacugauguaag
agguuucauauugcuaauagcagcuacaauccagcuaccauucugcuuuuauuuuaugguugggauaaggcugg
auuauucugaguccaagcuaggcccuuuugcuaaucauguucauaccucuuaucuuccucccacagcuccuggg
caacgugcuggucugugugcuggcccaucacuuuggcaaagaauucaccccaccagugcaggcugccuaucaga
aaguggugggcuggugugggcuaaugcccuggcccacaaguaucacuaagcucgcuuucuugcuguccaauuucua

uuaaagguuccuuuguucccuaaguccaacuacuaaacuggggggauauuaugaagggccuugagcaucuggauu
cugccuaauaaaaaacauuuauuuucauugcaaugauguau

B.2.4 Processing

PROCESSING $\models$
$\langle$fivepend$\rangle\langle'$cap$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle$
$\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle$
$\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle$
$\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle$
$\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle$
$\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle$
$\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle$
$\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle$
$\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle$
$\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle$
$\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle$
$\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle$
$\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle$ $\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle$
$\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle$ $\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle$ $\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle$
$\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle$
$\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle$
$\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle$
$\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$g$\rangle$
$\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle$
$\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle$
$\langle$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$\rangle$
$\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle$
$\vdots$
$\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$c$\rangle\langle'$c$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle$
$\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$g$\rangle\langle'$g$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$a$\rangle\langle'$a$\rangle\langle$u$\rangle\langle'$u$\rangle\langle$tpend$\rangle\langle'$polyatail$\rangle$**tt**;

Each formula of the type $\langle$b$\rangle\langle'$b$\rangle$ (where b is a base) represents the base read by the SPLICING process, followed by the same base produced as output; this is repeated until a *gu* sequence signalling the beginning of an intron is found. Starting from this point the formula becomes of the type $\langle$b$\rangle$, which indicates that each base read in this phase (performed by the *CUT* process) is

not produced as output. The CUT process continues until it reaches the *ag* sequence that signals the end of the intron; after that, the previous formula (of the type $\langle \mathtt{b} \rangle \langle' \mathtt{b} \rangle$) describes again the behaviour of the process.

The following is the sequence obtained after the *PROCESSING* has taken place:

cggcugucaucacuuagaccucacccuguggagccacacccuaggguuggccaaucuacucccaggagcagggagggcag
gagccagggcugggcauaaaagucagggcagagccaucuauugcuuacauuugcuucugacacaacuguguucacuagc
aaccucaaacagacaccauggugcaucugacuccugaggagaagucugccguuacugcccugugggcaaggugaacgug
gaugaaguuggugugaggcccugggcaggcugcugguggucuacccuuggacccagagguucuuugaguccuuugggg
aucuguccacuccugaugcuguuaugggcaacccuaaggugaaggcucauggcaagaaagugcucggugccuuuaguga
uggccuggcucaccuggacaaccucaagggcaccuuugccacacugagugagcugcacugugacaagcugcacguggauc
cugagaacuucaggcuccugggcaacgugcuggucugugugcuggcccaucacuuuggcaaagaauucaccccaccagu
gcaggcugccuaucagaaagugguggcuggugugguggcuaaugcccuggcccacaaguaucacuaagcucgcuuucuugcu
guccaauuucuauuaaagguuccuuuguucccuaaguccaacuacuaaacuggggguauauuaugaagggccuugagca
ucuggauucugccuaauaaaaaacauuuauuuucauugcaaugauguau

B.2.5 Translation

$TRANSLATION \models$
$\langle \mathtt{cap} \rangle \langle \mathtt{c} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle$
$\langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle$
$\langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle$
$\langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle$
$\langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle$
$\langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle$
$\langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{met} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{val} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{his} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{leu} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{thr} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{pro} \rangle \langle \mathtt{g} \rangle$
$\langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{glu} \rangle \langle \mathtt{g} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{glu} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{lys} \rangle \langle \mathtt{u} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{ser} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{ala} \rangle \langle \mathtt{g} \rangle \langle \mathtt{u} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{val} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle \mathtt{u} \rangle$
$\langle' \mathtt{thr} \rangle$
$\vdots$
$\vdots$
$\langle \mathtt{c} \rangle \langle \mathtt{u} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{leu} \rangle \langle \mathtt{g} \rangle \langle \mathtt{c} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{ala} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{his} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle \mathtt{g} \rangle \langle' \mathtt{lys} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{u} \rangle \langle' \mathtt{tyr} \rangle \langle \mathtt{c} \rangle \langle \mathtt{a} \rangle \langle \mathtt{c} \rangle \langle' \mathtt{his} \rangle \langle \mathtt{u} \rangle \langle \mathtt{a} \rangle$
$\langle \mathtt{a} \rangle \langle' \mathtt{stop} \rangle \mathbf{tt};$

The formula of the type $\langle \mathtt{b} \rangle$ (where b is a base) represents each base read before the RIBOSOME process reaches the *aug* codon; after that, the formula becomes of the type $\langle \mathtt{b1} \rangle \langle \mathtt{b2} \rangle \langle \mathtt{b3} \rangle \langle' \mathtt{aa} \rangle$ (where b1, b2 and b3 are bases, while aa is an amino acid), and describes how the process translates each codon it encounters until it reaches a stop codon (represented in this case by the formula $\langle \mathtt{u} \rangle \langle \mathtt{a} \rangle \langle \mathtt{a} \rangle \langle' \mathtt{stop} \rangle$).

The amino acid sequence of the HBB subunit of the Haemoglobin molecule finally is:

```
 MetValHisLeuThrProGluGluLysSerAlaValThrAlaLeuTrpGlyLysValAsnValAspGluValG
lyGlyGluAlaLeuGlyArgLeuLeuValValTyrProTrpThrGlnArgPhePheGluSerPheGlyAspLe
uSerThrProAspAlaValMetGlyAsnProLysValLysAlaHisGlyLysLysValLeuGlyAlaPheSerA
spGlyLeuAlaHisLeuAspAsnLeuLysGlyThrPheAlaThrLeuSerGluLeuHisCysAspLysLeuHi
sValAspProGluAsnPheArgLeuLeuGlyAsnValLeuValCysValLeuAlaHisHisPheGlyLysGlu
PheThrProProValGlnAlaAlaTyrGlnLysValValAlaGlyValAlaAsnAlaLeuAlaHisLysTyrHi
s
```

The validity of all the formulas in this section has been verified with the aid of the web-based tool CAAL [3]; due to the length of each formulas is not possible to provide the screenshot of the results obtained. A proof of the validity of each atomic (meaning indivisible) formulas which characterise each step will be provided in Chapter 4 on page 51.

Supplementary Information to Chapter 7

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

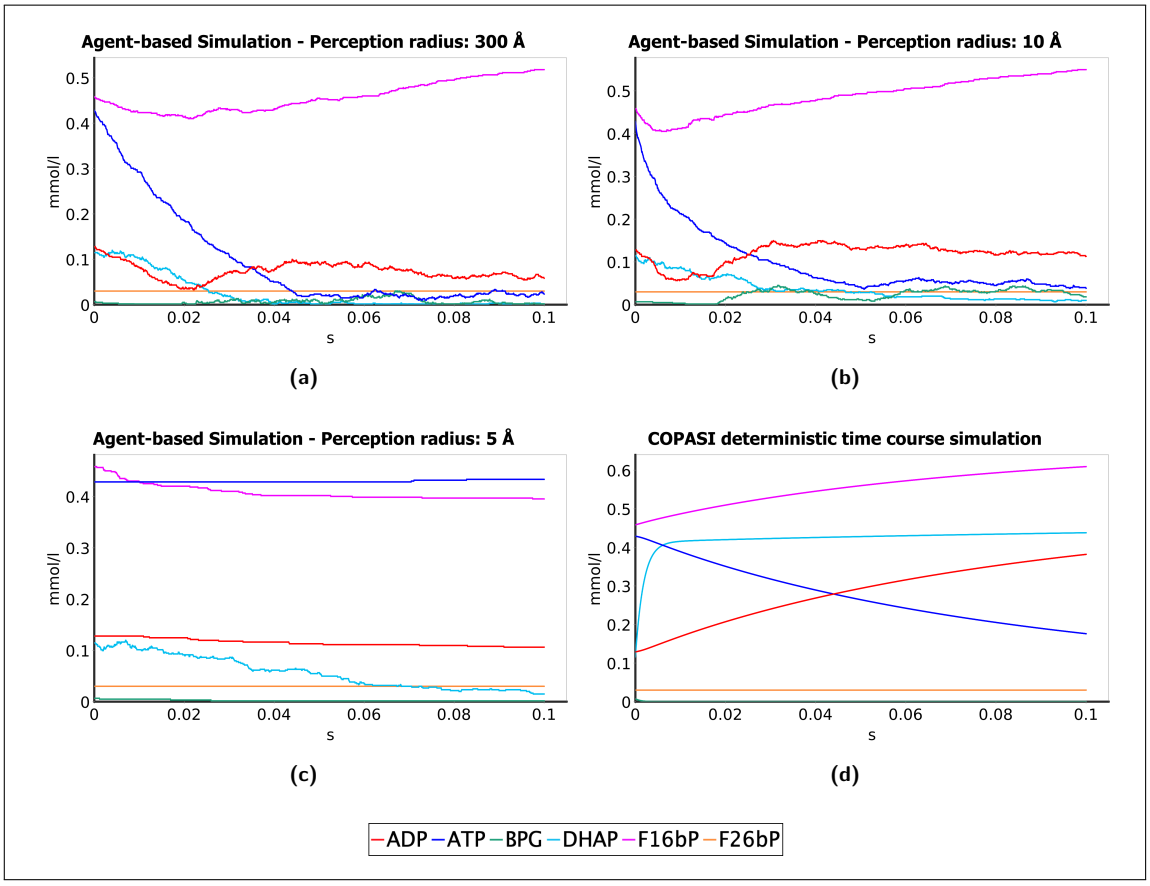


Figure C.1 – Panel 1

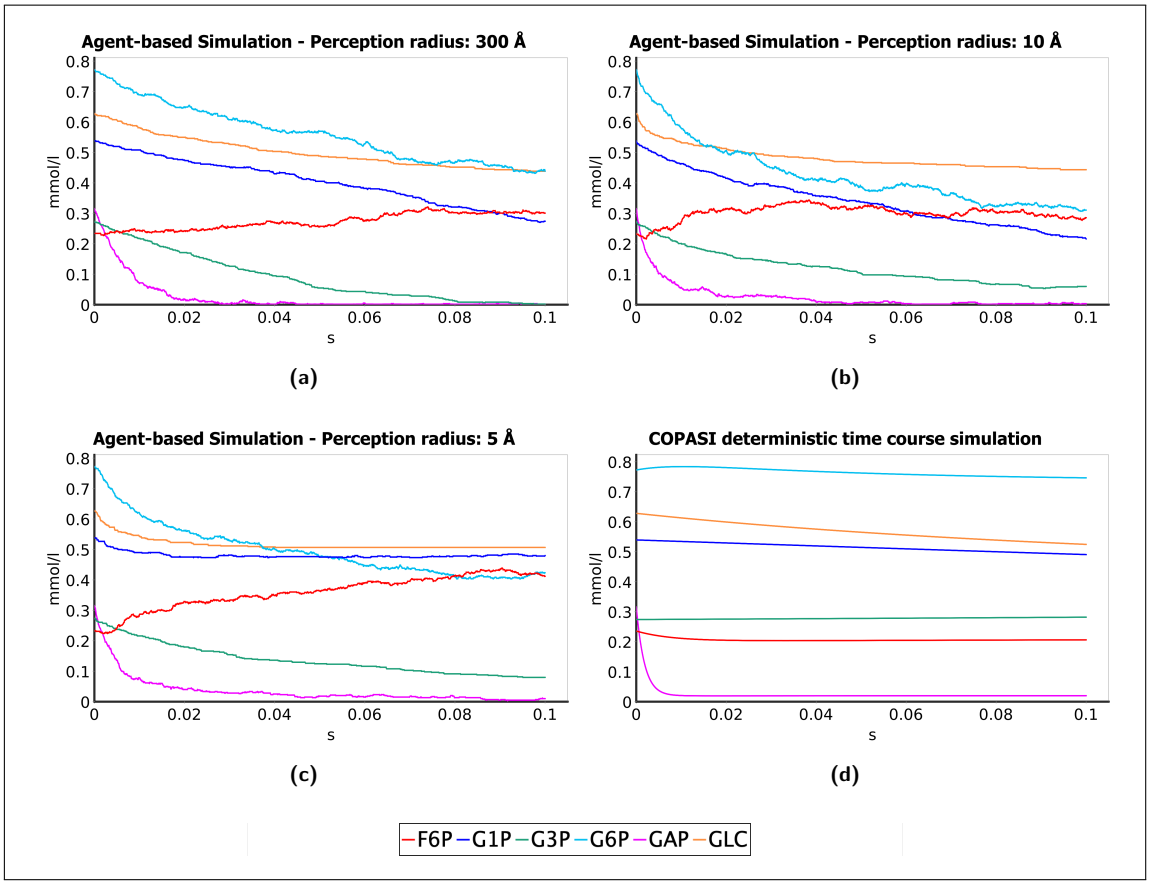


Figure C.2 – Panel 2

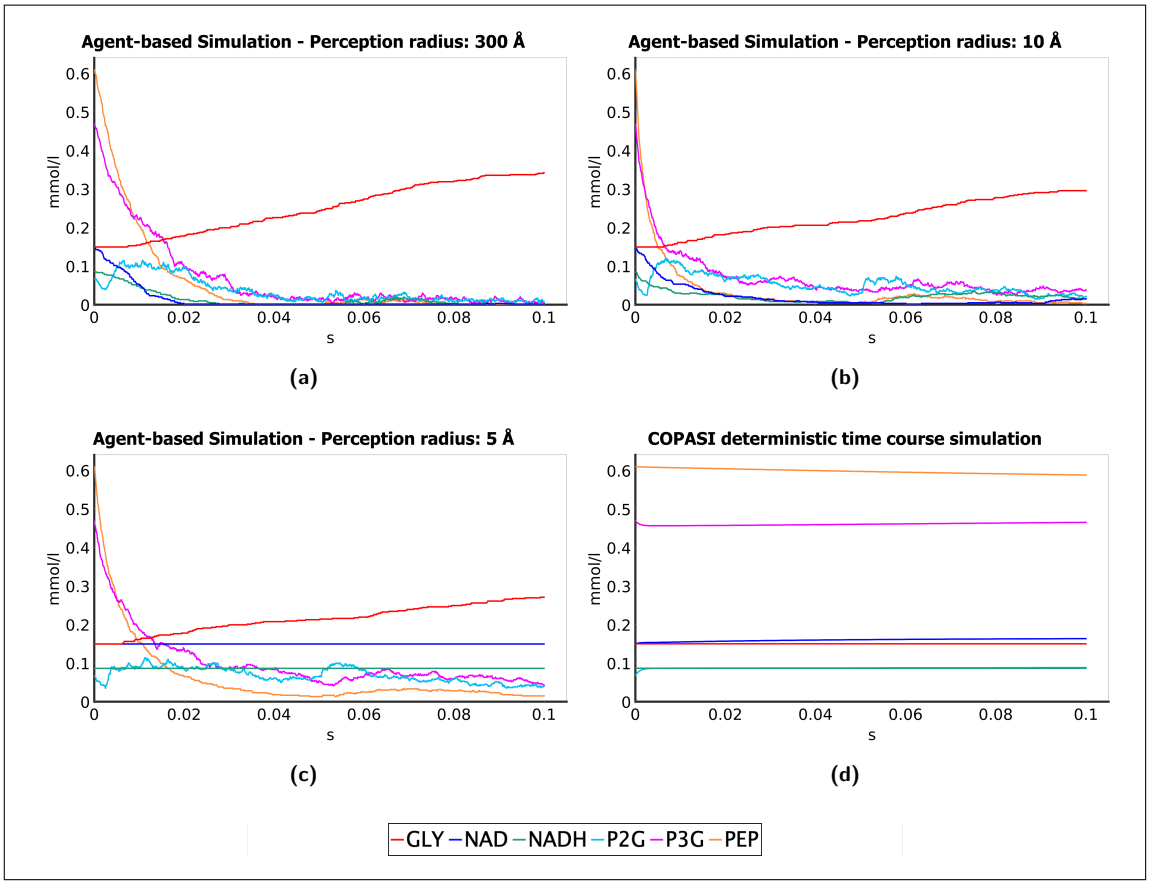


Figure C.3 – Panel 3

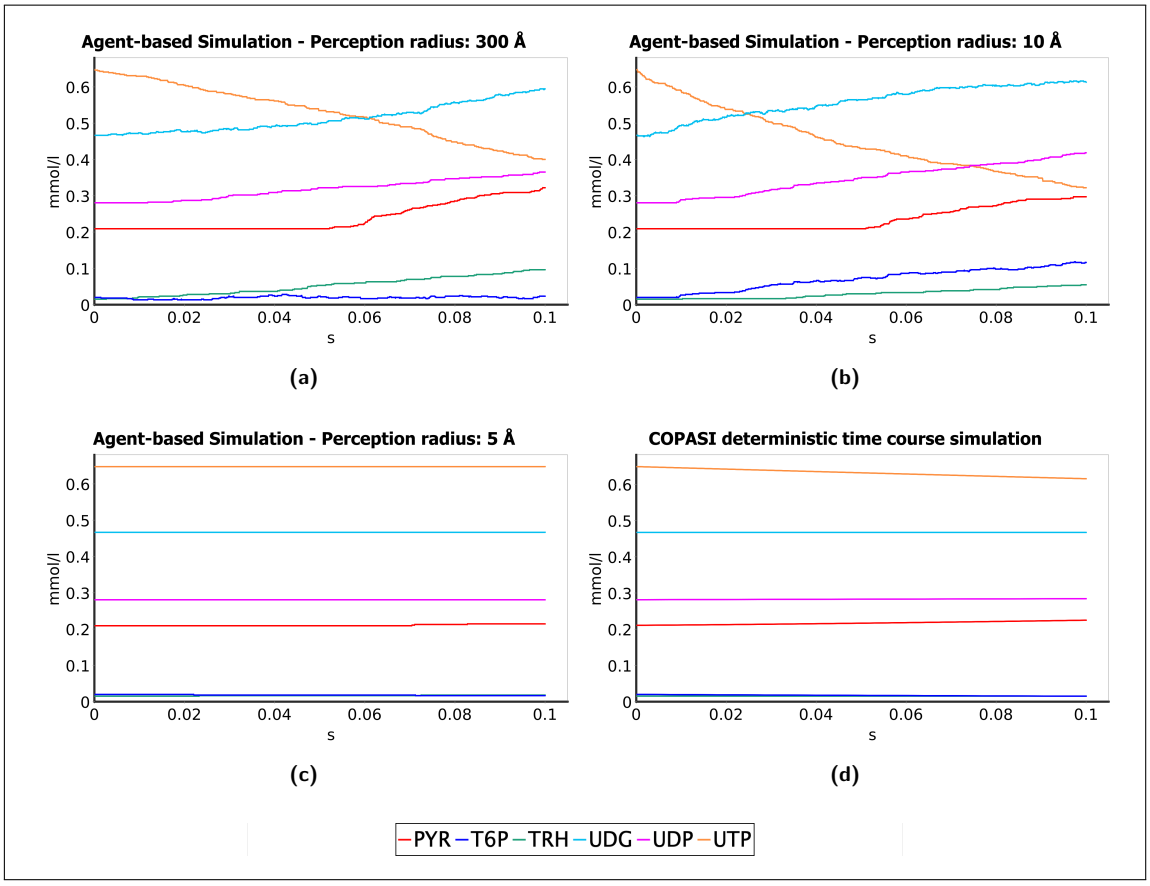


Figure C.4 – Panel 4

Bibliography

[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: 978-0-8153-4432-2 978-0-8153-4464-3 978-0-8153-4524-4.

[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] Aristote. *Aristote : Oeuvres complètes et annexes (annotées, illustrées)*. Ed. by Jules Barthélemy-Saint-Hilaire. 1st edition. Arvensa Editions, 2019.

[5] Aristotle. *Aristotle's Metaphysics: a revised text with introduction and commentary*. Ed. by W. D. Ross. Oxford : Clarendon Press, 1924.

[6] Arianna Baldoncini. *Orion: A Spatial Multi Agent System Framework for Computational Cellular Dynamics of Metabolic Pathways*. MSc Thesis UNICA-CS-2004-1. Universita' di Camerino, 2004.

[7] 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.

[8] J.M. Berg, J.L. Tymoczko, and L. 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/.

[9] Andrea Bernini et al. "Process calculi for biological processes". In: *Natural Computing* 17 (2018). DOI: 10.1007/s11047-018-9673-2.

[10] 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.

[11] 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.

[12] 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.

[13] Nicola Cannata, Flavio Corradini, and Emanuela Merelli. “Multiagent modelling and simulation of carbohydrate oxidation in cell”. In: *Int. J. Modelling, Identification and Control* 3 (2008). DOI: 10.1504/IJMIC.2008.018191.

[14] 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.

[15] Vincent Danos and Cosimo Laneve. “Formal molecular biology”. In: *Theoretical Computer Science* 325.1 (2004). Computational Systems Biology, pp. 69 –110. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2004.03.065.

[16] 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.

[17] 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.

[18] Claus Desler et al. “Is There a Link between Mitochondrial Reserve Respiratory Capacity and Aging?” eng. In: *J Aging Res* 2012 (2012), p. 192503. DOI: 10.1155/2012/192503.

[19] Jennifer Doudna and Jon Lorsch. “Ribozyme catalysis: Not different, just worse”. In: *Nature structural & molecular biology* 12 (2005), pp. 395–402. DOI: 10.1038/nsmb932.

[20] D Allan Drummond and Claus O Wilke. “Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution”. In: *Cell* 134.2 (2008), pp. 341–352.

[21] Harrison Echols and Carol Gross. *Operators and Promoters: The Story of Molecular Biology and Its Creators*. Berkeley: University of California Press, 2001. ISBN: 978-0-520-21331-9.

[22] Herbert Edelsbrunner and John Harer. “Persistent Homology-a Survey”. In: *Contemp. Math.* 453 (2008), pp. 257–282.

[23] Andrzej Ehrenfeucht et al. “Formal systems for gene assembly in ciliates”. In: *Theoretical Computer Science* 292.1 (2003), pp. 199–219.

[24] Adrian Ferré-D’Amaré. “The glmSribozyme: Use of a small molecule coenzyme by a gene-regulatory RNA”. In: *Quarterly reviews of biophysics* 43 (2010), pp. 423–47. DOI: 10.1017/S0033583510000144.

[25] B M Fischer, M Walther, and P 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.

[26] Claudio Forcato. *Orion: A Spatial Multiagent System Framework for Cellular Simulation. Implementation of Molecular Movement at the Mesoscale*. MSc Thesis UNICAM-CS-2004-2. Universita’ di Camerino, 2005.

[27] Gianluigi Forloni et al. "Protein misfolding in Alzheimer's and Parkinson's disease: genetics and molecular mechanisms". In: *Neurobiology of Aging* 23.5 (2002). Brain Aging: Identifying the Brakes and Accelerators, pp. 957 –976. ISSN: 0197-4580. DOI: https://doi.org/10.1016/S0197-4580(02)00076-3. URL: http://www.sciencedirect.com/science/article/pii/S0197458002000763.

[28] 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.

[29] 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.

[30] Walter Gilbert. "Origin of life: The RNA world". In: *Nature* 319.6055 (1986), p. 618.

[31] 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.

[32] Niels Gregersen et al. "Protein misfolding and human disease". In: *Annu. Rev. Genomics Hum. Genet.* 7 (2006), pp. 103–124.

[33] 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.

[34] 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.

[35] Antarip Halder et al. "How Does Mg2+ Modulate the RNA Folding Mechanism: A Case Study of the G: CW: W Trans Basepair". In: *Biophysical Journal* 113.2 (2017), pp. 277–289.

[36] 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.

[37] F Ulrich Hartl, Andreas Bracher, and Manajit Hayer-Hartl. "Molecular chaperones in protein folding and proteostasis". In: *Nature* 475.7356 (2011), p. 324.

[38] 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.

[39] 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.

[40] Matthew Hennessy and Robin Milner. "Algebraic laws for nondeterminism and concurrency". In: *Journal of the ACM (JACM)* 32.1 (1985), pp. 137–161.

[41] S. Hoops et al. "COPASI–a COmplex PAthway SImulator". en. In: *Bioinformatics* 22.24 (Dec. 2006), pp. 3067–3074. DOI: 10.1093/bioinformatics/btl485.

[42] M. 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.

[43] Nicholas R Jennings. "An Agent-Based Approach for Building Complex Software Systems". In: *Commun. ACM* 44.4 (2001), pp. 35–41.

[44] Randi Jimenez, Julio Polanco, and Andrej Lupták. "Chemistry and Biology of Self-Cleaving Ribozymes". In: *Trends in biochemical sciences* 40 (2015). DOI: 10.1016/j.tibs.2015.09.001.

[45] 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.

[46] Robert M Keller. "Formal verification of parallel programs". In: *Communications of the ACM* 19.7 (1976), pp. 371–384.

[47] Kim Larsen. "Proof systems for satisfiability in Hennessy-Milner Logic with recursion". In: *Theoretical Computer Science* 72 (1990), pp. 265–288. DOI: 10.1016/0304-3975(90)90038-J.

[48] 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.

[49] Lucasharr. *Ribozyme structure picutres*. CC BY-SA 4.0. Accessed 06 June 2020. 2014. URL: https://commons.wikimedia.org/wiki/File:Ribozyme_structure_picutres.png.

[50] Stefano Maestri and Emanuela Merelli. "Process calculi may reveal the equivalence lying at the heart of RNA and proteins". In: *Scientific Reports* 9.1 (2019), p. 559. ISSN: 2045-2322. DOI: 10.1038/s41598-018-36965-1.

[51] Adane Mamuye et al. "Persistent Homology Analysis of RNA". In: *Molecular Based Mathematical Biology* 4 (2016). DOI: 10.1515/mlbmb-2016-0002.

[52] A.L. Mamuye, E. Merelli, and L. Tesei. "A graph grammar for modelling RNA folding". In: *Electronic Proceedings in Theoretical Computer Science, EPTCS* 231 (2016), pp. 31–41.

[53] Michele Mattioni. *Orion: A Multiagent System Framework for Cellular Simulation. Implementation of Metabolic Reaction at the Mesoscale*. MSc Thesis UNICAM-CS-2005-2. Universita' di Camerino, 2005.

[54] Emanuela Merelli, Marco Pettini, and Mario Rasetti. "Topology Driven Modeling: The IS Metaphor". In: *Nat. Comput.* 14.3 (2015), pp. 421–430.

[55] 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.

[56] Emanuela Merelli and Michal Young. "Validating MAS simulation models with mutation". In: *Multiagent and Grid Systems* 3 (2007), pp. 225–243. DOI: 10.3233/MGS-2007-3206.

[57] 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.

[58] Robin Milner. *Communication and concurrency*. Prentice Hall International, UK, 1989.

[59] Shona A. Mookerjee, David G. Nicholls, and Martin D. Brand. "Determining Maximum Glycolytic Capacity Using Extracellular Flux Measurements". en. In: *PLoS ONE* 11.3 (Mar. 2016). Ed. by Jianhua Zhang, e0152016. DOI: 10.1371/journal.pone.0152016.

[60] "More Is Different". en. In: 177 (1972), p. 4.

[61] Uma Nagaswamy et al. "Database of non-canonical base pairs found in known RNA structures". In: *Nucleic Acids Research* 28.1 (2000), pp. 375–376.

[62] 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.

[63] NCBI. *Homo sapiens gene HBB, encoding hemoglobin, beta.* URL: ncbi.nlm.nih.gov/ieb/research/acembly/av.cgi?db=human&c=Gene&l=HBB.

[64] David G. Nicholls. "Spare Respiratory Capacity, Oxidative Stress and Excitotoxicity". eng. In: *Biochem Soc Trans* 37.Pt 6 (Dec. 2009), pp. 1385–1388. DOI: 10.1042/BST0371385.

[65] 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.

[66] Asha Pandey et al. "Advancements in Nucleic Acid Based Therapeutics against Respiratory Viral Infections". In: *Journal of Clinical Medicine* 8 (2018), p. 6. DOI: 10.3390/jcm8010006.

[67] Giovanni Petri et al. "Topological Strata of Weighted Complex Networks". In: *PloS One* 8.6 (2013).

[68] A. Phillips, L. Cardelli, and G. Castagna. "A graphical representation for biological processes in the stochastic pi-calculus". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4230 LNBI (2006), pp. 123–152.

[69] Marco Piangerelli, Stefano Maestri, and Emanuela Merelli. "Visualising 2-simplex formation in metabolic reactions". In: *Journal of Molecular Graphics and Modelling* 97 (2020), p. 107576. ISSN: 1093-3263. DOI: 10.1016/j.jmgm.2020.107576.

[70] Marco Piangerelli, Luca Tesei, and Emanuela Merelli. "A Persistent Entropy Automaton for the Dow Jones Stock Market". In: *Fundamentals of Software Engineering. FSEN 2019. Lecture Notes in Computer Science*. Ed. by Hossein Hojjat and Mieke Massink. Vol. 11671. Cham: Springer International Publishing, 2019, pp. 37–42. ISBN: 978-3-030-31517-7. DOI: 10.1007/11419822âĆĹ.

[71] 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.

[72] Matthew W Powner, Béatrice Gerland, and John D Sutherland. "Synthesis of activated pyrimidine ribonucleotides in prebiotically plausible conditions". In: *Nature* 459.7244 (2009), p. 239.

[73] Michela Quadrini, Luca Tesei, and Emanuela Merelli. "An algebraic language for RNA pseudoknots comparison". In: *BMC Bioinformatics* 20 (2019). DOI: 10.1186/s12859-019-2689-5.

[74] 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 & Bioscience* 10 (2020). DOI: 10.1186/s13578-020-0373-0.

[75] 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*. Cambridge University Press, 2016, pp. 1–42.

[76] Arun Renganathan and Emanuela Felley-Bosco. "Long Noncoding RNAs in Cancer and Therapeutic Potential". In: *Advances in Experimental Medicine and Biology* 1008 (2017), pp. 199–222. DOI: 10.1007/978-981-10-5203-3_7.

[77] 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.

[78] Alexander Serganov and Dinshaw Patel. "Ribozymes, riboswitches and beyond: regulation of gene expression without proteins." In: *Nature reviews. Genetics* 8 (2007), pp. 776–790. DOI: 10.1038/nrg2172.

[79] Kieran Smallbone et al. "A Model of Yeast Glycolysis Based on a Consistent Kinetic Characterisation of All Its Enzymes". In: *FEBS Letters*. A Century of Michaelis - Menten Kinetics 587.17 (Sept. 2013), pp. 2832–2841. DOI: 10.1016/j.febslet.2013.06.043.

[80] Kieran Smallbone et al. "A Model of Yeast Glycolysis Based on a Consistent Kinetic Characterisation of All Its Enzymes". In: *FEBS Letters*. A Century of Michaelis - Menten Kinetics 587.17 (Sept. 2013), pp. 2832–2841. DOI: 10.1016/j.febslet.2013.06.043.

[81] Giancarlo Solaini, Gianluca Sgarbi, and Alessandra Baracca. "Oxidative Phosphorylation in Cancer Cells". eng. In: *Biochim Biophys Acta* 1807.6 (June 2011), pp. 534–542. DOI: 10.1016/j.bbabio.2010.09.003.

[82] Xiao-Bo Tang, Gerd Hobom, and Dong Luo. "Ribozyme mediated destruction of influenza A virus". In: *Journal of medical virology* 42 (1994), pp. 385–95. DOI: 10.1002/jmv.1890420411.

[83] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. "JavaPlex: A Research Software Package for Persistent (Co)Homology". In: *Proceedings of ICMS 2014*. Ed. by Han Hong and Chee Yap. Lecture Notes in Computer Science 8592. 2014, pp. 129–136.

[84] 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.

[85] 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.

[86] 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.

[87] James D. Watson, ed. *Molecular Biology of the Gene*. Seventh edition. Boston: Pearson, 2014. ISBN: 978-0-321-76243-6 978-0-321-90537-6 978-0-321-90264-1.

[88] Hans V. Westerhoff et al. "From Silicon Cellsilicon Cell to Silicon Humansilicon 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.

[89] 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.

[90] A.A. Zamyatnin. "Protein Volume in Solution". en. In: *Progress in Biophysics and Molecular Biology* 24 (Jan. 1972), pp. 107–123. DOI: 10.1016/0079-6107(72)90005-3.

[91] 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.

[92] Afra Zomorodian. "Topological Data Analysis". In: *Advances in Applied and Computational Topology. Proceedings of Symposia in Applied Mathematics.* Vol. 70. 2007, pp. 1–39.