Structuration d'un flot de conception pour la biologie synthétique

Yves Gendrault

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MATHEMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGENIEUR

Laboratoire ICube

THÈSE présentée par :

Yves GENDRAULT

soutenue le : **6 décembre 2013**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : Sciences pour l'ingénieur - Microélectronique

Structuration d'un flot de conception pour la biologie synthétique

MEMBRES DU JURY :

Directeur de thèse :	**M. LALLEMENT Christophe,** Professeur, ICube, Strasbourg.
Rapporteur :	**Mme LEWIS Noëlle,** Professeur, IMS, Bordeaux.
Rapporteur :	**M. FAULON Jean-Loup**, Professeur, iSSB, Evry.
Examinateur :	**M. VACHOUX Alain,** Docteur, EPFL, Lausanne
Examinateur :	**M. HAIECH Jacques,** Professeur, LIT, Strasbourg
Examinateur :	**M. MADEC Morgan,** Docteur, ICube, Strasbourg
Invité :	**M. PECHEUX François,** Professeur, LIP6, Paris

Numéro d'ordre : 2241

Remerciements

Mes remerciements vont tout d'abord à mon directeur de thèse, Christophe Lallement, pour m'avoir permis de réaliser cette thèse et m'avoir accepté dans son équipe de recherche. Ses conseils et son soutien lors de ces trois années ont été particulièrement importants pour moi.

Je remercie également Morgan Madec, mon encadrant de thèse, pour tout le temps qu'il m'a consacré ainsi que pour ses précieux conseils. Je tiens particulièrement à le remercier pour l'aide qu'il m'a apportée lors de la rédaction de ce manuscrit et pour toutes ses idées qui ont permis d'enrichir non seulement ce manuscrit mais aussi tout le travail effectué durant ma thèse.

Je remercie Jacques Haiech pour sa contribution tout au long de ma thèse ; ses connaissances en biologie ayant été particulièrement utiles lors de mes recherches. Je remercie François Pêcheux pour sa collaboration et son aide précieuse sur le SystemC-AMS.

Un grand merci également à Daniel Mathiot pour son accueil au sein du laboratoire InESS ainsi qu'à Michel de Mathelin au sein du laboratoire ICube.

Merci à Noëlle Lewis et à Jean-Loup Faulon pour leur intérêt vis-à-vis de mon travail et pour avoir accepté d'être rapporteurs de ma thèse, ainsi qu'à Alain Vachoux pour avoir accepté d'être présent dans mon jury. Je remercie également Alfonso Jaramillo et Wilfried Uhring pour avoir accepté de faire partie du jury de mon comité de suivi à mi-parcours.

Je tiens également à remercier les étudiants ayant participés à ce travail : Vincent Wlotzko et Martin Andraud pour leur participation à la création d'un générateur de modèles dans le cadre de l'iGEM 2012, Loic Bauer pour son stage de fin d'études ayant porté sur l'adaptation des outils numériques, Alexandre Phe et Julien Catineau pour leurs stages de première année ayant porté respectivement sur le générateur de polynômes de liaison et sur la génération automatique de modèles.

Merci également à mes collègues : Dumitru Armeanu, Nicolas Chevillon, Anne-Sophie Cordan, Norbert Dumas, Fabien Ehrardt, Sadiara Fall, Jérôme Heitz, Jean-Baptiste Kammerer, Cyril Kern, Yann Leroy, Fabien Prégaldiny, Adam Raba, François Schwartz et Qing Sun.

Je tenais finalement à remercier ma famille et mes amis pour leur soutien tout au long de cette thèse et particulièrement Stéphanie Munier pour sa grande aide dans la relecture de ce manuscrit.

Table des matières

Glossaire de termes biologiques

Acide aminé : Un acide aminé est une molécule organique qui forme l'unité structurale de base des protéines. Il comporte un groupement acide carboxylique et un groupement aminé.

Activateur : L'activateur est la protéine ou le complexe protéique permettant de lancer la première phase de la synthèse des protéines, la transcription d'un gène.

ADN : L'Acide Désoxyribonucléique est une molécule, présente dans toutes les cellules vivantes, qui renferme l'ensemble des informations nécessaires au développement et au fonctionnement d'un organisme. Il porte donc l'information génétique et constitue le génome des êtres vivants.

ARNm : L'Acide Ribonucléique messager est une copie transitoire d'une portion de l'ADN correspondant à un ou plusieurs gènes. L'ARNm est utilisé comme intermédiaire par des organites cellulaires, les ribosomes pour la synthèse des protéines.

Bases azotées : Les bases azotées sont des molécules qui font partie des nucléotides. Ce sont les molécules suivantes : l'Adénine (A), la Guanine (G), la Cytosine (C), la Thymine (T) (uniquement dans l'ADN) et l'Uracile (U) (uniquement dans l'ARN).

Codon : Un codon est un triplet de nucléotides A, C, U ou G de l'ARN messager.

Complexe : Un complexe protéique est le groupement formé par plusieurs protéines.

Cytoplasme : Le cytoplasme désigne le contenu d'une cellule vivante délimitée par la membrane plasmique et le noyau.

Dégradation : La dégradation protéique est la destruction naturelle des protéines au cours du temps.

E. coli : *Escherichia coli*, également appelé colibacille ou *E. coli*, est une bactérie intestinale des mammifères très commune chez l'être humain. Il s'agit de l'organisme le plus utilisé à l'heure actuelle pour le développement de biosystèmes.

Eucaryote : Organisme constitué d'une ou plusieurs cellules, qui possède un noyau dans lequel est présent l'ADN.

Génotype : Le génotype d'un organisme est l'ensemble ou une partie de son information génétique (le génome) contenue sur l'ADN.

IPTG : L'isopropyl β-D-1-thiogalactopyranoside est une molécule chimique, analogue au lactose, associée à la protéine répresseur LacI et utilisée comme activateur pour la synthèse des protéines.

LacI : Il s'agit d'une protéine répresseur, qui se fixe sur l'ADN et empêche la synthèse des ARN messagers.

microARNs : Ce sont des ARNs très courts, présents chez les organismes eucaryotes, qui bloquent la traduction d'un brin d'ARN en s'y fixant.

Nucléotides : Les nucléotides sont des molécules constituées d'un sucre (un ribose pour l'ARN ou un désoxyribose pour l'ADN), d'une base nucléique et d'au moins un groupement phosphate. Ils constituent l'unité de base des acides nucléiques.

Peptides : Un peptide est une chaîne comportant moins de 50 acides aminés reliés par des liaisons peptidiques. C'est donc une petite protéine résultant d'une traduction génétique, d'une synthèse non-ribosomale ou d'une synthèse chimique.

Phénotype : Le phénotype d'un individu est l'ensemble ou une partie de ses caractéristiques observables.

Procaryote : Organisme constitué d'une ou plusieurs cellules et dont l'ADN est présent directement dans le cytoplasme et est caractérisé par l'absence de noyau.

Promoteur : Le promoteur est une région située à proximité d'un gène et indispensable à la transcription, sur laquelle se fixe l'ARN polymérase.

Protéines : Une protéine est une macromolécule biologique composée par une ou plusieurs chaîne(s) d'acides aminés liés entre eux par des liaisons peptidiques.

Répresseur : Le répresseur est la protéine ou le complexe protéique bloquant la première phase de la synthèse des protéines, la transcription d'un gène. Il est en général dominant par rapport à l'activateur.

Ribosome : Les ribosomes sont des complexes ribonucléoprotéiques (c'est-à-dire composés de protéines et d'ARN). Leur fonction est de synthétiser les protéines en décodant l'information contenue dans l'ARN messager.

Traduction : La traduction est l'interprétation des codons de l'ARNm en acides aminés. C'est la deuxième étape de la synthèse des protéines qui traduit l'ARNm en protéine.

Transcription : La transcription est un processus biologique qui consiste, au niveau de la cellule, en la copie des régions de l'ADN en molécules d'ARN.

ZFP : Les ZFPs sont des Zinc Finger Proteins ou protéines doigt de zinc. Ce sont des protéines capables de reconnaître une séquence précise d'ADN. Il existe des ZFP naturelles et artificielles. On peut les fusionner avec des modules activateurs ou répresseurs de la transcription.

Glossaire de termes microélectroniques

Bottom-up : Méthode de conception de systèmes, consistant à assembler des composants standards pour former des sous-systèmes qui seront ensuite assemblés pour former le système entier.

Design kit : Librairie contenant l'ensemble des informations nécessaires à la conception d'un système dans une technologie donnée (librairie des éléments standards, fonctions associées, modèles, layouts et règles de conception).

EDA : Electronic Design Automation. Il s'agit d'un ensemble d'outils utilisés pour la conception de systèmes électroniques associés pour répondre aux besoins du flot de conception.

Flot de conception : Ensemble des étapes guidant le concepteur d'un système tout au long de sa réalisation. Il est standardisé et peut reposer sur plusieurs approches (bottom-up, top-down).

HDL : Hardware Description Language. Ce sont des langages de programmation spécialisés servant à décrire la structure et le fonctionnement de systèmes électroniques. Pour les circuits numériques, nous retrouvons le VHDL, le Vérilog et le SystemC parmi les plus utilisés. Ils possèdent chacun leur extension pour les parties analogiques et mixtes (AMS : Analog Mixed-Signal).

Layout : Masque correspondant aux composants électroniques élémentaires qui sera appliqué sur le silicium pour réaliser la gravure des transistors et des autres composants directement intégrés au silicium.

Modèle compact : Modèle dépendant généralement de peu de paramètres, qui se base sur des équations physiques approximées et permet d'avoir des résultats de simulation rapides et efficaces.

Netlist : Fichier dans lequel est décrit l'ensemble des interconnections entre les différents composants d'un système électronique.

Niveau d'abstraction : Niveau de représentation de composants ou de modèles associés, allant du haut niveau (abstraction numérique ou comportementale) au bas niveau (abstraction analogique).

Prototypage virtuel : Méthodologie consistant à réaliser une modélisation du système, appelée prototype virtuel, permettant de tester le système virtuellement sans avoir recours à un prototype physique.

Synthèse RTL : Register-Transfer Level. Il s'agit de l'élaboration d'une description des flux de données d'un système en utilisant des registres et des fonctions logiques.

Top-down : Méthode de conception de systèmes, en opposition à l'approche bottom-up, qui consiste à transformer les spécifications d'un système en sous-systèmes. L'assemblage du système peut être automatisé et il peut être considéré juste par conception.

Introduction

Née avec la révolution génétique des années 2000, la biologie synthétique est une science récente qui se situe à l'interface entre les biotechnologies et les sciences pour l'ingénieur. Elle vise à créer de nouveaux organismes par une combinaison rationnelle d'éléments biologiques standardisés, découplés de leur contexte naturel. Son but à long terme est de concevoir et de construire des systèmes biologiques qui traitent l'information, manipulent des éléments chimiques, produisent de l'énergie, fournissent de la nourriture et maintiennent ou encore améliorent la santé humaine ainsi que notre environnement. Ce domaine concerne non seulement les compétences des biotechnologues mais aussi celles de l'ingénierie des systèmes qui peut apporter beaucoup dans les méthodologies de conception.

Le travail de cette thèse se focalise ainsi sur les aspects liés à la conception *in-silico* des biosystèmes. Cette étape a été identifiée comme indispensable dans le processus de synthèse, mais elle n'est à l'heure actuelle que peu développée. Nous avons donc cherché à élaborer et structurer un flot de conception pour les biosystèmes, basé sur celui de l'électronique numérique.

Le flot de conception est l'ensemble des étapes permettant de standardiser la fabrication de systèmes et de garantir la réussite de leur réalisation, mais aussi d'assurer un gain de temps et de matériel employé. Le flot de conception utilisé dans la conception de systèmes numériques a fait ses preuves et tire parti de l'importante expérience que la micro-électronique a accumulé dans ce domaine. En effet, en l'espace de 40 ans, les processeurs sont passés d'à peine plus de 2000 transistors intégrés à plus de 2 milliards. Cette évolution a été rendue possible grâce à l'évolution des technologies mais aussi grâce à la méthodologie de conception mise en place. Bien que la biologie synthétique n'en soit qu'à ses débuts, et que les systèmes conçus soient relativement simples, des indicateurs semblent montrer que l'on tend vers une évolution rapide de leur complexité. Un flot de conception pensé en amont et adapté aux contraintes inhérentes à ce domaine permettra de faciliter la conception des biosystèmes et ainsi d'augmenter considérablement leur complexité et leur fiabilité.

Le travail réalisé pendant cette thèse s'est déroulé dans l'équipe de recherche Systèmes et Microsystèmes Hétérogènes (SMH) du laboratoire ICube (anciennement InESS). Une des thématiques de ce groupe concerne la modélisation compacte des dispositifs avancés dont le but est de développer des modèles compacts pour les dispositifs consacrés aux circuits analogiques et mixtes. Le principal sujet de recherche de ce groupe concerne la modélisation compacte des

transistors multigrilles ultimes, mais depuis 2008, l'équipe travaille aussi sur la modélisation des dispositifs biologiques. Elle y apporte son expérience dans la modélisation des composants élémentaires en l'appliquant aux mécanismes biologiques élémentaires, ainsi que ses connaissances dans la technologie de la conception pour le développement d'outils et l'exploitation des modèles.

En rejoignant l'équipe en 2010 pour mon stage de fin de master, j'ai particulièrement été attiré par l'aspect pluridisciplinaire et le caractère novateur de cette thématique de recherche. Les travaux effectués au cours de l'élaboration des premiers modèles m'ont donné envie de continuer la recherche dans cette voie et de réaliser cette thèse, dont les principaux résultats sont présentés dans ce document.

Ce manuscrit est divisé en trois parties principales, reprenant le travail qui a été effectué. La première partie, constituée des chapitres 1 à 3, correspond à la présentation de la biologie synthétique et des différentes méthodologies de conception de systèmes. La deuxième partie, allant des chapitres 4 à 7, concerne le flot de conception développé et les modèles sur lesquels il repose. Enfin, la troisième et dernière partie, composée des chapitres 8 à 10, aborde la question de l'amélioration des différents modèles.

Dans le premier chapitre, nous présentons le domaine de la biologie synthétique et son rôle au sein des biotechnologies. Nous abordons l'évolution de cette science au fil des années, et nous proposons plusieurs exemples de biosystèmes ayant déjà été réalisés, avec leurs domaines d'application respectifs. Comme tous les domaines ayant trait au vivant, la biologie synthétique implique des interrogations relatives à la bioéthique et aux risques sanitaires, comme la biosûreté et la biosécurité, mais aussi aux brevets sur le vivant, ainsi que nous pourrons le voir.

Avec la complexité grandissante des biosystèmes synthétiques, les outils d'aide à la conception vont jouer un rôle important, comme nous le voyons dans le chapitre 2. Après avoir présenté les différentes approches de conception génériques de système, comme l'approche « bottom-up » et l'approche « top-down », nous détaillons le flot de conception utilisé en microélectronique pour l'élaboration de systèmes numériques. Un état de l'art des différents outils d'aide à la conception de biosystèmes est ensuite présenté. Le développement d'un flot de conception dédié à la biologie synthétique à partir de ces outils semble complexe. Plutôt que de recréer tout l'environnement de conception, nous avons privilégié une approche qui vise à réutiliser les outils existants en microélectronique, tout en les adaptant au matériel biologique.

Le troisième chapitre correspond à une description des différents mécanismes pouvant être utilisés dans des biosystèmes. La liaison entre plusieurs molécules, la synthèse des protéines et l'exocytose et l'endocytose sont ainsi décrits dans une optique « concepteur ». La fonction utile

de ces mécanismes est ensuite présentée, avec une abstraction numérique de leurs comportements.

Le chapitre 4, en introduction à la deuxième partie, présente les quatre étapes du flot de conception pour la biologie synthétique proposé dans le cadre de cette thèse. Il s'inspire de la microélectronique. Ces étapes sont : la synthèse fonctionnelle, la synthèse biologique, la partie modélisation, simulation et optimisation et enfin la finalisation d'un biosystème.

Dans le chapitre 5 nous détaillons l'étape de synthèse fonctionnelle d'optimisation du système obtenu. Cette étape est réalisée à partir de la description du système qui est fournie au synthétiseur RTL (ODIN II). Le mapping et l'optimisation sont ensuite effectués par le logiciel ABC à l'aide d'une bibliothèque d'éléments biologiques standards. Le fonctionnement de cette étape est illustré à l'aide de deux exemples : une machine d'état et un processeur biologique.

Dans le chapitre 6, nous présentons les différents modèles des mécanismes biologiques développés. Ces mécanismes sont modélisés par des équations différentielles ordinaires. Les systèmes peuvent ainsi être décrits sous la forme de schéma-blocs et donc simulés par des simulateurs de type flux de signal. Ce type de modèle est simple à développer mais possède des limitations, en particulier lorsque le nombre de couplages et de rétroactions nécessaires entre les blocs augmente. Pour pallier ce problème, nous présentons une autre approche de modélisation, dite conservative, permettant de résoudre toutes les équations différentielles en parallèle. Pour faciliter la formalisation de ce modèle, nous avons transposé les équations des mécanismes sous la forme d'un réseau électrique. L'implémentation de ces modèles à l'aide de deux langages de description matérielle utilisés en ingénierie des systèmes, le VHDL-AMS et le SystemC-AMS, est ensuite détaillée. Enfin, la génération automatique des modèles est illustrée sur plusieurs biosystèmes provenant de la littérature et permettant de montrer leur robustesse.

Le chapitre 7 est centré sur la modélisation du comportement des mécanismes biologiques à l'aide de la logique floue. Elle permet la description d'un système par le biais d'une approche discrète tout en fournissant des résultats quantitatifs, et sert à faire le lien entre l'abstraction numérique d'une fonction biologique et sa modélisation bas niveau. Après avoir décrit le fonctionnement de la logique floue, nous présentons le cœur de calcul développé et son utilisation dans la modélisation et l'aide à la conception de deux biosystèmes.

Dans le chapitre 8, nous présentons le premier travail effectué sur l'amélioration des modèles destinés à la conception, permettant de représenter de façon plus réaliste les interactions entre plusieurs molécules. Il s'agit d'une méthodologie générique, basée sur un polynôme de liaison, permettant de retrouver et d'unifier les modèles existants. Son application est illustrée grâce au cas de la calmoduline liant des ions calcium.

En pratique, les grandeurs biologiques n'évoluent pas idéalement, mais contiennent un aspect aléatoire. Le chapitre 9 est donc consacré à l'étude du bruit présent dans les mécanismes biologiques. Nous montrons comment une analogie peut être faite avec certains bruits électroniques, comme pour les bruits de grenaille, de phase et de quantification. Les modèles de ces bruits développés et intégrés aux modèles bas-niveau sont ensuite détaillés.

Dans le dernier chapitre, nous présentons un simulateur de fonctionnement de la cellule, qui a également été développé au cours de cette thèse. Il reprend le principe du « jeu de la vie », et l'applique aux mécanismes biologiques. Le fonctionnement du simulateur est tout d'abord introduit, puis ses résultats sont analysés, permettant ainsi de valider partiellement les modèles précédemment développés et d'aller vers la mise au point de nouveaux modèles plus performants.

Première partie

Présentation de la biologie synthétique

Chapitre 1

Introduction à la biologie synthétique

En recherchant la définition du principe de la biologie synthétique, nous pouvons tomber sur plusieurs descriptions, plus ou moins similaires. Parmi les plus pertinentes, nous retenons un extrait de celle proposée par le Synthetic Biology Engineering Research Center (Synberc) [1], un consortium regroupant les universités UC Berkeley, UC San Francisco, Stanford, Harvard et le MIT et qui résume le point du vue nord-américain sur cette science :

« Synthetic biology is the design and construction of new biological entities such as enzymes, genetic circuits, and cells or the redesign of existing biological systems. Synthetic biology builds on the advances in molecular, cell, and systems biology and seeks to transform biology in the same way that synthesis transformed chemistry and integrated circuit design transformed computing … »

Du côté européen, le projet SynBiology, faisant partie du programme NEST (New and Emerging Science and Technology) de la Commission Européenne, a eu pour but d'analyser les recherches effectuées sur le thème de la biologie synthétique en Amérique du Nord et en Europe [2]. Le rapport final de ce projet définit la biologie synthétique de la manière suivante :

« Synthetic biology is the engineering of biological components and systems that do not exist in nature and the re-engineering of existing biological elements; it is determined on the intentional design of artificial biological systems, rather than on the understanding of natural biology. »

Enfin, au niveau national, le ministère français, par l'intermédiaire du site « Biologie de Synthèse » [3], définit ainsi la biologie de synthèse (terme également employé pour désigner la biologie synthétique) :

« La biologie de synthèse est un domaine en pleine émergence. C'est l'ingénierie rationnelle de la biologie, son but est de concevoir de nouveaux systèmes biologiques. Elle fera progresser les connaissances du monde du vivant et permettra de développer de nombreuses applications industrielles dans les domaines de la santé, de l'énergie, des matériaux, de l'environnement et de l'agriculture. »

Notre vision de la biologie synthétique consiste à considérer cette science comme de l'ingénierie sur du matériel biologique, c'est-à-dire la création de fonctions biologiques, et, à terme, d'organismes nouveaux, par une combinaison rationnelle d'éléments biologiques standardisés, découplés de leur contexte naturel.

Ces différentes définitions permettent de mieux cerner le fonctionnement ainsi que le rôle de la biologie synthétique dans le monde de la recherche et de l'industrie. Selon une étude réalisée par BBC Research en 2011 [4] et appuyée par une étude réalisée par Transparency Market Research fin 2012 [5], le marché mondial des produits de la biologie synthétique valait 1,1 milliard de dollars en 2010 et a connu une croissance importante pour atteindre 2,12 milliards de dollars en 2012. Les prévisions indiquent que sa valeur dépassera les 10 milliards de dollars d'ici 2016 pour arriver à un marché de plus de 16 milliards de dollars en 2018, soit une croissance de 40 à 45% par an. Pour l'instant, le marché de la biologie synthétique consiste principalement dans l'élaboration de nouvelles énergies renouvelables, mais ces chiffres sont encourageants et indiquent un développement rapide des activités liées à la biologie synthétique.

Dans ce chapitre, nous allons tout d'abord présenter un historique de la biologie synthétique avant de repositionner ce domaine à l'interface entre les biotechnologies et les sciences pour l'ingénieur. L'accent sera notamment mis sur les approches de conception. En effet, elles représentent l'un des défis principaux dans le développement de biosystèmes complexes. Nous illustrerons ensuite le potentiel de cette nouvelle technologie par quelques exemples de biosystèmes réalisés dans différents domaines d'application. Enfin nous aborderons les épineuses questions d'éthique entourant cette nouvelle science, les risques sanitaires inhérents au développement de nouveaux biosystèmes synthétiques et nous conclurons par un bref point sur la propriété intellectuelle.

1.1 Historique

La biologie synthétique a bénéficié des avancées considérables réalisées dans le domaine des biotechnologies et a pu, à partir d'un concept dont les bases ont été posées il y a un siècle, devenir une science à part entière. Ce développement a été rendu possible notamment grâce à deux révolutions majeures : la naissance du génie génétique, puis le séquençage du génome.

1.1.1 Apparition du concept

La première occurrence du terme de biologie synthétique dans la littérature est attribuée au français Stéphane Leduc, qui publia deux ouvrages abordant ce concept. En 1910, il écrit « Théorie physico-chimique de la vie et générations spontanées » [6], résultat de ses recherches visant à recréer des formes ressemblant à des organismes vivants en mélangeant des sels

métalliques dans des solutions de carbonate de potassium. En 1912, il publie un nouvel ouvrage appelé sommairement « La biologie synthétique » [7], dans lequel il expose un peu plus ses idées sur la nécessité de devoir fabriquer ou « synthétiser » des objets ou des phénomènes biologiques afin de valider les connaissances acquises en biologie. Il affirme ainsi que « *La biologie est une science comme les autres,* (…) *elle doit être successivement descriptive, analytique et synthétique.* » ce qui pour l'époque fait preuve d'une rare clairvoyance.

Bien que ses travaux correspondent en réalité uniquement à l'observation des effets d'osmose et ont été par la suite contestés, il pose les bases d'un domaine nouveau qui ne trouvera ses développements que 60 ans plus tard. En 1912, Jacques Loeb publie lui aussi le résultat de ses travaux sur la pathogénèse artificielle, qui sera par la suite l'une des briques de base de la biologie actuelle [8].

1.1.2 La naissance du génie génétique

Il faut attendre 1974 pour que le généticien polonais Wacław Szybalski reparle de la biologie synthétique dans l'une de ses publications [9] puis déclare que les travaux des prix Nobel de médecine Werner Arber, Daniel Nathans et Hamilton O. Smith sur les enzymes de restrictions vont constituer une avancée majeure dans le domaine de la biologie synthétique [10]. Le développement du génie génétique dans les années 1970-1980 va effectivement bouleverser la perception de la biologie. Parmi les découvertes les plus remarquables et fondatrices, nous pouvons citer les techniques d'ADN recombinants, développées par Stanley Cohen et Herbert Boyer en 1973 [11]. Il s'agit de l'une des deux méthodes utilisées pour réaliser la réplication directe de n'importe quelle séquence d'ADN. La deuxième est la technique de la réaction en chaîne par polymérase (PCR) inventée par l'équipe de Kary Mullis en 1984 [12].

1.1.3 Le séquençage des génomes et les premiers systèmes synthétiques

Le séquençage de l'ADN, c'est-à-dire la mesure ou l'observation de l'enchaînement des nucléotides constituant les brins d'ADN, a été inventé vers la fin des années 1970 par les équipes de Walter Gilbert [13] et de Frederick Sanger [14]. Les deux équipes utilisèrent des techniques diamétralement opposées, la première se basant sur une méthode par dégradation chimique sélective alors que la seconde employait une méthode par synthèse enzymatique sélective. Ces techniques, appliquées par la suite à des génomes entiers, ont permis d'avoir une vision globale du vivant concernant le lien existant entre une séquence d'ADN, une protéine synthétisée et la fonction de cette protéine.

En 2000, les premiers biosystèmes synthétiques commencent à apparaître. Les recherches portent tout d'abord sur des oscillateurs synthétiques. Ainsi l'oscillateur d'Elowitz et Leibler [15] est considéré comme l'un des premiers biosystèmes synthétiques réalisés. Par la suite, les biosystèmes se sont diversifiés et répondent maintenant à des besoins par des fonctions très variées.

1.1.4 Les années 2000, un premier pas vers les sciences de l'ingénieur

Le début des années 2000 correspond également à un nouvel essor de la biologique synthétique, avec le début d'une collaboration efficace entre les biotechnologues et les spécialistes des sciences pour l'ingénieur. Ce rapprochement a été initié dans le but de réduire les difficultés rencontrées par les acteurs du domaine lors des étapes en amont menant à la conception des biosystèmes synthétiques. Il semble en effet possible d'exploiter les similitudes entre ces deux domaines. Les premiers travaux de recherche remarquables à cette interface ont été réalisés par des équipes du MIT et nous pouvons citer comme principaux acteurs de ce rapprochement Drew Endy, Tom Knight, Randy Redberg et Christopher Voigt.

Outres les aspects technologiques, une grande partie des contributions de ce début de XXI^{ème} siècle concerne la mise en place de règles de description, de formalismes et de standards permettant de manipuler les objets biologiques à assembler pour réaliser des systèmes artificiels. Ces concepts ne sont pas naturels dans la culture des biologistes dont la démarche scientifique était, jusque-là, plus descriptive que prédictive. Dans une vision idéalisée, la biologie synthétique deviendrait une sorte de jeu de Lego® où la conception de nouveaux systèmes serait basée sur l'assemblage de briques élémentaires, les BioBriques, introduites en 2003 par Tom Knight [16].

Par extension de ces travaux, Drew Endy décrit de manière exhaustive la *« boîte à outil du concepteur en biologie synthétique »* [17], dont les concepts sont proches de ceux que nous connaissons dans beaucoup de domaines des sciences pour l'ingénieur. En particulier, nous retrouvons en microélectronique des outils et des bibliothèques de composants fournis par les fondeurs qui sont réputés pour être très performants.

1.1.5 Après les années 2000

Les progrès constants dans les biotechnologies, associés à un engouement pour le domaine de la biologie synthétique, la rendent de plus en plus populaire. Une étude a été réalisée par Peccoud portant sur l'évolution des publications consacrées à la biologie synthétique publiées dans le

journal PLoS ONE depuis la création de cette revue en 2006, montrant l'évolution de ce domaine pendant 6 années [18]. Les résultats de cette étude sont illustrés Figure 1.1.

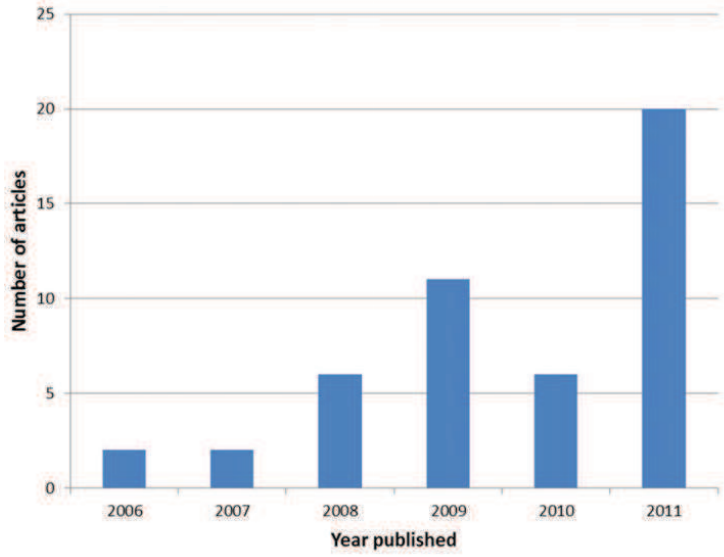


Figure 1.1 : Répartition du nombre d'articles relatifs à la biologie synthétique dans PLoS ONE, en fonction des années (source Peccoud et Isalan [18]).

Nous avons repris le même principe, et dénombré les publications référencées sur le métamoteur de recherche de publications scientifiques ScienceDirect, pour les termes « Synthetic Biology ». Nous les avons classés par date de publication et nous avons illustré ce classement par l'histogramme Figure 1.2.

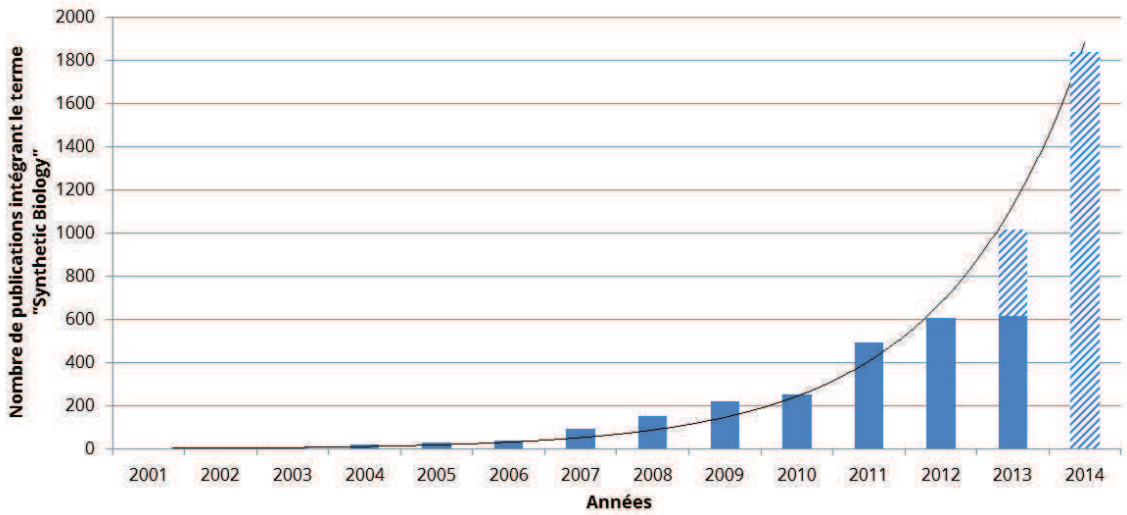


Figure 1.2 : Histogramme du nombre de publications répondant aux termes « Synthetic Biology » sur ScienceDirect et classées par années. Les résultats représentés par des stries bleues correspondent aux prévisions pour l'année 2013 (la mesure ayant été réalisée en août, un calcul par rapport aux recherches déjà publiées, en bleu uni, et le nombre de jours restants a été effectué), et l'année 2014.

Même si certaines publications ne sont pas référencées, nous constatons une forte augmentation. Il est également important de voir que le spectre de ces publications s'est également élargi au cours des années, couvrant maintenant des recherches fondamentales visant à mieux comprendre les mécanismes biologiques élémentaires, les modèles, les outils et les méthodes de conception et de nombreuses applications émergentes. Cette tendance suit l'évolution du nombre de nouveaux biosystèmes réalisés par les différentes équipes de recherche à travers le monde ainsi que la complexité des systèmes réalisés.

1.2 L'évolution temporelle des sciences du vivant

Pour résumer l'évolution temporelle des sciences du vivant, nous reprenons le concept énoncé par Stéphane Leduc d'une science passant successivement par des étapes descriptives, analytiques puis synthétiques. Nous l'extrapolons afin d'obtenir le graphique Figure 1.3 présentant cette évolution temporelle et le transfert technologique associé entre la biologie en tant que science fondamentale vers des produits et des services en passant par les biotechnologies.

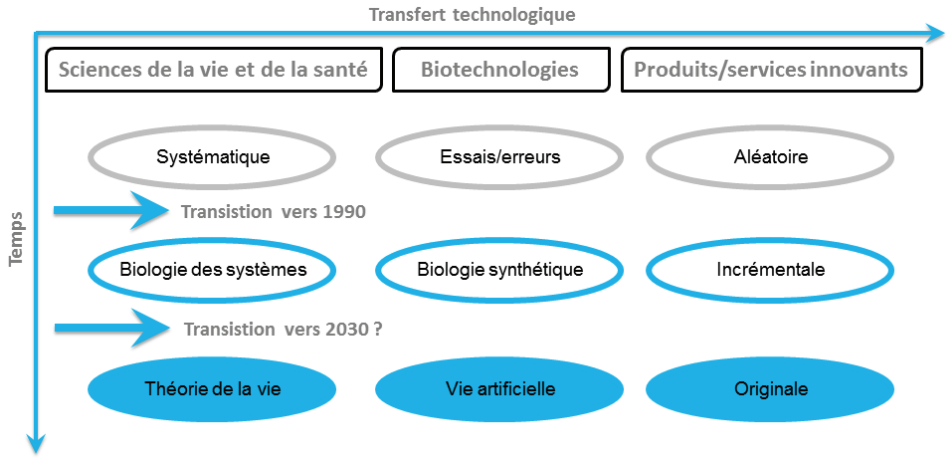


Figure 1.3 : Evolution temporelle des sciences du vivant et transfert technologique associé (source J. Haiech et al. [19]).

1.2.1 Biologie traditionnelle et biotechnologies

La biotechnologie constitue donc une première étape de transfert de technologie de la biologie « traditionnelle » vers l'application, visant ainsi à utiliser les connaissances acquises dans le domaine du vivant pour la production de biens et de services. Malgré ce transfert, il est cependant important de constater que la biologie traditionnelle et la biotechnologie continuent de coexister et de se développer conjointement. L'exemple classique illustrant cette synergie est celui de la levure et du pain. Le pain existe depuis l'Antiquité et les boulangers en connaissaient

la technique de fabrication, avant même de savoir ce qu'était la levure. La biologie traditionnelle a pu ensuite décrire ce qu'est une levure et l'action qu'elle peut avoir dans la fabrication du pain. La recherche conjointe, en amont et applicative, a ensuite permis de mettre au point d'autres types de levures, plus performantes ou moins coûteuses.

Le domaine des biotechnologies intègre des compétences de nombreuses disciplines, comme la biologie moléculaire, la biochimie, la microbiologie et la génétique, pour les sciences purement biologiques, mais aussi la chimie et l'informatique, ainsi que les sciences de l'ingénieur pour la conception. En règle générale, la biotechnologie implique la nécessité de modifier le vivant pour le faire évoluer vers un nouveau biosystème qui répond mieux à un besoin identifié. La branche des biotechnologies qui nous intéresse particulièrement dans cette thèse est le génie génétique où l'objectif est de modifier le patrimoine génétique d'un organisme afin d'améliorer sa fonction. Il existe trois approches possibles dans ce domaine :

- La mutagénèse, qui consiste à faire subir à un organisme un traitement (physique, chimique…) favorisant les mutations génétiques afin d'obtenir de nouvelles fonctionnalités. Les mutations, correspondant à la modification d'une partie de l'ADN ou de l'ARN, se produisent naturellement et peuvent être de plusieurs types (substitution, insertion ou suppression d'un nucléotide par exemple). Associées à la théorie de la sélection naturelle (entre un organisme original et un organisme muté, celui qui aura tendance à survivre et à proliférer est celui qui sera le plus adapté à son environnement), les mutations génétiques sont à la base de l'évolution des organismes vivants. En mutagénèse, la probabilité d'apparition de ces mutations est augmentée, soit de manière aléatoire, soit de manière dirigée, et la sélection naturelle est favorisée dans le but de développer de nouveaux organismes présentant des nouvelles fonctions intéressantes.
- La transgénèse, qui correspond à l'insertion d'un transgène (une séquence isolée d'un gène), provenant d'un premier organisme, dans un second organisme, afin de lui apporter une nouvelle propriété. L'application de cette technique la plus connue est celle qui est faite dans le cadre de la création d'organismes génétiquement modifiés (OGM), en particulier dans l'agroalimentaire. Elle est constituée de plusieurs étapes : le gène à implanter est tout d'abord identifié puis encapsulé afin qu'il puisse être exprimé (ajout d'un promoteur par exemple), et ensuite ajouté à l'organisme cible par transfection (par projection de microbilles contenant le transgène, par utilisation d'un vecteur biologique, comme des virus par exemple, ou par injection directe dans la cellule). La sélection des organismes exprimant correctement le transgène, et donc qui possèdent la fonctionnalité souhaitée, est finalement réalisée à l'aide d'antibiotiques et d'une résistance à ces antibiotiques intégrée dans la construction plasmidique portant le transgène. Seuls les

organismes au sein desquels le plasmide a été correctement implanté vont être résistants aux antibiotiques alors que les autres organismes vont mourir.

- La biologie synthétique, pour laquelle nous passons d'une démarche d'innovation par essais-erreurs à une démarche rationnelle de conception de nouveaux organismes répondant à une fonction souhaitée, comme nous l'avons vu précédemment.

Les biotechnologies peuvent aussi être regroupées en fonction de leur domaine d'application et identifiées à l'aide de « couleurs ». Nous retrouvons ainsi les biotechnologies « vertes » se référant aux applications agricoles, les biotechnologies « rouges » appliquées au domaine de la santé, les biotechnologies « blanches » permettant une application industrielle des procédés naturels, les biotechnologies « jaunes » pour la protection de l'environnement, les biotechnologies « bleues » pour les applications en rapport avec le milieu marin. D'autres domaines d'application devraient également voir le jour dans le futur.

1.2.2 Approches utilisées

Les travaux de recherche et de développement sur la mise au point de nouveaux organismes biologiques se font aujourd'hui selon deux approches qui coexistent : l'approche « top-down » et l'approche « bottom-up ». Cette terminologie est bien connue des spécialistes des sciences pour l'ingénieur. Toutefois, le sens de ces termes n'est pas tout à fait le même lorsqu'on parle de biologie ou de système électronique. Nous allons donc dans un premier temps définir les termes « top-down » et « bottom-up » dans le contexte de la biologie synthétique, avant de discuter des approches de conception, telles que définies dans les sciences pour l'ingénieur.

1.2.2.1 L'approche « top-down »

L'approche « top-down » est l'approche historique dans la conception de nouveaux systèmes synthétiques, utilisée depuis les années 70. Elle consiste à partir d'un organisme biologique existant naturellement (« top ») et de simplifier (« down ») les différentes fonctions qui le composent en les retirant, jusqu'à l'obtention d'un nouvel organisme dont les mécanismes et le fonctionnement sont plus faciles à comprendre. L'utilisation de cette approche permet de disposer d'un organisme châssis sur lequel de nouvelles fonctions peuvent ensuite être ajoutées ou retirées en fonction de l'application recherchée.

L'exemple le plus important de la littérature utilisant cette approche est l'utilisation de l'organisme *Mycoplasma genitalium* par l'institut J. Craig Venter [20]. Le génome de cet organisme est constitué de 470 gènes et est considéré comme l'un des organismes naturels ayant le plus petit génome. L'équipe de J. Craig Venter a simplifié le génome de cet organisme au fil des ans pour former la lignée des *Mycoplasma laboratorium*. Récemment, l'équipe a réussi à intégrer un

génome synthétique dans l'organisme *Mycoplasma mycoides JCVI-syn1.0* [21], composant ainsi la base de nouveaux organismes capables de produire des composés chimiques sur mesure.

1.2.2.2 L'approche « bottom-up »

L'approche inverse, appelée « bottom-up » est plus récente et généralement considérée comme plus complexe. Elle consiste à partir de structures génétiques élémentaires (« bottom ») que l'on assemble hiérarchiquement (« up ») jusqu'à former un système fonctionnel à intégrer dans un organisme hôte, voire à créer un organisme vivant totalement synthétique, possédant les fonctions des éléments génétiques ajoutés. A l'instar des Lego®, l'objectif de cette approche est de développer un ensemble d'éléments synthétiques de base possédant des fonctionnalités identifiées et prévisibles, agissant chez divers organismes hôtes.

Les BioBriques ont été développées dans cette optique par Tom Knight, Drew Endy et Christopher Voigt au MIT en 2003 [16]. Il s'agit de séquences d'ADN standardisées correspondant à une fonction définie. Elles sont conçues pour pouvoir être intégrées dans des organismes tels que les bactéries (comme *E. coli*) ou d'autres sortes d'organismes hôtes. Il existe trois niveaux de BioBriques, classés selon leur complexité :

- les « Parts », qui sont les blocs de base et codent pour des fonctions biologiques élémentaires (les promoteurs ou les gènes),
- les « Devices », qui sont un assemblage de plusieurs « Parts » et correspondent à des fonctions plus complexes,
- les « Systems », servant à effectuer des tâches de plus haut niveau.

Les codes ADN correspondant à ces BioBriques sont disponibles librement sur le site internet de la BioBricks Foundation (http://partsregistry.org). La base de données contenant les BioBriques est régulièrement alimentée par le concours iGEM (International Genetically Engineered Machine competition) [22]. Ce concours est destiné aux étudiants de niveaux d'étude inférieurs au master, issus des cursus de biologie mais aussi d'autres domaines, la biologie synthétique étant par essence transdisciplinaire. Plus de 100 équipes à travers le monde s'affrontent chaque année afin de développer de nouvelles BioBriques par le biais de projets innovants.

1.2.2.3 L'approche de conception des sciences pour l'ingénieur, appliquée à la biologie synthétique

L'approche « top-down » décrite ci-dessus est, par nature, plus expérimentale et basée sur un principe d'essais-erreurs. Bien que des outils aient été mis au point pour faciliter les décisions et

prévoir le comportement des cellules obtenues, cette approche reste très loin des problématiques abordées par les sciences pour l'ingénieur. En réalité, de la même manière qu'un wafer de silicium est à la base de la construction d'un circuit électronique, une cellule minimaliste issue de l'approche « top-down » pourrait être la base de la construction de nouveaux systèmes biologiques.

L'approche « bottom-up » correspond davantage à ce qui est connu en microélectronique ou dans d'autres domaines proches. Le terme « bottom-up » se réfère alors au support physique et au matériel utilisé, mais en aucun cas à la démarche de conception utilisée pour choisir les briques élémentaires à assembler, qui peut varier. Dans ce travail de thèse, nous allons mettre à profit certaines propriétés des mécanismes biologiques, et en particulier la possibilité de les décrire avec une abstraction numérique (voir chapitre 3), pour essayer de développer une démarche de conception « top-down », au sens science pour l'ingénieur, pour accompagner l'approche de conception « bottom-up » au sens biologique. Le principe retenu est celui du prototypage virtuel, c'est-à-dire la capacité d'effectuer toutes les étapes de conception *in silico* (depuis la description haut-niveau jusqu'aux briques élémentaires), plutôt que *in vivo* par essais-erreurs. Cette démarche sera décrite plus en détail au chapitre 2.

1.2.3 Perspectives d'évolution

En électronique, l'évolution de la complexité des systèmes réalisés a été prédite par la loi de Moore dès 1965 [23]. Plusieurs variations de cette loi existent mais à l'origine celle-ci dictait que la densité des transistors dans les circuits intégrés doublerait tous les ans. Par la suite elle a été ramenée à un doublement tous les deux ans. En biotechnologie, une loi similaire montrant la diminution du coût pour le séquençage de l'ADN au fil du temps a été énoncée par Carlson en 2009 [24]. Cette tendance a été comparée à des données hypothétiques tenant compte de la loi de Moore et actualisée en 2013. Nous retrouvons la comparaison de ces deux courbes Figure 1.4.

Bien que le séquençage de l'ADN ne couvre qu'une partie des technologies à maîtriser pour la biologie synthétique, ce comportement laisse tout de même à penser que la biologie synthétique pourrait connaître, dans un proche avenir, une complexification des systèmes conçus, suivant un essor au moins aussi important que celui qu'a connu la microélectronique depuis 40 ans.

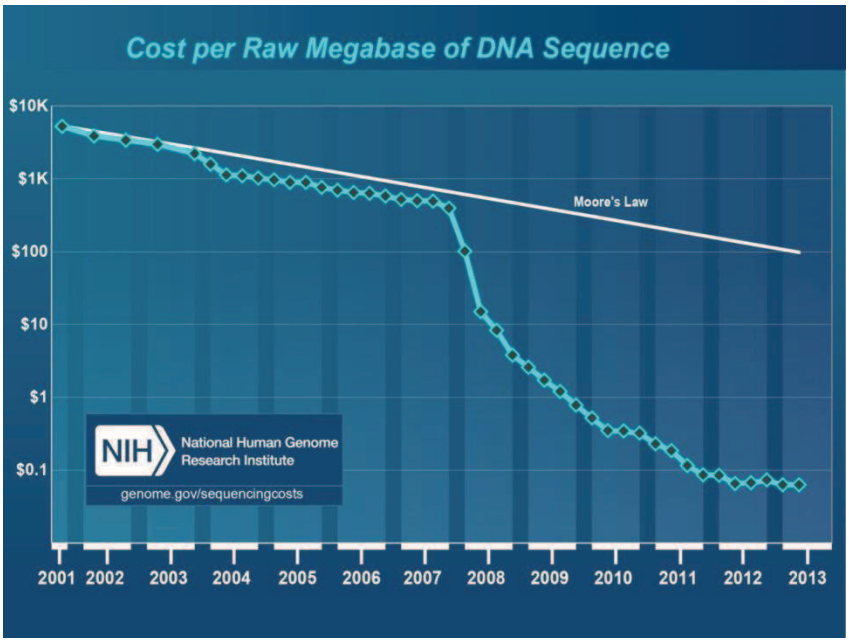


Figure 1.4 : Loi de Carlson comparée à la loi de Moore pour le coût du séquençage de l'ADN [24].

1.3 Exemples d'applications

Nous pouvons classer les biosystèmes synthétiques développés en fonction de leur domaine d'application, à savoir l'environnement, l'agroalimentaire et la santé [25]. Les travaux cités dans ce chapitre ne constituent pas un état de l'art exhaustif des projets en biologie synthétique mais quelques exemples d'applications dans les différents domaines principaux.

1.3.1 L'environnement

Le domaine de l'environnement est particulièrement porteur à l'heure actuelle, et beaucoup de moyens lui sont consacrés. L'application de la biologie synthétique dans ce domaine va permettre le développement de carburants verts et de capteurs biochimiques de nouvelle génération. C'est le cas d'un détecteur d'arsenic à base de bactéries qui modifient le pH d'une eau contaminée, rendant sa détection peu coûteuse [26]. Ces futurs biosystèmes pourront même mener à une maîtrise directe de différents types de pollutions, avec par exemple la dégradation de certains alcanes [27]. Bien que nous n'ayons pour l'instant pas connaissance dans la littérature de biosystèmes permettant de générer des énergies renouvelables à grande échelle, de nombreuses recherches pour la production de biocarburants sont également en cours depuis une dizaine d'années [28], [29].

Le premier exemple de biosystème présenté est celui d'un capteur de polluant du pétrole réalisé à partir d'une bactérie modifiée par l'équipe de Simpson [30]. La bactérie *Pseudomonas fluorescens* HK44, initialement décrite en 1990, a été le premier micro-organisme génétiquement

modifié servant de rapporteur biologique du naphtalène, un des principaux polluants du pétrole. Cette fonction est possible grâce à un phénotype bioluminescent directement lié à la dégradation du naphtalène. Cette bactérie modifiée a ensuite été intégrée à un photodétecteur CMOS par l'équipe de Simpson, grâce à une encapsulation dans un biofilm. Le capteur pose certains problèmes liés à l'utilisation de matériel vivant, comme la survie de la bactérie, mais permet de mesurer le polluant de manière autonome avec exactitude.

Bien que cet exemple soit assez ancien, puisqu'il date de 1998, il s'agit de l'un des premiers biosystèmes pouvant être considéré comme un produit de la biologie synthétique et présente donc un aspect historique. Depuis, les biocapteurs sont devenus plus complexes et permettent de mesurer la présence d'autres molécules polluantes, comme l'arsenic.

1.3.2 L'agroalimentaire

L'agroalimentaire est le deuxième domaine pouvant bénéficier des avancées de la biologie synthétique. Au vu du contexte actuel sur la qualité de la nutrition et à cause des réticences concernant les OGMs, la production de nouveaux aliments par des systèmes biosynthétiques n'a pas encore retenu l'intérêt des diverses équipes de recherche travaillant sur la biologie synthétique. Cependant, au niveau agricole, d'importantes retombées sont à prévoir avec, par exemple, des biosystèmes de tests de la qualité d'un sol permettant d'améliorer le rendement tout en économisant les additifs comme l'engrais, illustré par le travail réalisé par l'équipe de l'université de Bristol dans le cadre de l'iGEM 2010 [31].

L'exemple applicatif choisi pour le domaine agroalimentaire concerne l'insémination de la vache. Malgré les tests effectués par les vétérinaires, l'insémination actuelle ne permet d'obtenir que 60% de succès car il est difficile de prévoir l'ovulation. De plus, les ovules et les spermatozoïdes ont une durée de vie très courte (quelques dizaines d'heures seulement). Le biosystème développé par l'équipe de Fussenegger [32] est constitué de microcapsules renfermant le système ainsi que le sperme de taureau, le protégeant des défenses immunitaires de la vache. Le signal choisi pour déclencher le système est l'hormone lutéinisante (LH) qui est produite par la vache au moment de l'ovulation. Le taux de LH dans le sang présente un pic de quelques heures autour de l'ovulation. Le biosystème a été conçu avec des capteurs de LH qui vont déclencher la production d'une enzyme qui dissout les capsules de cellulose jusqu'à leur totale disparition. Les spermatozoïdes sont ainsi libérés et peuvent aller féconder l'ovule. Ces différentes étapes sont illustrées Figure 1.5.

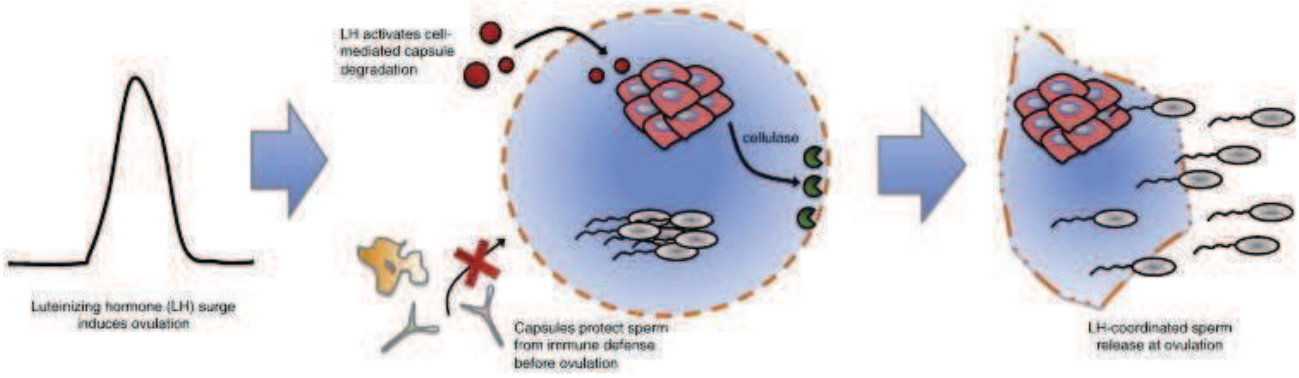


Figure 1.5 : Biosystème d'aide à l'insémination chez la vache (source : Kemmer et al. *[32])*

Les capsules étant très petites, les instruments utilisés pour l'insémination restent les mêmes que pour une insémination classique. Le système proposé a permis d'augmenter considérablement le taux de succès de l'insémination pour atteindre quasiment 100% et un dépôt de brevet est actuellement en cours.

1.3.3 La santé

Le dernier domaine dans lequel la biologie synthétique va permettre des évolutions importantes est celui de la santé et de toutes les technologies associées. Les champs d'application sont très variés. Nous pensons notamment au développement de nouveaux types de médicaments intelligents pouvant détecter une carence chez l'hôte et déclencher la synthèse de la molécule manquante [33], [34]. Il est également envisageable d'utiliser des biosystèmes pour détecter ou cibler des maladies difficilement décelables ou curables (comme le cancer) par les méthodes classiques [35], [36], ou d'améliorer les méthodes classiques de détection comme l'IRM [37], ou encore pour synthétiser des molécules à bas coût.

L'exemple illustrant cette application pour le domaine de la santé entre totalement dans cette dernière catégorie. Les terpénoïdes sont des composés organiques appartenant à la classe des isoprénoïdes, présents naturellement chez certaines plantes. Ils sont entre autres utilisés dans le traitement du paludisme sous la forme de l'artémisinine, un composé terpénique. Cette molécule est présente naturellement en concentrations très faibles. De plus, la synthèse de la molécule d'artémisinine est coûteuse, ce qui est particulièrement problématique dans les pays en développement. Le système développé par l'équipe de Keasling [38] a pour but la production de terpénoïdes par la bactérie *E. coli* ou par une levure. La modification de la bactérie *E. coli* a permis de synthétiser cette molécule à moindre coût.

1.4 Bioéthique et sécurité

Bien que notre équipe n'intervienne qu'au niveau de la conception théorique de biosystèmes, il est important de prendre en compte les notions d'éthique ainsi que les risques biologiques liés au développement de nouveaux biosystèmes synthétiques. A l'heure actuelle, des éléments de lois nationales peuvent s'appliquer en théorie à divers aspects de la biologie synthétique, comme certains passages de la loi bioéthique de 2004 ainsi que quelques-unes des dispositions prises pour la réglementation des OGMs. Cependant, il n'existe aucun cadre, aussi bien à l'échelle nationale, européenne qu'internationale, permettant la supervision et la prise en charge complète de ce domaine scientifique.

L'un des processus important à mettre en place est une vulgarisation vis-à-vis de la biologie synthétique. De nombreuses personnes n'ont jamais entendu parler de ce domaine scientifique ou en ont une vision négative ou erronée car ils le confondent avec les OGMs ou avec le clonage. Le principe même d'évaluation des risques utilisé pour les OGMs, comme le concept d'équivalence en substance pour les aliments (qui spécifie qu'un composé alimentaire essentiellement semblable à un composé existant peut être traité de la même manière par rapport à ses risques), ne sont plus du tout pertinents dans le cadre de la création de biosystèmes totalement nouveaux comme c'est le cas en biologie synthétique. Nous allons donc faire le tour des différentes questions qui se posent et tenter d'apporter des pistes pour y répondre.

1.4.1 Ethique

Les premières questions à se poser à propos de la biologie synthétique concernent bien sûr l'éthique ainsi que l'aspect social qu'engendre l'apparition de nouvelles sciences ayant trait au vivant. Avec le développement des premiers organismes possédant un génome entièrement synthétique et la reprogrammation de bactéries pour réaliser de nouvelles fonctions, notre conception du vivant est fortement bouleversée. Il devient donc indispensable de poser des règles pour une pratique de la biologie synthétique conforme à l'éthique. Le site du gouvernement français dédié à la promotion de la biologie de synthèse encourage l'innovation responsable [3]. Le Conservatoire National des Arts et Métiers (CNAM) a aussi mis en place un site internet pour répondre aux questions du public [39], et organise des rencontres sous la forme de forums permettant la discussion et le débat entre citoyens et professionnels. D'autres rapports très complets ont été rendus sur les questions d'éthique et de sécurité qui ont été soulevées depuis l'avènement de cette science.

Parmi les plus intéressants pour l'Europe, nous retenons le rapport mandaté par José Manuel Barroso, le président de la Commission Européenne, qui a sollicité l'avis du Groupe Européen d'Ethique des Sciences et Nouvelles Technologies (GEE) sur l'éthique de la biologie synthétique en 2009 [40]. Le GEE a ainsi affirmé que *« la biologie synthétique soulève des questions éthiques fondamentales qui nécessitent une analyse conceptuelle de la vie et de la nature »* et aborde les questions liées aux inquiétudes par rapport aux risques sanitaires ainsi qu'à la protection intellectuelle.

Le deuxième rapport concerne les Etats-Unis, où le président Barack Obama a demandé à la Commission Présidentielle pour l'Etude des Problèmes Bioéthiques (PCSBI) d'examiner le domaine de la biologie synthétique et d'identifier les limites éthiques appropriées afin de maximiser les bénéfices publiques et de minimiser les risques [41]. Ce rapport est très intéressant au niveau des retombées envisagées dans les différents domaines tels que les énergies renouvelables, la santé ou encore l'agriculture, la nourriture et l'environnement. Il permet de faire le tour des différents problèmes liés aux risques et propose des solutions potentielles. Nous aborderons une partie de ces solutions dans la suite de ce chapitre.

1.4.2 Risques sanitaires

En termes de risques sanitaires et d'impact de la biologie sur la sécurité, nous pouvons distinguer deux domaines d'étude. Le premier est la biosûreté avec la mise en place de processus efficaces afin de prévenir toute contamination et infection accidentelle, ou, le cas échéant, de les confiner. Le deuxième est la biosécurité avec l'instauration de mesures de protection efficaces permettant d'empêcher que des agents biologiques dangereux puissent être obtenus, manipulés et/ou transformés en arme biologique, dans l'intention délibérée de nuire.

1.4.2.1 Biosûreté

Les risques de biosûreté peuvent être quantifiés en fonction du domaine d'application de la biologie synthétique. Nous retrouvons ainsi les trois grandes catégories que sont l'environnement, l'agroalimentaire et la santé.

Une évaluation complète des risques de biosûreté concernant les biosystèmes à applications environnementales est nécessaire, surtout avec l'arrivée proche sur le marché des premiers systèmes permettant de produire des énergies renouvelables. Le risque majeur concerne la contamination ou le croisement avec d'autres organismes ainsi que la modification de la biodiversité par la prolifération incontrôlée due à la libération accidentelle des organismes nouvellement développés dans la nature. Contrairement aux substances chimiques synthétiques habituellement produites, les organismes biologiques synthétiques peuvent être plus difficiles à

contrôler. Cependant, l'un des avantages de la biologie synthétique est de pouvoir intégrer directement pendant la conception des biosystèmes des mesures visant à limiter au maximum ces risques. Parmi les techniques utilisées nous retrouvons les gènes «suicides» ainsi que les technologies « terminator » qui peuvent être insérés dans les organismes, les empêchant de se reproduire ou de survivre en dehors d'un laboratoire ou d'un milieu contrôlé.

Les risques et les solutions apportées concernant les applications agroalimentaires de la biologie synthétique sont très similaires à ceux mentionnés pour l'application environnementale. De même que des algues synthétiques servant à la fabrication de carburant vert pourraient être relâchées dans la nature ou contaminer un milieu, l'impact de nouvelles plantes ou de nouvelles bactéries est difficile à contrôler. Cependant, les expériences montrent que la vie des organismes synthétiques ayant été développés en laboratoire a tendance à être de courte durée par rapport à ceux qui ont évolué dans la nature. Ces résultats sont encourageants vis-à-vis de la biosûreté, mais n'éliminent pas la nécessité de prendre des précautions.

Les applications biomédicales de la biologie synthétique augmentent forcément les risques potentiels pour l'homme et l'environnement car leur milieu de vie n'est plus confiné à un laboratoire. Les nouveaux organismes développés pour traiter des maladies peuvent déclencher des effets négatifs imprévus chez les patients comme des infections ou des réponses immunitaires inattendues voire des risques pour la santé totalement nouveaux, car l'évolution de ces organismes n'est pas prédictible. Pour limiter au maximum ces risques, les systèmes de précaution précédemment cités peuvent aussi être mis en place pour empêcher ou fortement limiter la croissance et la réplication des organismes synthétiques.

Le rapport de la PCSBI recommande également de limiter les activités de recherche en biologie synthétique à des laboratoires de niveau de biosûreté L3 ou L4 tant que l'évaluation des risques n'infirme pas la nécessité de cette protection [41]. Cette mesure concerne la biosûreté mais aussi la biosécurité.

Cette préoccupation intervient dans un contexte où émerge un phénomène récent, la biologie de garage ou « Do It Yourself Biology ». Celle-ci consiste à réaliser des recherches ou des expériences dans un cadre personnel hors laboratoire professionnel [42]. La biologie synthétique n'échappe pas à cette nouvelle communauté de scientifiques et le développement d'appareils de moins en moins coûteux, ainsi que la mise à disposition de séquences d'ADN « open source » comme les BioBriques, facilite son développement. Cela entraine principalement des questions de biosûreté, les groupes de volontaires n'étant généralement pas des spécialistes et le matériel employé ainsi que les recherches n'étant pas contrôlés comme c'est le cas dans des laboratoires ou des entreprises. Cependant ces personnes sont le plus souvent encadrées par des

professionnels et communiquent fréquemment sur leurs avancées. Ils disposent également de moyens leur permettant de se renseigner sur la biosûreté auprès d'experts.

1.4.2.2 Biosécurité

L'un des risques de biosécurité les plus couramment exprimés à l'encontre de la biologie synthétique est qu'entre de mauvaises mains, elle peut être employée pour créer des organismes nuisibles utilisables en tant qu'armes biologiques pour le bioterrorisme. Les exemples récents de reconstruction de virus en utilisant des techniques d'ADN recombinant alimentent ces préoccupations. La biologie synthétique en elle-même constitue donc un risque pour la biosécurité.

Malgré la relative facilité d'accès à des séquences d'ADN connues à travers les bases de données génétiques publiques, la plupart des experts de la communauté scientifique s'accorde à dire que la simple connaissance d'un génome viral est loin d'être suffisante pour être en mesure de reconstituer ou de créer une maladie pathogène. En effet, il faut pouvoir disposer d'un hôte et des conditions appropriées pour qu'un virus puisse se multiplier. Même si cela est scientifiquement réalisable, il est très peu probable qu'un groupe malveillant dispose à la fois des moyens financiers et des compétences techniques nécessaires à la réalisation de telles armes biologiques. Le risque concernant la biosécurité est donc très faible. D'autre part, la biosécurité peut être améliorée en utilisant les mêmes techniques présentées précédemment qui permettraient de garantir la biosûreté.

La biologie synthétique permet aussi d'améliorer la biosécurité en permettant aux chercheurs de marquer le code génétique des nouveaux organismes qu'ils développent, comme cela a été le cas pour la bactérie synthétique développée par l'Institut J. Craig Venter [21]. Lorsque ce processus de marquage est combiné avec les autres mesures présentées pour assurer la biosécurité, il peut fournir un moyen efficace de dissuasion afin de prévenir une utilisation malveillante.

1.4.3 Les brevets sur le vivant

La propriété intellectuelle sur le vivant et les brevets associés forment un vaste sujet qui connait de nombreux débats à travers le monde et dépend grandement des lois et de l'opinion publique de chaque pays. Une différence importante existe entre les Etats-Unis et l'Europe par exemple. Dans les deux cas, les êtres vivants en tant que tels et tous les produits de la nature associés ne peuvent pas être brevetés. Ce sont uniquement les procédés biotechnologiques, les produits obtenus par ces procédés ou leur utilisation qui peuvent être brevetables. Cependant, les Etats-Unis sont plus laxistes au niveau du champ d'application des brevets alors qu'en Europe, la

directive 98/44/CE des Parlement et Conseil Européens du 6 juillet 1998 relative à la protection juridique des inventions biotechnologiques confère aux pays membres une meilleure protection juridique.

Le domaine des végétaux est le cas historique de la protection intellectuelle sur le vivant. Aux Etats-Unis, la protection de certaines espèces végétales est garantie par le dépôt de brevets dès 1930, alors qu'en Europe, il a fallu attendre 1961 pour que l'Union pour la protection des obtentions végétales (UPOV) soit créée et garantisse une protection sur la sélection des semences par les Certificats d'Obtention Végétale (COV). Ceux-ci sont similaires aux brevets en termes de protection, mais concernent des variétés végétales sélectionnées qui n'entraient pas dans le domaine d'application des brevets.

Les biotechnologies ont bouleversé les connaissances et, de ce fait, ont permis de multiplier les applications potentielles. Le développement des brevets associés à ces recherches a ainsi été grandement accéléré. C'est le cas pour le génie génétique et les brevets déposés sur une séquence d'ADN. La directive 98/44/CE spécifie qu'en Europe, le génome d'organismes naturels ainsi que les séquences d'ADN ne peuvent pas être brevetés, car il s'agit de découvertes et non d'inventions. Aux Etats-Unis, le dépôt de ce genre de brevets était possible jusqu'à très récemment. Un des cas les plus médiatiques concerne la société Myriad Genetic Inc qui a déposé en 1998 plusieurs brevets sur des gènes susceptibles de provoquer des cancers du sein et de l'ovaire. Celle-ci a été la première à séquencer les gènes BRCA-1 et BRCA-2 (brevets US5747282 [43] et US5837492 [44]), sur lesquels certaines mutations augmentent le risque de cancer. Elle s'appuyait sur le fait que les séquences brevetées, d'après elle, n'existent pas de manière isolée à l'état naturel, et ne peuvent donc pas être considérées comme des produits de la nature. Ses détracteurs ne contestaient pas les brevets à proprement parler, mais reprochaient principalement à cette société de bloquer la recherche associée à leurs brevets par la non-délivrance de licences permettant l'exploitation par des sociétés tierces ou des laboratoires publics. L'affaire a finalement été portée devant la Cour suprême des Etats-Unis, qui a récemment rendu son verdict et invalidé ces brevets [45]. En effet, des équipes de chercheurs ont récemment démontré que des fragments d'ADN sont présents naturellement lors de certaines étapes de division cellulaire. La Cour suprême des Etats-Unis a ainsi considéré que les gènes brevetés sont des produits de la nature et n'entrent donc pas dans le domaine d'application des brevets.

Les biosystèmes développés dans le cadre de la biologie synthétique ne devraient quant à eux pas poser de problème à l'égard des lois actuelles car le but de cette science est bien la création de nouveaux organismes par l'ingénierie. Les produits générés entrent donc sous le couvert de la protection intellectuelle par des brevets. Cependant, seul le système entier peut être breveté et

non les parties élémentaires de ce système, puisque ceux-ci sont considérés comme des produits de la nature.

1.5 Conclusion

Ce chapitre introductif a pour but de présenter la biologie synthétique et les problématiques associées. Comme il s'agit d'une technoscience à l'interface entre deux communautés scientifiques bien distinctes, le premier challenge est de comprendre, pour chacune des deux communautés, les attentes, les habitudes, les problématiques et les technologies de l'autre. Néanmoins, comme beaucoup de scientifiques du domaine s'accordent à le dire, la biologie synthétique est promise à un bel essor qui pourrait être catalysé par la réussite de cette collaboration.

Il existe cependant un risque, communément admis comme étant un des principaux freins à ce développement, qui concerne le manque de structure, de réutilisabilité et d'interopérabilité des travaux des différentes équipes de recherche et de développement. Cette question a été posée et résolue il y a plusieurs années par la microélectronique par différentes méthodes. L'objet de cet travail de thèse est d'étudier jusqu'à quel point ces méthodes, mises au point pour l'électronique, pourraient s'appliquer à la biologie synthétique.

Chapitre 2

Flot de conception et outils associés existant

Nous avons pu voir dans le chapitre précédent que la complexité des biosystèmes est en pleine augmentation. Si son évolution, décrite par la loi de Carlson, se confirme, des outils d'aide à la conception vont devenir indispensables. Dans ce chapitre, nous allons présenter la conception d'un système de manière générale. Celle-ci est basée principalement sur deux approches : le « bottom-up » et le « top-down ». Cette dernière a conduit au développement du prototypage virtuel, qui consiste à effectuer un maximum d'étapes par simulation, plutôt que de réaliser des prototypes physiques.

La microélectronique est un domaine où l'expérience en conception de systèmes est très importante et qui a bénéficié de dizaines d'années d'évolution pour se perfectionner. Nous présentons ici le flot de conception utilisé pour la conception de systèmes numériques. Nous abordons aussi un élément clé de celui-ci, le design kit. Enfin, les modèles orientés conception sont passés en revue.

La biologie synthétique est un domaine beaucoup plus récent que la microélectronique, mais beaucoup de logiciels ont déjà été développés pour cette science. Nous avons donc réalisé un état de l'art des outils les plus aboutis répondant aux problématiques de conception, de modélisation et de simulation et d'assemblage des éléments biologiques. Les travaux de plusieurs équipes, montrant la volonté de tendre vers un outil d'aide à la conception biologique (Genetic Design Automation ou GDA en anglais), seront ensuite présentés.

Durant cette thèse, nous avons fait le choix de privilégier une démarche de réutilisation des outils de conception de la microélectronique pour arriver à l'élaboration d'un GDA, au lieu d'en développer de nouveaux, spécifiquement programmés pour du matériel biologique. Cela a nécessité l'adaptation du flot de conception et des outils associés ainsi que le développement de nouveaux modèles compréhensibles par ces outils.

2.1 Les approches de conception d'un système

Les différentes approches de conception présentées dans cette section sont utilisées dans l'élaboration de nombreux systèmes électriques, mécaniques, thermiques mais aussi informatiques. Cette conception est réalisée sous deux approches principales : l'approche

« bottom-up », aussi appelée approche ascendante, et l'approche « top-down » aussi appelée approche descendante. Si ces approches portent le même nom que les approches actuelles de création d'un biosystème présentées dans le chapitre 1, elles divergent cependant dans leur sens.

2.1.1 L'approche « Bottom-up »

L'approche « bottom-up » est l'approche historique utilisée dans la conception de systèmes. Elle consiste à partir de composants standards provenant d'une librairie qui vont être assemblés pour former des sous-systèmes indépendants, correspondant aux diverses fonctions du système total. Une fois validés, ceux-ci sont également assemblés pour former le système complet. Le système est finalement comparé aux spécifications exigées. Nous pouvons comparer cette approche à la construction d'une maison, ou à l'élaboration d'une structure en Lego®. Cette approche est illustrée Figure 2.1.

Le principal avantage de cette approche est de bénéficier d'une plus grande visibilité sur l'évolution de la conception, grâce au développement progressif de chaque sous-système. Ceci peut améliorer la rentabilité du projet à court terme, dans le cas de la mise en service anticipée des sous-systèmes ou de leur réutilisation pour plusieurs projets. En revanche, elle ne permet pas d'avoir une vision globale du système. Le système final peut alors être différent des spécifications initiales, entraînant ainsi un retour en arrière pour corriger les sous-systèmes déjà réalisés, ce qui s'avère coûteux.

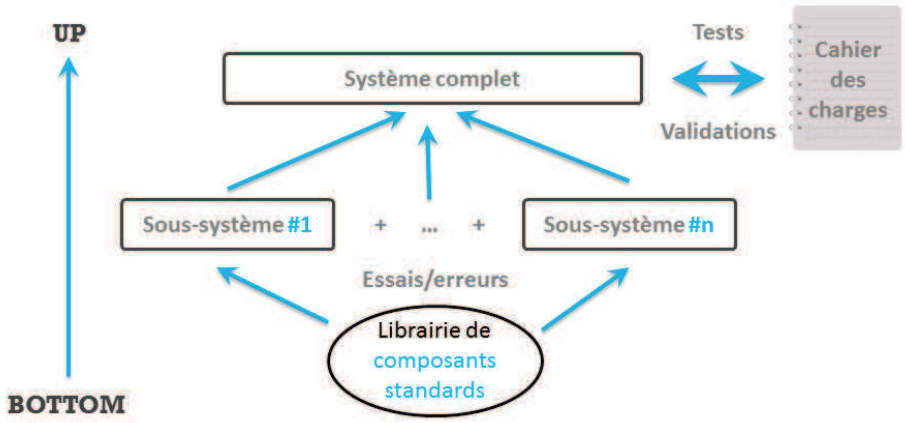


Figure 2.1 : Approche de conception « bottom-up ».

2.1.2 L'approche « Top-down »

La définition de l'approche « Top-down » présentée ici est différente de celle utilisée dans le chapitre 1 pour la création d'un organisme synthétique. L'utilisation de cette approche d'un point

de vue biologique se réfère à la simplification d'un organisme dans le but d'obtenir le biosystème synthétique fonctionnel le plus simple possible. Son utilisation dans le domaine de l'ingénierie correspond à la vision inverse. Chaque étape de la conception sous cette approche va fournir des informations permettant d'enrichir les connaissances dans la conception d'un système. Cette approche consiste, dans un premier temps, à se baser sur les spécifications globales d'un système et à les transformer en sous-systèmes possédant chacun leur propre cahier des charges. On procède ensuite à l'assemblage du système à partir de ces sous-systèmes, en piochant dans une librairie de composants standards. Cette étape peut être automatisée. Le système ainsi obtenu peut être considéré juste par conception. Cette approche est similaire à celle d'un artiste qui commencerait par réaliser des esquisses de son œuvre dans son entier, avant de peindre chaque élément en détail. Elle est illustrée Figure 2.2.

Cette approche « top-down » présente principalement l'intérêt d'être plus rapide, et à long terme moins coûteuse, que l'approche « bottom-up ». La définition en début de projet des cahiers des charges des différents sous-systèmes permet aussi une bonne estimation préalable des coûts de conception et de fabrication du système. Seul inconvénient, l'automatisation du processus nécessite des modèles prédictifs afin de donner des résultats intéressants. Cependant, ces modèles ne sont pas toujours simples à réaliser, notamment lorsque l'on ne dispose que de descriptions sommaires du système.

Généralement la conception d'un système fait intervenir un mélange de ces deux approches en fonction des éléments du système.

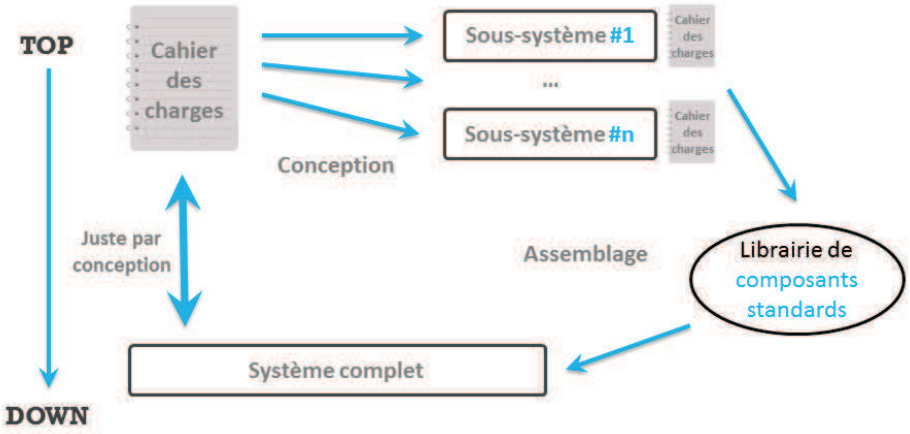


Figure 2.2 : Approche de conception « Top-down ».

2.1.3 Le prototypage virtuel

Le développement du prototypage virtuel qui se produisit à la fin du siècle dernier constitue l'une des principales révolutions dans la conception de systèmes [46]. Il consiste à réaliser une

modélisation du système, appelée prototype virtuel, permettant de tester le système en cours de conception dans un environnement virtuel de tests et dans des conditions s'approchant au plus près de la réalité. Nous évitons ainsi d'avoir à créer un prototype réel, bien plus coûteux.

Cette méthodologie permet d'accélérer le processus de conception. En effet, la réactivité est plus grande, car des modifications peuvent être apportées au cours de la conception en fonction des résultats obtenus. L'efficacité et la fiabilité d'un tel prototypage virtuel sont directement liées à l'exactitude des modèles et à la capacité de prévoir correctement le comportement du système. Pour cela, un important effort a été réalisé au cours de la dernière décennie afin de développer des modèles précis et rapides ainsi que des langages et des outils associés pour réaliser les différentes étapes de ce processus.

2.2 Les outils en microélectronique

La microélectronique a vu évoluer grandement sa méthodologie de conception des systèmes au cours des 60 dernières années. Dans la catégorie des microprocesseurs, le nombre de transistors intégrés est passé de 2300 dans le 4004 d'Intel, sorti en 1971, à plus de 2,3 milliards dans le Xeon Nehalem-EX d'Intel en 2010. Cette évolution a été rendue possible grâce au perfectionnement des technologies, mais aussi grâce à un flot de conception performant.

2.2.1 Présentation du flot de conception

Les outils de CAO (Conception Assistée par Ordinateur) sont particulièrement efficaces pour la conception des parties numériques d'un système. Pour ces systèmes, l'approche « top-down » peut être automatisée à l'aide d'outils, comme « la suite Cadence ». Le flot de conception pour les systèmes numériques réside en cinq étapes principales, illustrées Figure 2.3 et détaillées dans cette section.

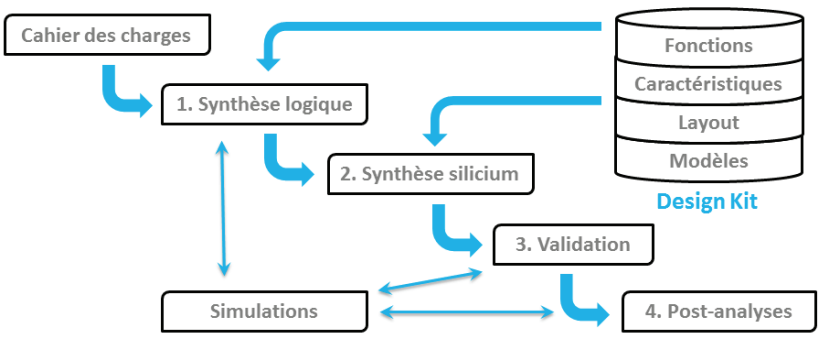


Figure 2.3 : Le flot de conception des systèmes numériques en microélectronique.

2.2.1.1 Description comportementale

La description du comportement d'un système est le point de départ de la conception. Cette description, qui est très abstraite et très loin de la réalisation finale, peut être directement effectuée à l'aide de langages de description matérielle (Hardware Description Language : HDL), comme le VHDL, le Verilog, ou le SystemC. Ces langages sont compréhensibles par les concepteurs, mais peuvent aussi être directement interprétés et simulés par des outils d'EDA (Electronic Design Automation).

2.2.1.2 Synthèse logique

L'étape de synthèse logique est composée de trois sous-étapes :

- La première est la transformation de la description comportementale effectuée à l'étape précédente sous la forme d'une description RTL (Register-Transfer Level). Il s'agit d'une description des flux de données en utilisant des registres et des fonctions logiques.
- La deuxième sous-étape est la description à un niveau « porte logique » qui correspond à une vue schématique du système, impliquant des portes logiques et des registres. Elle est obtenue automatiquement à partir de la description RTL par les logiciels d'EDA.
- La troisième sous-étape est constituée par l'assemblage de cellules standards. Comme les registres et les portes logiques sont des composants classiques connus, la description précédente peut être directement traduite pour obtenir le système fonctionnel. Cependant, bien que cette construction soit juste, ce n'est pas forcément la plus optimisée. Dans les outils d'EDA, cette étape est également réalisée automatiquement. Elle correspond au choix du meilleur chemin électrique entre les composants permettant de réduire les retards et la consommation énergétique tout en conservant une fréquence maximale de fonctionnement.

2.2.1.3 Synthèse silicium

Cette étape consiste, une fois l'assemblage de cellules standards obtenu, à remplacer ces cellules par leurs équivalents transistors pour réaliser la fonction donnée. Des étapes d'optimisation interviennent aussi afin d'optimiser le positionnement des cellules standards sur le silicium pour réduire la surface et la facilité de connexion entre les cellules.

2.2.1.4 Validations et post-analyses

Une extraction des paramètres est réalisée afin de valider le schéma silicium en le comparant aux résultats des modèles. Après quelques post-traitements, le schéma est prêt à être envoyé à une fonderie de silicium.

Dans le schéma Figure 2.3, nous constatons que les étapes de synthèse logique et de synthèse silicium ont besoin de récupérer des informations contenues dans ce qui est appelé « le design kit ».

2.2.2 Le design kit

Le design kit est la clé de voûte du flot de conception. Il contient toutes les informations liées aux technologies utilisées pendant la conception. Nous y retrouvons ainsi une librairie d'éléments standards correspondant à des fonctions données (des composants combinatoires comme des portes logiques, des composants séquentiels comme des bascules, etc.) avec leurs caractéristiques associées, les modèles adaptés à la conception de ces différents composants, leurs « layouts », correspondant aux masques appliqués sur le silicium, et enfin des règles de design nécessaires pour garantir le bon fonctionnement des éléments de la librairie. Ces différentes parties du design kit sont schématisées Figure 2.4.

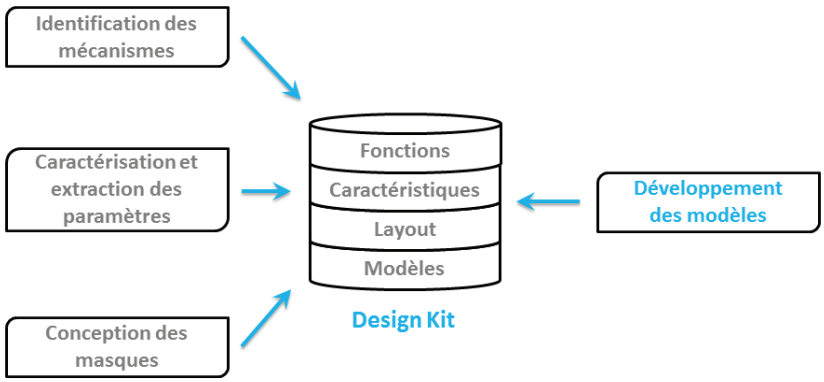


Figure 2.4 : Le design kit, élément clé du flot de conception.

Ce design kit est particulièrement important dans le flot de conception, et la réussite du développement d'un système dépend grandement de la justesse de son contenu. Celui-ci est alimenté à plusieurs niveaux à l'issue de conceptions de systèmes antérieurs. L'identification de différents mécanismes permettent d'ajouter des fonctions à la librairie d'éléments standards. Leur caractérisation permet l'extraction des paramètres associés à ces fonctions, nécessaire pour renseigner leurs caractéristiques. A partir de ces paramètres, l'élaboration de modèles spécifiques à la conception peut ensuite être réalisée.

2.2.3 Les modèles orientés conception

Les modèles orientés conception sont destinés à être utilisés durant les étapes du flot de conception afin de les valider. Leur développement est effectué de trois façons différentes, réalisées en parallèle.

A partir d'une étude théorique du système à modéliser, la première approche consiste à élaborer des modèles compacts. Ce sont des modèles relativement simples qui se basent sur des équations approximées de la physique et permettent d'avoir des résultats de simulation rapides et efficaces. Ils dépendent généralement de peu de paramètres et peuvent avoir plusieurs abstractions (numérique, analogique, etc.).

La deuxième approche consiste à élaborer des modèles beaucoup plus complexes, reposant directement sur les lois physiques et chimiques régissant le comportement des matériaux utilisés. Ils peuvent se baser sur des méthodes gourmandes en temps de calcul, comme la méthode des éléments finis et nécessiter des simulateurs dédiés. Plusieurs outils permettent de décrire ces modèles, comme les environnements « Comsol » [47] pour la simulation des systèmes physiques ou « Silvaco » [48] pour les dispositifs microélectroniques.

La dernière approche repose sur l'élaboration de prototypes concrets permettant de disposer de résultats du comportement réel de certaines parties du système voire de sa totalité.

Ces trois approches sont illustrées Figure 2.5.

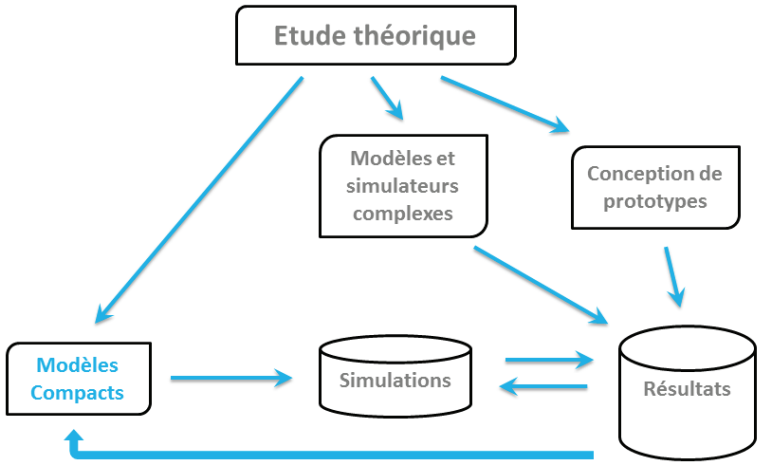


Figure 2.5 : Modélisation orientée conception.

En règle générale, une combinaison de ces trois approches est souvent nécessaire pour établir les modèles les plus aboutis. En effet, les modèles complexes peuvent permettre de mettre en évidence une dépendance d'un certain paramètre physique qui serait passé au travers de

l'analyse effectuée pour les modèles compacts. Après validation de cette dépendance grâce aux résultats obtenus sur les prototypes, elle pourra être intégrée aux modèles compacts.

2.3 Les outils pour la biologie

Si nous souhaitons transposer ces techniques à la biologie, il faut d'abord étudier les outils existants. Actuellement, plus de 200 outils ont été développés dans le cadre de l'aide à la conception pour la biologie synthétique. Cependant, ils ont été conçus pour des applications spécifiques et/ou dans un contexte donné et ne correspondent seulement qu'à certaines étapes de conception. Dans cette section, nous allons présenter un état de l'art des différents logiciels pertinents, puis nous analyserons les efforts effectués dans la conception de biosystèmes dans le but de se rapprocher d'un outil GDA.

2.3.1 Etat de l'art

Le nombre important de logiciels dans ce domaine nous a contraints à ne sélectionner dans cet état de l'art que les plus aboutis à l'heure actuelle. Une étude réalisée par D. Chandran *et al.* traite un plus grand nombre de ces outils [49]. Ils peuvent être classés en fonction de leur utilité. Quatre thèmes principaux se dégagent :

- la conception des biosystèmes,
- la sélection et l'assemblage des différents éléments biologiques,
- l'analyse, la modélisation et la simulation des biosystèmes,
- leur optimisation.

2.3.1.1 La conception de systèmes biologiques

La première étape dans le développement d'un nouveau biosystème concerne les outils de conception de systèmes biologiques, qui doivent répondre aux besoins des ingénieurs. Parmi les plus avancés, nous retrouvons BioJADE, GenoCAD, SysBioSS et Tinkercell.

BioJADE est une application permettant la conception graphique d'un biosystème [50]. Elle a la particularité d'utiliser des symboles de fonctions électroniques pour les différents blocs du réseau de composants biologiques. BioJADE permet de procéder à des ajustements et à des optimisations du système ainsi que de simuler son comportement à l'aide de plusieurs simulateurs. Cet outil permet aussi de faire le lien entre ces blocs et les bases de données contenant les informations et les séquences ADN correspondantes, comme le registre des BioBriques [51].

GenoCAD est une application CAO pour la biologie synthétique reposant sur une interface web [52]. Le principal intérêt de GenoCAD est qu'il considère l'ADN comme un langage et peut identifier certaines séquences comme des mots, à partir de comparaisons avec une base de données. Grâce à une grammaire spécifiquement élaborée, basée sur des règles de conception, GenoCAD vérifie et guide le concepteur dans l'agencement des différentes séquences d'ADN. Cet outil permet aussi de réaliser des simulations des systèmes conçus, grâce à l'interfaçage avec le cœur de simulation COPASI, présenté par la suite. L'interface web de GenoCAD est présentée Figure 2.6, qui illustre les trois parties de l'outil : les éléments biologiques, la conception du système et la simulation.

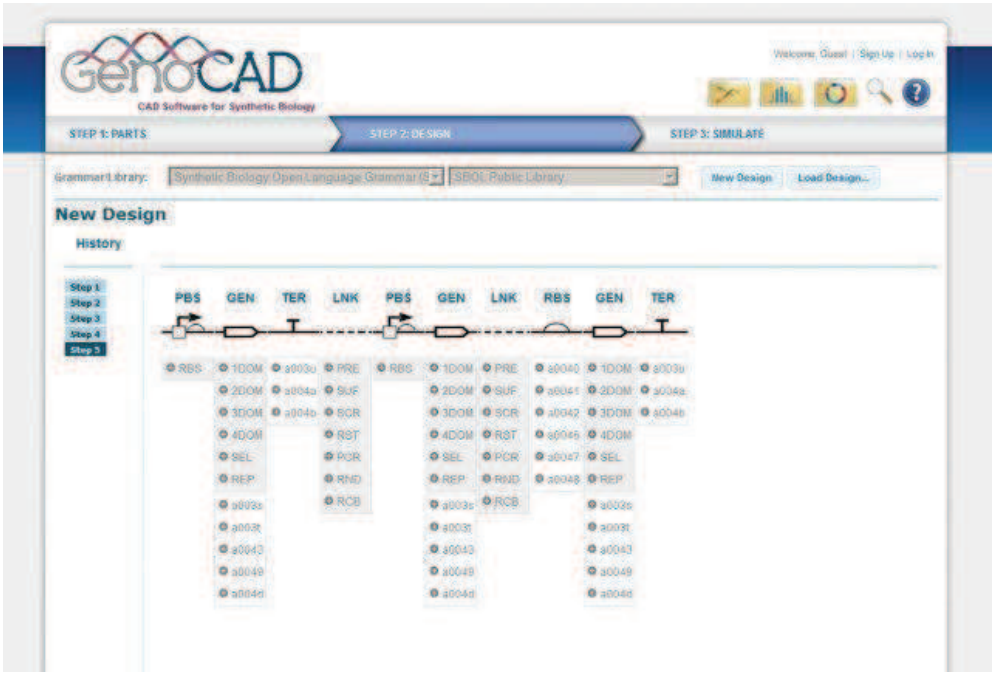


Figure 2.6 : Interface web de GenoCAD.

SynBioSS est un outil composé de deux interfaces. La première, une application web, intègre un éditeur permettant d'assembler différentes BioBriques, comme des promoteurs avec des régions codantes. La deuxième, un logiciel plus complet, ajoute des outils de simulation stochastique [53].

Tinkercell est quant à lui un outil CAO assez complet, destiné spécifiquement à la biologie synthétique [54]. Il consiste en une interface graphique permettant la saisie de diagrammes représentant les interactions entre les espèces du système à concevoir. A partir de ces diagrammes, il peut ensuite extraire automatiquement les modèles mathématiques associés et réaliser plusieurs types de simulations. Une des forces de cette application est le support de modules additionnels qui permettent d'ajouter assez facilement des fonctionnalités spécifiques. L'interface de TinkerCell est présentée Figure 2.7.

Nous retrouvons également les travaux de l'équipe de D. Densmore et R. Weiss dans l'élaboration d'une plateforme permettant de générer automatiquement un biosystème à partir de sa description haut niveau ou comportementale [55], [56]. Ce flot de conception nommé TASBE [57], transforme automatiquement cette description en un réseau génétique composé d'éléments standardisés, qui sont ensuite assemblés pour pouvoir être implémentés chez un organisme hôte.

Figure 2.7 : Interface de TinkerCell.

2.3.1.2 Sélection et assemblage des éléments biologiques

Parmi les outils présentés précédemment, la plupart permet de se connecter aux bases de données contenant des éléments biologiques standardisés, comme les BioBriques, afin de les sélectionner puis de les assembler. Cette étape est très importante et nécessite des algorithmes permettant de lier les fonctions désirées à des séquences d'ADN particulières.

Clotho est un outil qui fournit lui aussi au concepteur une connexion aux bases de données contenant les éléments biologiques [58]. Il est couplé avec de nombreux modules permettant l'analyse des fonctions réalisées, la recherche de BioBriques correspondant à ces fonctions, ainsi que l'échange de données au sein de la communauté. L'interface de Clotho est présentée Figure 2.8, qui illustre la multitude de modules disponibles.

L'équipe de D. Densmore propose également un algorithme de recherche et de sélection de BioBriques, nommé MatchMaker [59]. Il permet de faire correspondre des séquences d'ADN à des descriptions abstraites de biosystèmes.

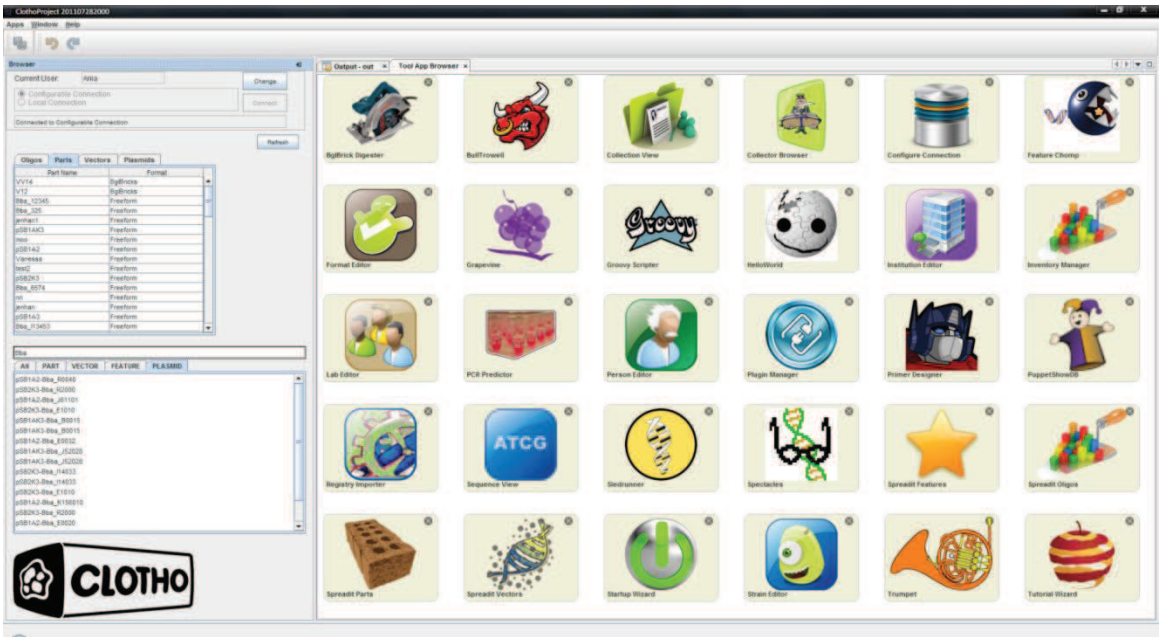


Figure 2.8 : Interface de Clotho.

2.3.1.3 Analyses et simulation

Afin de connaître le comportement d'un biosystème, il est nécessaire de disposer d'outils fournissant des résultats quantitatifs à partir de modèles. Ceux-ci peuvent être un ensemble d'équations régissant le fonctionnement du système, ou bien des représentations stochastiques de ces équations. Parmi les outils servant à élaborer ces modèles et à réaliser les simulations nous retrouvons JDesigner et CellDesigner, ainsi que COPASI.

JDesigner est un environnement de modélisation graphique pour des réseaux de réactions biochimiques [60]. Il permet de saisir le système et de sélectionner différentes lois cinétiques pour la modélisation des réactions. JDesigner se base ensuite sur le Systems Biology Workbench (SBW) pour réaliser les simulations et les analyses. Le SBW est un ensemble d'éditeurs de modèles, de simulateurs et d'outils d'analyses dédiés à la biologie. JDesigner accepte aussi la description d'un système au format standard SBML, présenté dans la sous-section Modélisation. L'interface de cet outil est illustrée Figure 2.9 sur un exemple d'oscillateur.

CellDesigner est un éditeur de schéma structuré de description des réseaux de régulation génétique et biochimique [61]. Cette modélisation est réalisée sur la base du format standard SBML, ce qui lui permet d'interagir avec d'autres outils comme le SBW mais aussi l'outil COPASI

présenté ci-dessous. CellDesigner réalise ainsi des simulations statiques et dynamiques et permet aussi l'analyse de paramètres en intégrant des outils de résolution des équations différentielles utilisées pour la modélisation. Cet outil est très répandu dans la communauté des biotechnologues et s'adapte très bien aux spécificités de la biologie synthétique.

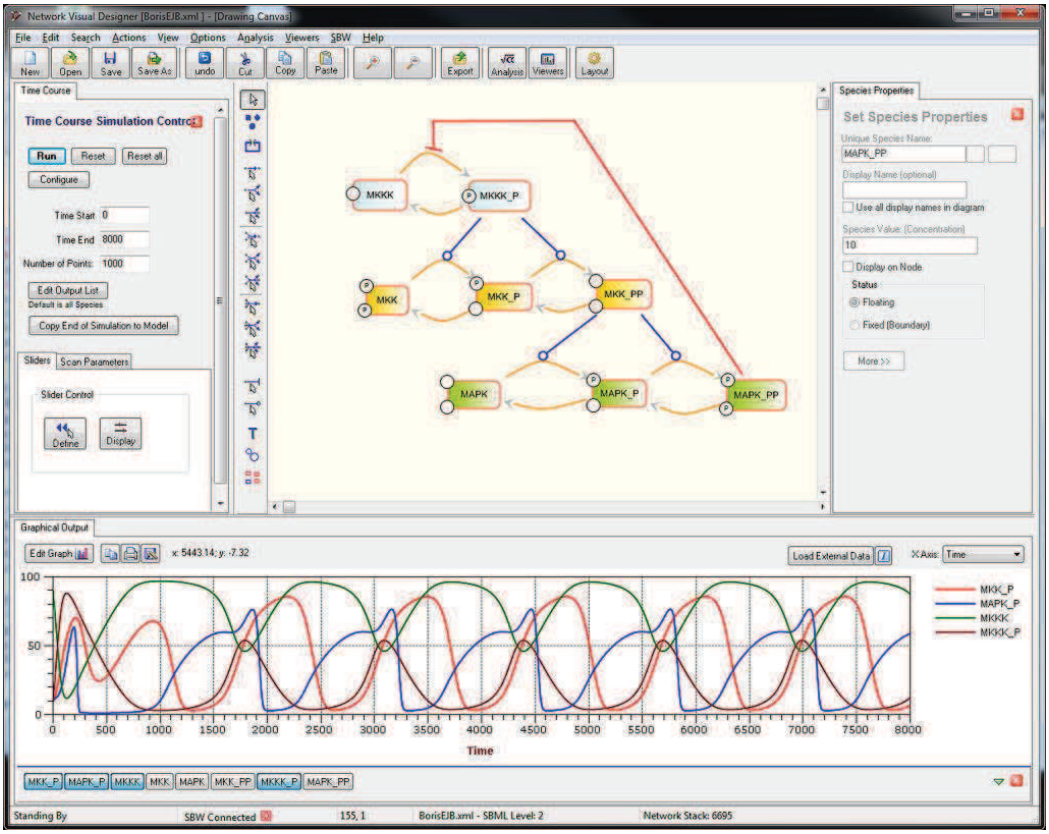


Figure 2.9 : Interface de JDesigner.

COPASI est une application destinée à la simulation et à l'analyse de réseaux de réactions biochimiques et à leur évolution dynamique [62]. COPASI est un programme autonome qui prend en charge les modèles décrits selon la norme SBML et permet la simulation de leur comportement en utilisant des équations différentielles ou un algorithme de simulation stochastique. Des événements discrets arbitraires peuvent être inclus dans les simulations afin d'étudier le comportement du système avec l'ajout instantané d'une espèce. Cet outil propose plusieurs sortes d'analyses et permet l'estimation des paramètres des modèles ainsi que leur optimisation.

2.3.1.4 Optimisation

Les derniers outils correspondent à l'optimisation et à l'épuration de l'assemblage des codes ADN des éléments biologiques. Certaines séquences codant pour des sites de restriction ainsi que les régions redondantes qui mènent à des résultats indésirables comme la recombinaison peuvent

être supprimées. Pour effectuer l'édition des séquences ADN, plusieurs outils sont disponibles. Parmi eux, nous retrouvons GeneDesign, GeneDesigner et Sequence Refiner.

GeneDesign est un ensemble d'applications web qui fournit divers outils de manipulation nucléotidique pour la biologie synthétique [63]. Ceux-ci permettent des opérations d'édition des séquences comme l'optimisation des codons et la suppression des sites de restriction mais aussi l'annotation de certaines parties des séquences.

GeneDesigner est un logiciel possédant une interface graphique intuitive permettant la visualisation et la modification des séquences ADN [64]. De nombreux algorithmes d'optimisation sont également intégrés.

Enfin, Sequence Refiner, un module de l'outil BioFAB basé sur le logiciel Proto [65], rempli le même rôle que les deux outils précédents, mais il permet en plus de vérifier la conformité des séquences d'ADN avec les standards définis pour la biologie synthétique.

2.3.2 Développement de librairies standards

Le développement de librairies d'éléments biologiques standards, libres d'accès, a connu une forte croissance avec la création des BioBriques par Tom Knight, Drew Endy et Christopher Voigt en 2003 [16]. La création de BioBriques est couplée à une base de données, le Registry of Standard Biological Parts [51]. Elle regroupe la totalité des BioBriques conçues, avec un système de notation, de test et de validation par les utilisateurs. Ces librairies facilitent l'échange et la mise en commun des avancées réalisées sur certains éléments biologiques, par le biais de l'utilisation d'un même standard.

L'avantage de ces BioBriques réside aussi dans leur classification, qui reprend celle que l'on utilise dans des domaines où la conception de systèmes est maitrisée, comme c'est le cas en microélectronique. En terme de complexité, nous pouvons ainsi faire l'analogie entre les BioBriques « parts » et les transistors, entre les BioBriques « devices » et les portes logiques et enfin entre les BioBriques « systems » et des systèmes complets comme les processeurs. Cela a pour conséquence de faciliter l'adaptation des méthodologies de conception employées dans des domaines différents à celui de la biologie.

B. Canton et D. Endy proposent aussi une description standardisée des BioBriques, présentant leurs caractéristiques de la même manière que dans les datasheets des composants électroniques [66]. L'adoption de telles datasheets de composants biologiques par la communauté des biotechnologues, couplée à l'utilisation des méthodes de caractérisation poussées des BioBriques, simplifierait leur utilisation et leur modélisation.

2.3.3 Modélisation

La représentation des systèmes biologiques est un point crucial qui nécessite d'une part une compréhension aisée du langage ou de la méthodologie de représentation par les concepteurs, et d'autre part une analyse rapide et pertinente par les logiciels de simulation. Deux standards principaux coexistent actuellement pour accomplir cette tâche. Il s'agit du SBML (Systems Biology Markup Language) [67], pour la représentation de systèmes biologiques en général, et le SBOL (Synthetic Biology Open Language) [68], plus spécifiquement adapté à la biologie synthétique.

Au début des années 2000, H. Kitano et J. C. Doyle cherchaient à redéfinir une nouvelle infrastructure de représentation des systèmes biologiques, adaptée à la modélisation informatique. Leur travail permit d'aboutir à la première version du SBML en 2001. Après plus de 10 ans de développement, son utilisation s'est étendue à de nombreux logiciels et le langage continue à être maintenu régulièrement avec l'ajout de nouvelles fonctionnalités.

Le SBML est donc un format compréhensible par les outils de simulation permettant de modéliser des systèmes biologiques. Ces systèmes sont décrits en définissant les différentes espèces biologiques qui sont impliquées, et les processus qui lient ces espèces entre elles, comme les réactions biochimiques. La dernière version introduit la modularité dans la description des structures ce qui le rapproche d'un langage de description matérielle comme le VHDL. Le SBML est basé sur le langage XML (Extensible Markup Language) qui permet de structurer une description à l'aide de balises dont le vocabulaire et la grammaire sont personnalisables. Le XML se prête particulièrement bien à la description de systèmes biologiques de par son aspect structurel. Un exemple de fichier de description SBML est présenté Figure 2.10, pour le cas de la réaction de complexation du chapitre 3.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sbml level="2" version="3" xmlns="http://www.sbml.org/sbml/level2/version3">
    <model name="Complexation">
        <listOfSpecies>
            <species compartment="cytosol" id="AB" initialAmount="0"     name="AB"/>
            <species compartment="cytosol" id="B"  initialAmount="1e-20" name="B"/>
            <species compartment="cytosol" id="A"  initialAmount="5e-21" name="A"/>
        </listOfSpecies>
        <listOfReactions>
            <reaction id="veq">
                <listOfReactants>
                    <speciesReference species="A"/>
                    <speciesReference species="B"/>
                </listOfReactants>
                <listOfProducts>
                    <speciesReference species="AB"/>
                </listOfProducts>
                <kineticLaw>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <apply>
                            <times/>
                            <apply>
                                <minus/>
                                <apply>
                                    <times/>
                                    <ci>kon</ci>
                                    <ci>A</ci>
                                    <ci>B</ci>
                                </apply>
                                <apply>
                                    <times/>
                                    <ci>koff</ci>
                                    <ci>AB</ci>
                                </apply>
                            </apply>
                        </apply>
                    </math>
                    <listOfParameters>
                        <parameter id="kon"  value="1000000"/>
                        <parameter id="koff" value="0.2"/>
                    </listOfParameters>
                </kineticLaw>
            </reaction>
        </listOfReactions>
    </model>
</sbml>
```

Figure 2.10 : Standard de représentation SBML, code la réaction de complexation présentée au chapitre 3.

En 2008, un autre standard, le SBOL, fait son apparition. Il a pour vocation de réaliser un équivalent des langages de description matérielle utilisés en électronique, comme le Verilog ou le VHDL, mais adapté à la description de systèmes développés dans le cadre de la biologie synthétique. Il permet de combler le manque de standards en biologie synthétique pour la description des systèmes et permet la modélisation des structures biologiques, allant des séquences élémentaires d'ADN aux composants entiers, ce qui correspond aux différents niveaux des BioBriques. Le SBOL est lui aussi basé sur le format de fichier XML, ce qui est illustré Figure 2.11, avec l'instanciation de plusieurs BioBriques. Un code graphique, faisant le lien avec les spécifications du langage SBOL, a aussi été développé. Appelé SBOL Visual, il permet la conception graphique des assemblages de BioBriques.

Figure 2.11 : Standard de représentation SBOL, avec plusieurs BioBriques.

2.3.4 Volonté d'aller vers un GDA

Les différents outils présentés précédemment permettent de répondre aux besoins des concepteurs pour plusieurs étapes d'un flot de conception dédié à la biologie synthétique. La première approche pour réaliser un outil complet d'aide à la conception pour la biologie synthétique, ou GDA, consiste à regrouper ces outils en une suite couvrant toutes les étapes du flot de conception. La classification des outils présentés illustre bien la possibilité d'envisager la construction d'une telle suite, dont les différentes étapes sont documentées par D. Chandran *et al.* [49].

Une tendance montrant la volonté de développer un outil GDA semble aussi se détacher de la littérature. Nous retrouvons ainsi les travaux de l'équipe de C. J. Myers [69] et de l'équipe de D. Densmore [70], qui vont dans ce sens. Ceux-ci soulignent la complexité du développement d'une suite d'aide à la conception pour la biologie synthétique, et la nécessité de l'élaboration d'algorithmes efficaces.

2.3.5 Faisabilité

Dans une revue de 2011, W. Lux *et al.* s'interrogent sur la faisabilité et les résultats d'un outil GDA [71]. En effet, la pertinence d'un tel outil est remise en cause par trois principales limitations actuelles :

- Le caractère prédictif des composants biologiques qui fait défaut.
- Le découplage entre la conception et la fabrication de ces constructions biologiques qui n'est pas dans les habitudes des biotechnologues.
- Les méthodes de caractérisation des éléments biologiques qui ne sont pas très abouties.

Ces trois éléments peuvent laisser à penser que les GDA vont avoir certaines difficultés à s'imposer dans la conception de biosystèmes. Leur application dans la conception de systèmes biologiques pourrait se limiter, dans un premier temps du moins, à accélérer les progrès réalisés par les expérimentateurs en leur permettant de disposer d'un système de prototypage virtuel. Celui-ci servirait à tester certaines hypothèses avant des investigations plus poussées sur l'élaboration réelle du biosystème.

Cependant, au vu de la complexité grandissante des biosystèmes présents dans la littérature et des efforts réalisés dans la standardisation des éléments biologiques de base comme les BioBriques, une démocratisation des outils de GDA semble très probable dans les années à venir. Le travail présenté dans cette thèse est de proposer une approche innovante qui consiste à repartir des outils de la microélectronique.

2.4 Adaptation des outils de la microélectronique pour la biologie

L'approche de conception de la microélectronique a fait ses preuves au courant des dernières décennies. En effet, les outils sont performants et tirent parti de l'expérience du domaine de la microélectronique pour s'ajuster à des systèmes de plus en plus multi-domaines. En revanche, la biologie synthétique est un domaine beaucoup plus récent et les nombreux outils développés n'ont pas encore pu s'illustrer et montrer leur efficacité dans la conception de systèmes biologiques. Ceux-ci ont d'ailleurs été développés dans un cadre spécifique ou avec un formalisme propre à certaines équipes, ce qui réduit leur adaptabilité à des biosystèmes génériques. Au final, bien que certains logiciels possèdent plusieurs éléments répondant à des besoins d'aide à la conception, aucun ne couvre un flot de conception complet.

Notre idée a donc été de nous baser sur le flot de conception des systèmes numériques, et, grâce à des analogies entre mécanismes biologiques et fonctions logiques, d'adapter les outils de ce flot de conception à du matériel biologique.

2.4.1 Le flot de conception

La méthodologie générale employée dans le flot de conception a été conservée mais plusieurs étapes ont dû être adaptées. Les modifications nécessaires sont illustrées Figure 2.12.

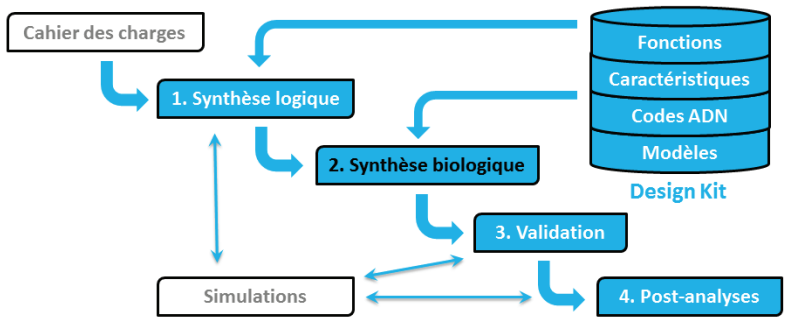


Figure 2.12 : Adaptations à réaliser sur le flot de conception de la microélectronique pour une application biologique.

En comparant ce schéma à la Figure 2.3, le principal changement concerne l'étape de la synthèse silicium, qui se transforme en une synthèse biologique. Les autres étapes qui devront être modifiées sont celles de la synthèse logique (car les outils doivent prendre en compte d'autres contraintes) ainsi que la validation et les analyses effectuées sur les systèmes qui ne seront plus exécutées de la même manière. L'adaptation de ce flot de conception est présentée au chapitre 4.

2.4.2 Le design kit

Le design kit doit quant à lui être complètement reconstruit. Il ne comportera plus les « layout » des éléments sur silicium, mais les codes ADN correspondant aux différents mécanismes utilisés. Le développement de nouvelles fonctions pour alimenter la librairie d'éléments standards, ainsi que leur caractérisation, est un travail qui doit être effectué par des biotechnologues.

Le développement de modèles efficaces et pertinents étant la clé d'une approche de conception « top-down » réussie, le travail effectué pendant cette thèse s'est concentré sur la modélisation des différents mécanismes biologiques et des fonctions associées. Ce travail est présenté dans les chapitres 4 à 7.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les deux principales approches utilisées dans la conception de systèmes : l'approche « bottom-up » et l'approche « top-down ». Ces approches

sont particulièrement efficaces dans la conception de systèmes en microélectronique, l'approche « top-down » pouvant même être automatisée dans le cas de circuits numériques.

Dans le cas de la conception de biosystèmes, celle-ci peut être guidée grâce à des suites de logiciels d'aide à la conception appelées GDA. Pour aboutir à de tels outils, deux solutions se présentent : la première est d'assembler différents outils spécifiquement développés pour le domaine de la biologie, correspondant aux différentes parties d'un flot de conception pour la biologie synthétique. Cette méthode pose plusieurs problèmes, notamment celui de la compatibilité entre les différents outils, généralement développés pour une application spécifique.

La deuxième approche consiste à adapter le flot de conception utilisé en électronique numérique à du matériel biologique. C'est cette solution qui a été retenue par notre équipe. Le travail effectué sur les différents outils composant le flot de conception microélectronique, ainsi que le développement de modèles orientés conception sont présentés dans la deuxième partie de cette thèse.

Chapitre 3

Mécanismes biologiques de base

Après avoir défini les objectifs (au chapitre 1) et les outils (au chapitre 2), nous allons maintenant étudier la manière de réaliser les fonctions biologiques. Nous avons démontré au cours des précédents chapitres qu'il existe des similitudes structurelles et méthodologiques entre la biologie et l'électronique. Il reste maintenant à trouver un équivalent à l'électron comme vecteur de l'information, et au transistor comme brique élémentaire servant au traitement de cette information.

En ce qui concerne le vecteur d'information, plusieurs niveaux d'abstractions peuvent être imaginés, comme nous l'illustrons Figure 3.1. En partant de la plus petite taille à la plus grande taille, nous avons : les molécules dont les interactions définissent les fonctions biologiques d'une cellule, puis les cellules dont les interactions définissent les organes, ensuite les organes dont les interactions définissent les organismes ou individus et enfin les individus dont les interactions définissent la société.

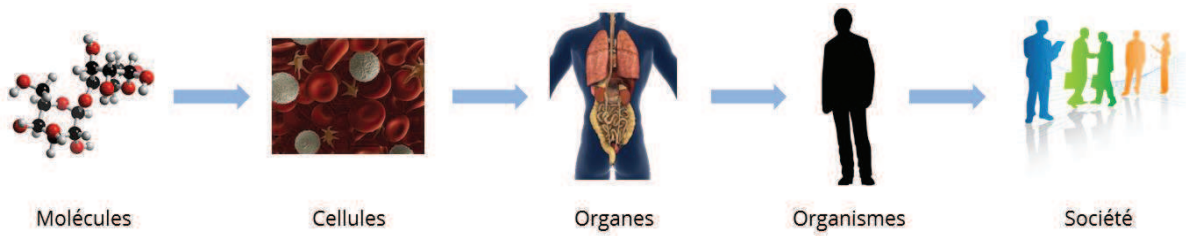


Figure 3.1 : Niveaux d'abstraction de l'information.

Pour des raisons techniques (mais aussi éthiques et sociétales), il est plus facile de manipuler des organismes unicellulaires et de travailler au niveau moléculaire. Le vecteur d'information biologique est donc constitué de molécules, ce qui implique quelques différences fondamentales avec les électrons en électronique, comme nous le verrons par la suite.

Quant au transistor biologique, plusieurs mécanismes sont possibles. Le transistor électronique est un composant grâce auquel il est possible de contrôler le passage du courant en fonction des tensions appliquées. Il s'agit en ce sens d'un composant dit actif. Le transistor biologique devra donc également être un mécanisme permettant de modifier des quantités de molécules en fonction des quantités d'autres molécules. Pour remplir ce rôle, trois mécanismes seront principalement étudiés :

- le mécanisme d'association entre une protéine et d'autres molécules, qui est à la base de tous les mécanismes biologiques,
- le mécanisme de synthèse des protéines, dont l'activité dépend de la présence de molécules régulatrices,
- les mécanismes d'endocytose et d'exocytose permettant respectivement l'entrée et la sortie de protéines à travers la membrane d'une cellule, et qui peuvent également être contrôlés par d'autres molécules.

D'autres mécanismes, comme les interactions entre ARN et diverses molécules ne sont pas présentés dans ce chapitre. Cependant, ils présentent un intérêt dans la conception de certains systèmes biologiques (riborégulateurs). Un exemple de leur utilisation est présenté au chapitre 6 avec le système de détection de cellules cancéreuses faisant intervenir des microARNs qui vont se lier à l'ARN. En perspective, nous allons également voir quelles espèces biologiques pourraient être utilisées dans de futurs biosystèmes synthétiques.

3.1 Liaison entre une protéine et une autre molécule

Pour agir sur la quantité d'une molécule, l'une des solutions les plus simples est de la faire intervenir dans une réaction chimique contrôlée par un autre réactif. C'est le cas pour les protéines qui peuvent interagir avec d'autres molécules et ainsi se lier avec celles-ci afin de former un complexe moléculaire. Ces liaisons sont classifiées en trois grandes catégories en fonction du type d'espèce avec laquelle la protéine peut former un complexe : la liaison protéine-protéine [72], la liaison protéine-ADN [73] et la liaison protéine-ligand [74].

3.1.1 Mécanisme

Le mécanisme de la liaison entre deux espèces biochimiques est très simple à comprendre. Il peut être vu de manière abstraite comme un système clé/serrure. La structure moléculaire de la protéine forme une serrure (appelé site de liaison) sur lequel un ligand dont la forme est complémentaire peut se lier, donnant ainsi des propriétés nouvelles au complexe formé. Ce mécanisme ainsi que le code graphique utilisé dans les différents schémas présentés dans ce manuscrit est décrit en Annexe A.

Nous considérons ici la liaison entre une protéine P et un ligand L, mais le comportement est le même pour une liaison protéine-protéine ou protéine-ADN. Cette liaison peut aussi être représentée par l'équation biochimique à l'équilibre suivante :

$$P + L \overset{K}{\Leftrightarrow} PL \tag{3.1}$$

où K est la constante d'association entre les différents composants du complexe. La liaison présentée ici peut appartenir à deux grandes sortes de liaisons, les covalentes, ou les non-covalentes. Les liaisons covalentes sont beaucoup plus fortes que les liaisons non covalentes car, dans les premières, il y a partage d'une paire d'électrons entre les deux composants du complexe. Les liaisons non-covalentes sont quant à elles réparties en plusieurs catégories :

- les liaisons hydrogènes : elles mettent en jeu le partage d'un électron entre un atome donneur, l'hydrogène et d'un atome accepteur (souvent l'oxygène ou l'azote) ;
- les liaisons ioniques : elles se forment entre deux atomes ayant une grande différence d'électronégativité, par exemple l'attraction entre un anion et un cation. Elles sont grandement dépendantes de la distance entre les atomes ;
- Les liaisons de van der Waals : ces liaisons sont constituées de la somme des forces d'attraction et de répulsion entre les molécules autres que les liaisons covalentes, hydrogènes et ioniques. Elles sont très faibles mais comme elles sont nombreuses, la somme devient importante ;
- Les interactions hydrophobiques : elles apparaissent au niveau des molécules possédant une majorité de liaisons non-polaires, ce qui entraine leur incapacité à réaliser des liaisons hydrogènes. L'interaction avec les molécules d'eau étant rendue impossible, ces molécules ont tendance à former des liaisons avec des composés organiques.

Ces différentes liaisons peuvent être soit réversibles, c'est-à-dire que les deux composants du complexe moléculaire peuvent se délier, soit être irréversibles. La liaison d'une molécule sur la protéine peut aussi modifier la conformation de la protéine, et ainsi faciliter ou empêcher la liaison d'une molécule supplémentaire. Les liaisons seront décrites plus en détails au chapitre 8.

3.1.2 Fonction

Pour identifier la fonction utile de ce mécanisme, en termes de conception de biosystèmes, nous analysons les différents cas de figure de la présence et de l'absence de la protéine et du ligand, et le résultat sur la constitution du complexe. Les résultats obtenus sont résumés Table 3.1.

Protéine (P)	Ligand (L)	Complexe (PL)
Absente	Absent	Absent
Présente	Absent	Absent
Absente	Présent	Absent
Présente	Présent	Présent

Table 3.1 : Cas de figure de la présence et de l'absence d'une protéine et de son ligand.

Nous pouvons faire l'analogie entre le comportement de ce mécanisme et une porte logique, la porte ET, dont la table de vérité est donnée Table 3.2.

P	L	PL
0	0	0
1	0	0
0	1	0
1	1	1

Table 3.2 : Table de vérité de la porte logique ET.

Cette analogie nous permet de considérer le mécanisme de complexation moléculaire comme une porte ET. L'équation numérique de cette porte biologique est fournie équation (3.2) et son schéma Figure 3.2.

$$Complexe = Protéine.Ligand \tag{3.2}$$

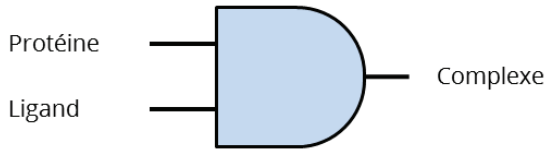


Figure 3.2 : Porte logique biologique basée sur le mécanisme de liaison moléculaire.

3.2 Synthèse des protéines

Le mécanisme de synthèse des protéines est à la base du fonctionnement de la plupart des biosystèmes synthétiques actuels. Il se déroule en deux étapes principales : la transcription de l'ADN en ARN messager et la traduction de l'ARN messager en une ou plusieurs protéines [75], [76]. Ces deux molécules sont des polymères nucléotidiques très semblables mais ils possèdent tout de même des différences significatives. L'ADN est constitué de deux brins de nucléotides formant une structure en double hélice alors que l'ARN n'est composé que d'un seul brin. L'ADN est constitué de quatre nucléotides différents, l'adénine (A), la thymine (T), la cytosine (C) et la guanine (G). L'ARN est également constitué des nucléotides A, G et C, mais la thymine est remplacée par l'uracile (U).

3.2.1 Transcription

La première partie du mécanisme se nomme la transcription. Nous pouvons distinguer deux sortes d'organismes, les eucaryotes, qui possèdent un noyau dans lequel est présent l'ADN, et les

procaryotes, dont l'ADN est présent dans le cytoplasme. La transcription a lieu là où se situe l'ADN, donc le noyau cellulaire pour les eucaryotes et le cytoplasme pour les procaryotes. Nous pouvons comparer ce mécanisme avec l'usinage d'une pièce mécanique. L'ADN contient le plan de construction de la pièce (la protéine). Ce plan étant unique, il est précieux et conservé en un seul endroit. Pour que la pièce soit usinée, il faut dans un premier temps faire une copie de ces plans. Cette étape est réalisée par l'ARN polymérase, une enzyme qui va parcourir une partie du brin d'ADN pour en réaliser une copie conforme.

L'ARN polymérase commence cette transcription au niveau d'un endroit particulier de l'ADN : une zone un peu en aval du promoteur. Il s'agit d'une séquence définie de nucléotides permettant à l'ARN polymérase de reconnaitre le début d'un fragment d'ADN à copier. L'enzyme de transcription va tout d'abord dérouler la double hélice d'ADN. Les deux brins sont ensuite séparés l'un de l'autre et l'ARN polymérase identifie le brin d'ADN à partir duquel elle va réaliser la copie. Il s'agit du brin non codant, et l'enzyme va ajouter, à la suite du brin d'ARN, le complémentaire de chaque nouveau nucléotide lu. Les complémentaires sont définis de la manière suivante : A avec T (respectivement A avec U pour l'ARN) et C avec G. Le deuxième brin d'ADN, appelé brin codant, étant le complémentaire du brin non codant, il correspond à la séquence d'ARN transcrit (à l'exception des nucléotides T remplacés par les U). Lorsque l'ARN transcrit est complet, l'ARN polymérase se sépare de l'ADN, ses deux brins se recollent et l'ADN reprend sa structure en double hélice. Cette étape est résumée Figure 3.3.

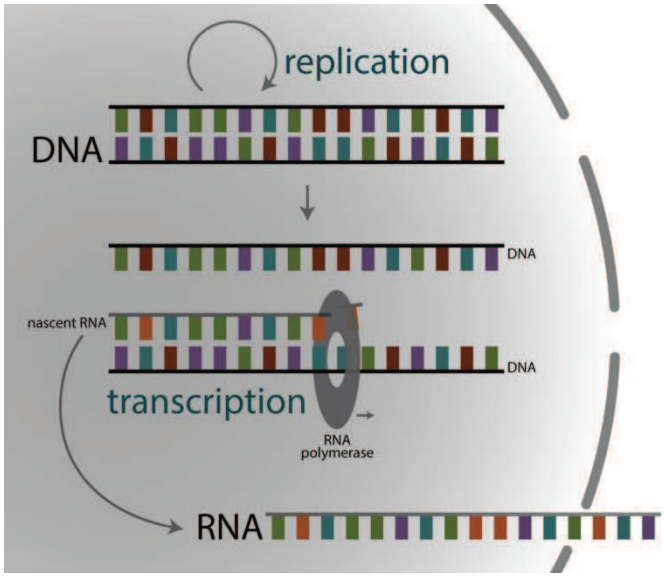


Figure 3.3 : Mécanisme de transcription de l'ADN en ARN (Source MITx 7.00x).

Chez les eucaryotes, une fois que l'ARN est transcrit, il passe ensuite par une étape de traitement. Le brin est complété par une coiffe en début de brin et par une queue en fin de brin. Ces ajouts lui permettent entre autres d'être correctement reconnu lors de l'étape suivante de traduction

mais aussi d'éviter sa dégradation. L'ARN transcrit contient en réalité des régions codantes, appelées exons, et des régions non codantes, appelées introns. L'étape qui suit est appelée épissage. Elle consiste en un ensemble de coupures et de ligatures du brin d'ARN permettant de ne garder que les exons dans le brin d'ARN messager final. Cette étape de suppression des introns est illustrée Figure 3.4.

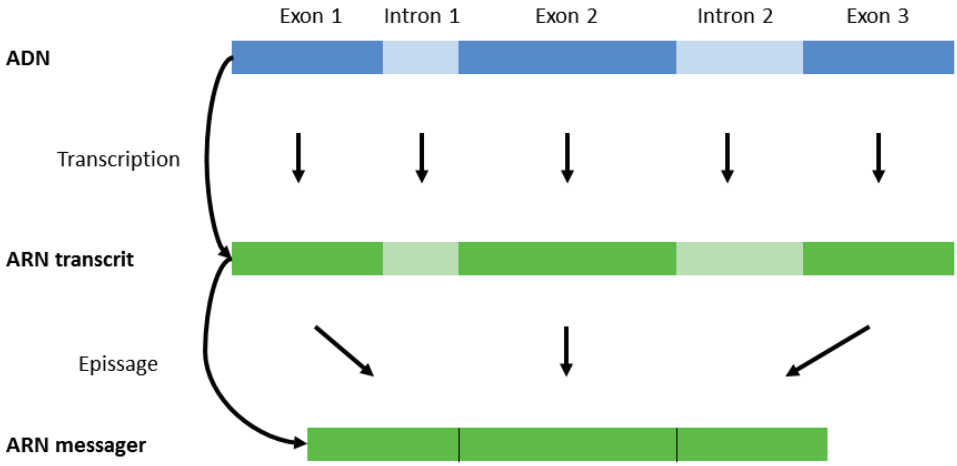


Figure 3.4 : Epissage du brin d'ARN transcrit pour supprimer les introns.

Le brin d'ARN messager (ARNm) finalisé est ainsi constitué de groupes de trois nucléotides, appelés codons. Après une zone correspondant au site de fixation du ribosome, ceux-ci vont chacun coder pour un acide aminé particulier, ce que nous allons voir dans la deuxième étape de la synthèse de protéines, la traduction.

3.2.2 Traduction

Cette deuxième partie du mécanisme de synthèse des protéines, appelée traduction, consiste à assembler la protéine à partir du code de l'ARNm. De manière très simpliste, l'ARNm contient des groupes de trois nucléotides, appelés codons, chacun codant pour un acide aminé. Ce mécanisme se déroule au niveau d'un ribosome et conduit à la synthèse d'une protéine, une séquence d'acides aminés donnée par la séquence des codons.

Dans les détails, cette étape se déroule dans le cytoplasme, après que le brin d'ARNm soit sorti du noyau de la cellule chez les eucaryotes. Lors de cette étape, nous avons besoin du brin d'ARNm à traduire mais aussi d'autres éléments comme les ribosomes et des ARN de transfert (ARNt) ainsi que de l'énergie fournie par de l'ATP. Les ribosomes sont des complexes moléculaires constitués de protéines et d'ARN ribosomiques (ARNr). Ils sont composés de deux unités principales : la première, plus petite, qui va lire la séquence codée sur l'ARNm et la seconde, plus importante, chargée de faire la liaison peptidique entre les différents acides

aminés constituant la protéine. Ces acides aminés sont apportés par les ARNt. Pour reprendre l'analogie avec l'usinage d'une pièce mécanique, les ribosomes correspondent aux machines dans lesquelles les pièces sont produites et les ARNt aux ouvriers qui amènent la matière première. Cette deuxième étape de la synthèse de protéine est illustrée Figure 3.5.

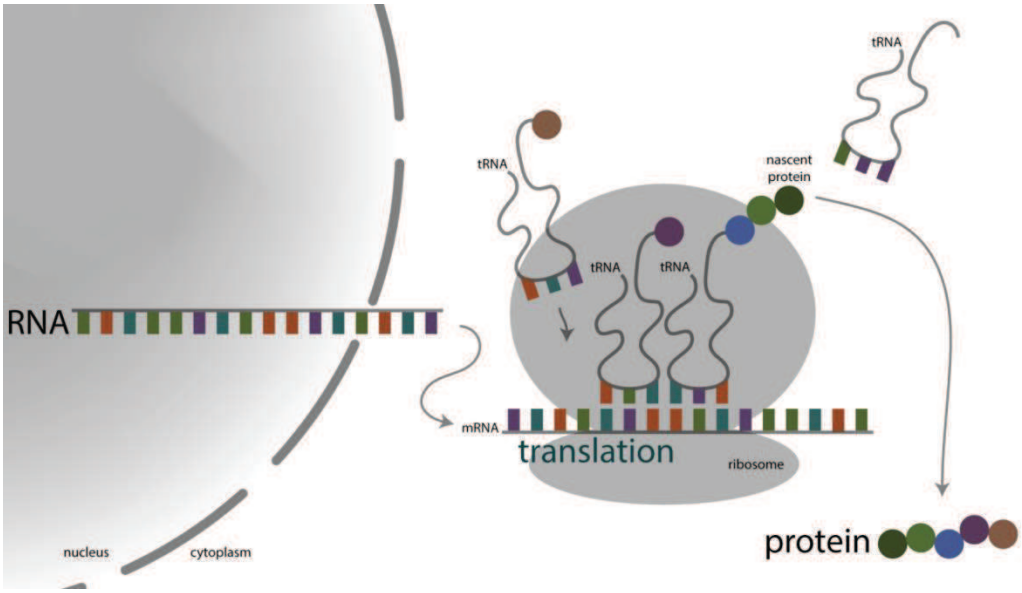


Figure 3.5 : Traduction de l'ARNm en protéine (Source MITx 7.00x).

Les ARNt sont aussi constitués d'un anticodon. L'anticodon est une séquence de nucléotides complémentaire à un codon donné. L'ARNt va ainsi fixer un acide aminé correspondant à la séquence de son anticodon dans le cytoplasme et l'apporter au niveau du ribosome. Le ribosome va ainsi parcourir le brin d'ARNm, et les ARNt, correspondant aux codons en cours de lecture, vont venir se lier sur le brin d'ARNm au fur et à mesure. Pendant ce temps, le ribosome réalise la liaison peptidique entre les acides aminés apportés par les ARNt. Quand la liaison est effectuée, l'ARNt se détache de l'ARNm et retourne dans le cytoplasme. Ce mécanisme se poursuit, allongeant la chaine polypeptide des acides aminés, jusqu'à tomber sur un codon STOP. A ce stade, la traduction s'arrête et la chaîne polypeptidique entière se détache et se replie pour former une protéine complète et fonctionnelle. Le ribosome ainsi que le brin d'ARNm vont pouvoir ensuite être réutilisés dans d'autres traductions. Comme un seul brin d'ARNm peut servir à produire plusieurs fois la même protéine, nous parlons de mécanisme d'amplification.

Chaque codon de l'ARNm code ainsi pour un acide aminé mais plusieurs codons différents peuvent coder pour le même acide aminé. Une correspondance entre les codons et les acides aminés est présentée Figure 3.6. La traduction commence au niveau du site de liaison du ribosome. Dans la plupart des cas cette zone correspond au codon AUG qui code pour la méthionine. La fin de la traduction d'un brin d'ARNm est définie par un codon stop, qui est codé par plusieurs codons différents, ce qui est illustré Figure 3.6.

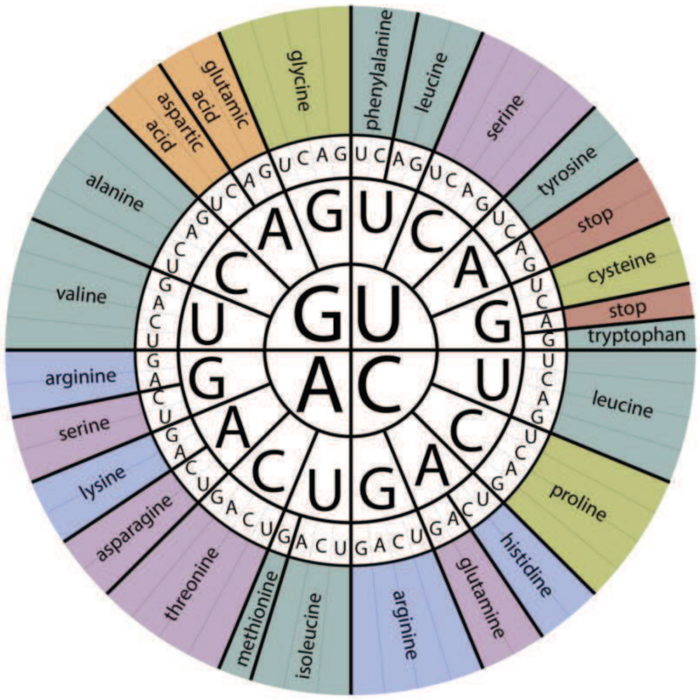


Figure 3.6 : Table de correspondance entre les codons et les acides aminés (Source MITx 7.00x).

3.2.3 Fonction réalisée

Le mécanisme de synthèse des protéines est un mécanisme qui régule le fonctionnement élémentaire des êtres vivants. Pour l'utilisation de ce mécanisme dans la conception de biosystèmes, il est nécessaire d'en donner une description plus abstraite. L'abstraction consiste à considérer les mécanismes de transcription et de traduction comme des boîtes noires et d'exprimer les quantités des protéines synthétisées en fonction d'entrées, et/ou de paramètres environnementaux. Les entrées de ce système seraient alors d'autres espèces chimiques capables de moduler les deux boites noires.

En pratique, l'essentiel du contrôle se situe au niveau du promoteur, où des protéines peuvent se lier et agir comme des facteurs de modulation de la transcription. Ces protéines peuvent soit accélérer la transcription, dans ce cas nous parlons d'activateurs, soit la bloquer, et dans ce cas nous parlons de répresseurs. Le promoteur peut aussi ne pas être régulé par les facteurs de transcription, mais permettre une transcription continue du gène. Nous parlons dans ce cas de promoteurs constitutifs.

En considérant le promoteur comme une partie de contrôle d'un gène, et le restant du code ADN comme la protéine qui sera produite, nous obtenons Figure 3.7, le schéma simplifié d'un gène comportant ces deux parties.

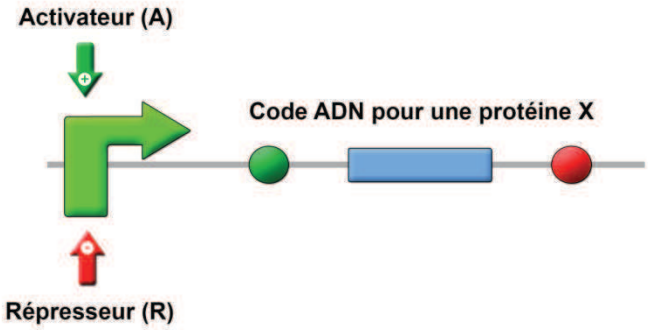


Figure 3.7 : Représentation simplifiée d'un gène, comprenant la partie de contrôle (le promoteur), et le code ADN correspondant à la protéine synthétisée. Le rond vert symbolise la zone de début et le rond rouge la fin de la transcription (appelé aussi terminateur de transcription).

A partir de cette représentation, plusieurs modèles peuvent être imaginés afin de compléter les boîtes noires. Ils sont décrits de manière exhaustive dans le chapitre 6. Ici, nous présentons uniquement la vision la plus abstraite afin de bien comprendre la fonction réalisée par ce mécanisme. Nous considérons les différents cas de figure de la présence et de l'absence de l'activateur (A) et du répresseur (R) et nous analysons le résultat sur la synthèse de la protéine X. Nous obtenons les différentes situations résumées Table 3.3.

Activateur (A)	Répresseur (R)	Gène	Protéine (X)
Absent	Absent	Non exprimé	Non synthétisée
Présent	Absent	Exprimé	Synthétisée
Absent	Présent	Non exprimé	Non synthétisée
Présent	Présent	Non exprimé	Non synthétisée

Table 3.3: Impact de la présence et de l'absence de l'activateur et du répresseur lors de la synthèse d'une protéine.

A partir de la table 3.3, nous pouvons faire l'analogie entre le comportement de ce mécanisme et une porte logique, la porte INH, aussi appelée SI-NON (IF-NOT en anglais), dont la table de vérité est donnée Table 3.4.

A	R	X
0	0	0
1	0	1
0	1	0
1	1	0

Table 3.4: Table de vérité de la porte logique INH.

Le mécanisme se comporte donc comme l'équation numérique (3.3) et son schéma est illustré Figure 3.8.

$$Protéine = Activateur.\overline{Répresseur} \tag{3.3}$$

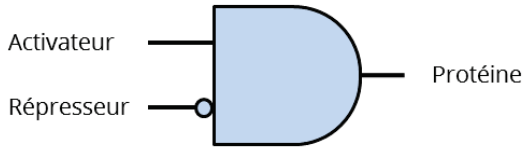


Figure 3.8 : Porte logique biologique basée sur le mécanisme de synthèse des protéines.

L'intérêt de ce mécanisme réside aussi dans sa fonction numérique équivalente. En effet la fonction SI-NON est un opérateur universel, ce qui signifie que le groupe complet d'opérateurs logiques, comme par exemples les fonctions ET, OU, NON-ET, NON-OU et OU exclusif, peut être obtenu à l'aide de combinaisons de cette seule fonction. Ainsi, à partir de combinaisons de plusieurs mécanismes de synthèse de protéines différentes, nous pouvons a priori créer n'importe quelle fonction logique. Cette analogie réalisée entre ce mécanisme biologique et les portes logiques est très importante car elle est à la base d'analogies entre la biologie et l'électronique, sur laquelle repose notre travail d'adaptation du flot de conception (voir chapitre 4).

La fonction logique ET peut par exemple être réalisée à l'aide de deux synthèses de protéines, ce qui est illustré Figure 3.9 et dont les équations logiques sont données en (3.4). Cependant, nous constatons que cette implémentation implique au moins deux gènes et nécessite au minimum quatre protéines pour produire la fonction recherchée.

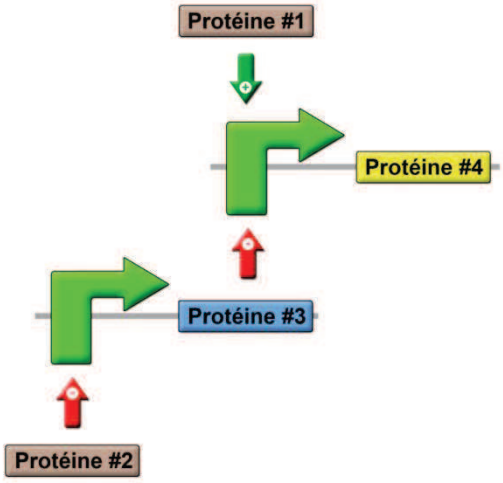


Figure 3.9 : Fonction logique ET réalisée à partir de deux mécanismes de synthèse.

$$Protéine\#3 = \overline{Protéine\#2}$$

$$Protéine\#4 = Protéine\#1.\overline{Protéine\#3} = Protéine\#1.\overline{\overline{Protéine\#2}} \\ = Protéine\#1.Protéine\#2 \tag{3.4}$$

Cette fonction semble donc plus simple à réaliser à l'aide de deux protéines synthétiques pouvant former un complexe moléculaire, ce qui est présenté dans la section 3.1.2. Le critère de choix entre ces deux mécanismes va résider dans les implications inhérentes à chacun de ces mécanismes. Dans le cas d'une synthèse, les entrées (l'activateur et le répresseur) sont considérées comme n'étant pas consommées par le mécanisme. En effet, ces molécules se lient lors de la transcription, mais une fois celle-ci terminée, elles sont libérées et peuvent à nouveau participer au mécanisme. La production de la sortie d'un tel système ne consomme donc pas de protéines, ce qui est plutôt recherché pour correspondre au mieux au comportement numérique. Le mécanisme de complexation moléculaire va par contre consommer les protéines des entrées puisqu'elles se lient pour former le complexe moléculaire de sortie. Dans certaines applications, cette consommation des entrées est très utile et nous privilégierons alors plutôt ce mécanisme.

3.3 Endocytose et exocytose

Les deux mécanismes présentés jusque-là interviennent à l'intérieur même de la cellule. Les deux mécanismes de cette partie, l'endocytose et l'exocytose, permettent des échanges de différentes espèces entre l'intérieur et l'extérieur de la cellule, à travers la membrane [77]. Ils peuvent être considérés en quelque sorte comme des ports d'entrée/sortie d'un biosystème.

3.3.1 Endocytose

Ce mécanisme est utilisé par la cellule pour faire entrer de larges molécules qui ne peuvent pas passer à travers la barrière hydrophobe que constitue la membrane de la cellule. Lors de ce processus, la membrane cellulaire se déforme localement et forme une vésicule cellulaire hermétique qui se retrouve ainsi dans le cytoplasme. Nous pouvons distinguer trois principaux types d'endocytoses :

- la phagocytose, qui correspond à la liaison puis à l'absorption par la cellule de grosses particules comme des déchets cellulaires ou des micro-organismes ;
- la pinocytose, qui est l'incorporation de petites quantités de liquide extracellulaire, pouvant contenir du matériel de faible taille dissous dans le liquide ;

- l'endocytose d'absorption, ou endocytose liée à des récepteurs, qui, en présence des molécules associées à des récepteurs présents à la surface membranaire, provoque la formation d'une vésicule qui va les intégrer et permettre leur entrée dans le cytoplasme.

Ces trois mécanismes d'endocytose sont illustrés Figure 3.10.

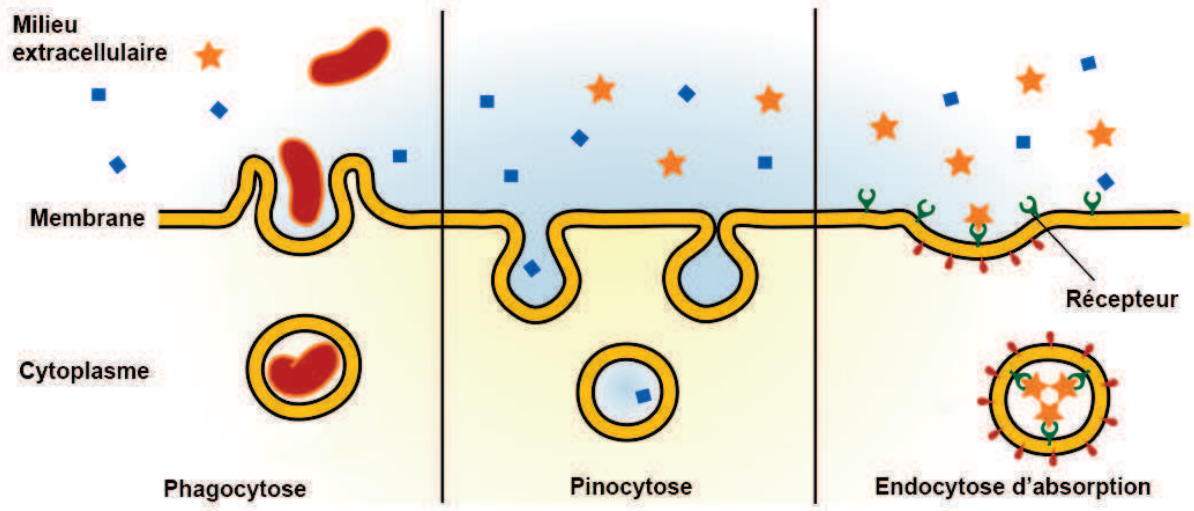


Figure 3.10 : Trois types d'endocytose : la phagocytose, la pinocytose et l'endocytose d'absorption (source : Wikimedia Commons).

3.3.2 Exocytose

L'exocytose correspond au mécanisme inverse de l'endocytose, c'est-à-dire à la sécrétion à l'extérieur de la cellule de molécules présentes dans le cytoplasme. L'appareil de Golgi produit des vésicules incorporant les molécules à faire sortir de la cellule, puis celles-ci migrent jusqu'à la membrane cellulaire et fusionnent avec elle, libérant ainsi leur contenu dans le milieu extracellulaire. Les molécules qui sortent de la cellule peuvent être des enzymes, des hormones, des protéines nécessaires à un autre endroit, des neurotransmetteurs ainsi que des déchets cellulaires. Il existe deux sortes d'exocytose, illustrées Figure 3.11 :

- l'exocytose constitutive, utilisée pour faire sortir naturellement des composants hors de la cellule ou pour renouveler la membrane ;
- l'exocytose régulée, déclenchée par des ions comme le calcium par exemple dans les synapses, utilisé dans la signalisation interneuronale pour la production de neuromédiateurs.

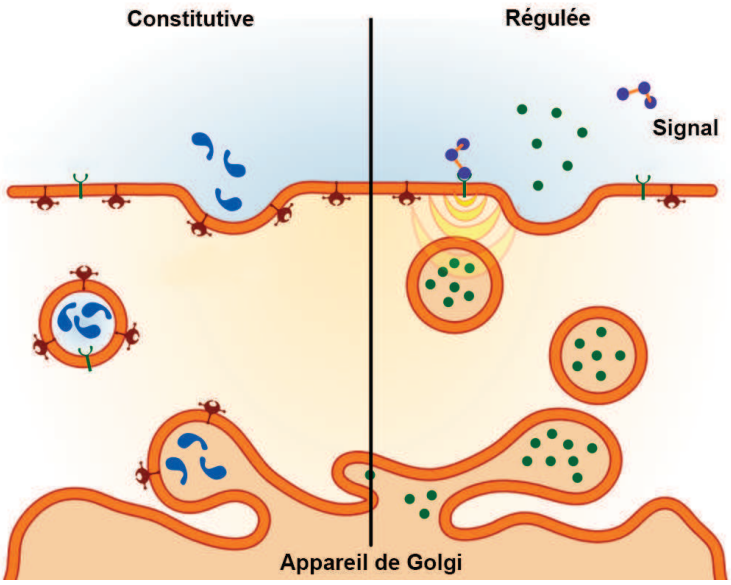


Figure 3.11 : Deux types d'exocytose : constitutive et régulée (source : Wikimedia Commons).

3.3.3 Fonctions utiles

Afin de pouvoir disposer d'un mécanisme de transport de l'information extracellulaire pour la conception de biosystèmes, seulement un type d'endocytose et d'exocytose peuvent être exprimés sous forme de fonctions logiques. Pour l'endocytose, il s'agit de l'endocytose d'absorption qui permet une sélection du type de molécule pouvant entrer dans la cellule grâce aux récepteurs spécifiques à certaines protéines extérieures, nécessaires à ce mécanisme. Pour l'exocytose, la régulation par des ions peut être utilisée pour faire sortir de la cellule une protéine spécifique. Seule l'exocytose régulée par le biais de la fixation de molécules spécifiques à un récepteur peut donc être utilisée afin d'obtenir un mécanisme sélectif.

Nous pouvons simplifier ces deux mécanismes et les résumer par les deux équations logiques en (3.5) qui nous permettent ainsi de disposer de fonctions d'entrée et de sortie contrôlées d'une protéine dans la cellule.

$$Protéines\_ext = Protéines\_int.Récepteurs$$
$$Protéines\_int = Protéines\_ext.Récepteurs \quad (3.5)$$

3.4 Autres porteurs d'information

Ce chapitre est axé sur les mécanismes associés aux protéines, qui sont majoritairement utilisées comme messagères dans les biosystèmes synthétiques actuels. Cependant d'autres espèces

biochimiques peuvent jouer ce rôle. C'est le cas de différents ions ainsi que de l'ADN, naturellement utilisés comme porteurs d'information.

3.4.1 Les ions pour le transport de l'information

Une fonction importante d'une membrane cellulaire est de servir de barrière vis-à-vis du monde extérieur. Cependant, les membranes ne sont pas totalement impénétrables [78]. Nous avons vu plusieurs mécanismes d'endocytose et d'exocytose permettant le transport depuis et vers l'extérieur de la cellule de différentes macromolécules comme les protéines. Les membranes sont aussi sélectivement perméables à d'autres espèces biologiques comme certains ions. C'est le cas par exemple des ions calcium [79], dont le rôle important a été démontré dans de nombreux événements cellulaires. Il agit par exemple comme un messager pour la libération de neurotransmetteurs par les neurones, ainsi que dans la régulation de la contraction des cellules musculaires [80].

De récents développements dans la création de canaux transmembranaires synthétiques tendent à appuyer l'utilisation des ions comme porteurs d'information extracellulaire dans des biosystèmes synthétiques [81]. Les premiers canaux synthétiques développés sont relativement rudimentaires mais permettent le passage d'ions comme le sodium et le potassium à travers la membrane. L'élaboration future de canaux plus complexes pourrait permettre leur utilisation en biologie synthétique en tant que mécanismes de transport de l'information entre cellules.

3.4.2 L'ADN comme stockage de l'information

Tous les mécanismes présentés permettent de réaliser des fonctions simples mais qui peuvent devenir des systèmes complexes une fois combinés. Cependant, la principale méthode utilisée actuellement dans les biosystèmes pour le stockage de l'information est de maintenir un état grâce à une expression génétique constante [82]. Cela est très coûteux en ressources et n'est pas forcément sûr à long terme, à cause de fluctuations stochastiques dans les processus cellulaires (voir chapitre 9). L'autre approche consiste à insérer directement l'information désirée à l'intérieur du matériel génétique [83]. La technique employée a permis d'enregistrer *in vitro* l'équivalent d'environ 750 kilooctets de données informatiques sur des brins d'ADN. Les brins ont ensuite été séquencés et les données ont pu être reconstruites sans erreurs. La technique utilisée pour réaliser ce stockage n'est par contre possible qu'*in vitro*.

Une équipe de l'université de Stanford a cependant réussi à développer un biosystème permettant l'enregistrement de données *in vivo* dans de l'ADN en utilisant d'autres techniques [84]. Ce système s'appuie sur des enzymes de recombinaison, afin d'inverser une séquence d'un fragment d'ADN. En fonction de l'ordre de cette séquence, l'information de ce registre génétique

est considérée à '0' ou à '1'. Ce système permet ainsi l'enregistrement d'un bit *in vivo* et cette mémoire génétique est réinscriptible plusieurs fois. L'équipe prévoit aussi d'étendre la capacité de stockage à 8 bits en utilisant plusieurs enzymes différentes.

Ces deux études très récentes sont encourageantes et permettent d'envisager à plus ou moins long terme le stockage de l'information sous forme de registres génétiques et ainsi de pouvoir disposer d'une mémoire non volatile pour des biosystèmes bien plus complexes nécessitant une capacité de stockage.

3.5 Conclusion

Dans ce chapitre, nous avons abordé trois mécanismes biologiques à la base du fonctionnement de biosystèmes synthétiques plus complexes. Nous disposons ainsi du mécanisme de synthèse des protéines, dont l'abstraction numérique nous permet de disposer de la fonction de la porte logique universelle SI-NON. Le mécanisme de liaison entre une protéine et d'autres espèces équivaut au niveau de son abstraction numérique à la fonction de la porte logique ET. Enfin les mécanismes d'endocytose et d'exocytose rendent possible le transport de l'information en dehors de la cellule.

Ces différents mécanismes constituent un ensemble de fonctions qui nous permettent de réaliser la plupart des biosystèmes actuels en les associant. Ces mécanismes constituent bien évidement une base, et bien d'autres devront être pris en compte lorsque la connaissance de leur fonctionnement sera plus avancée, comme c'est le cas pour différents ions intervenant dans le transport de l'information.

Deuxième partie

Flot de conception pour la biologie synthétique et modélisation compacte

Chapitre 4

Flot de conception proposé pour la biologie synthétique

La première partie de cette thèse a été consacrée à la présentation du domaine de la biologie synthétique et aux défis que représente le développement de nouveaux biosystèmes synthétiques. Nous avons notamment vu que la création d'outils d'aide à la conception était indispensable pour accompagner l'essor de ce domaine, et que ceux-ci peuvent intervenir à différents niveaux de la démarche de conception de ces systèmes.

Jusqu'à présent, ces outils ont été développés davantage pour répondre à des problématiques spécifiques que dans un objectif d'utilisation généralisée. Pour pallier ce manque, nous pourrions envisager deux approches pour la mise au point d'une suite complète d'aide à la conception (Genetic Design Automation ou GDA). La première consisterait à construire des interfaces entre les outils existants et à développer les outils manquants. La seconde consisterait à se baser sur un squelette d'une suite logicielle existante ayant fait ses preuves (comme par exemple les EDA utilisés en électronique) et à adapter les outils de la suite aux contraintes spécifiques de la biologie, en s'inspirant éventuellement d'outils existants. Nous avons choisi de privilégier cette dernière approche.

Nous allons dans un premier temps présenter le flot de conception général proposé. Nous détaillerons ensuite les étapes de ce flot de conception, en mettant l'accent sur deux étapes en amont : la synthèse automatique et son optimisation (chapitre 5), et la modélisation et la simulation (chapitre 6). Le chapitre 7, consacré à la modélisation en logique floue, tentera de faire le lien entre les modèles comportementaux et les modèles quantitatifs.

4.1 Le flot de conception général

L'outil proposé se base sur le flot de conception des circuits micro-électroniques mixtes. L'abstraction numérique des comportements biologiques (décrite dans le chapitre 3), permet également l'introduction d'outils spécifiques à la conception des circuits numériques. Ceux-ci ne sont malheureusement pas suffisants dans le contexte de la biologie, et donc l'approche illustrée Figure 4.1 présente une approche hybride analogique/numérique.

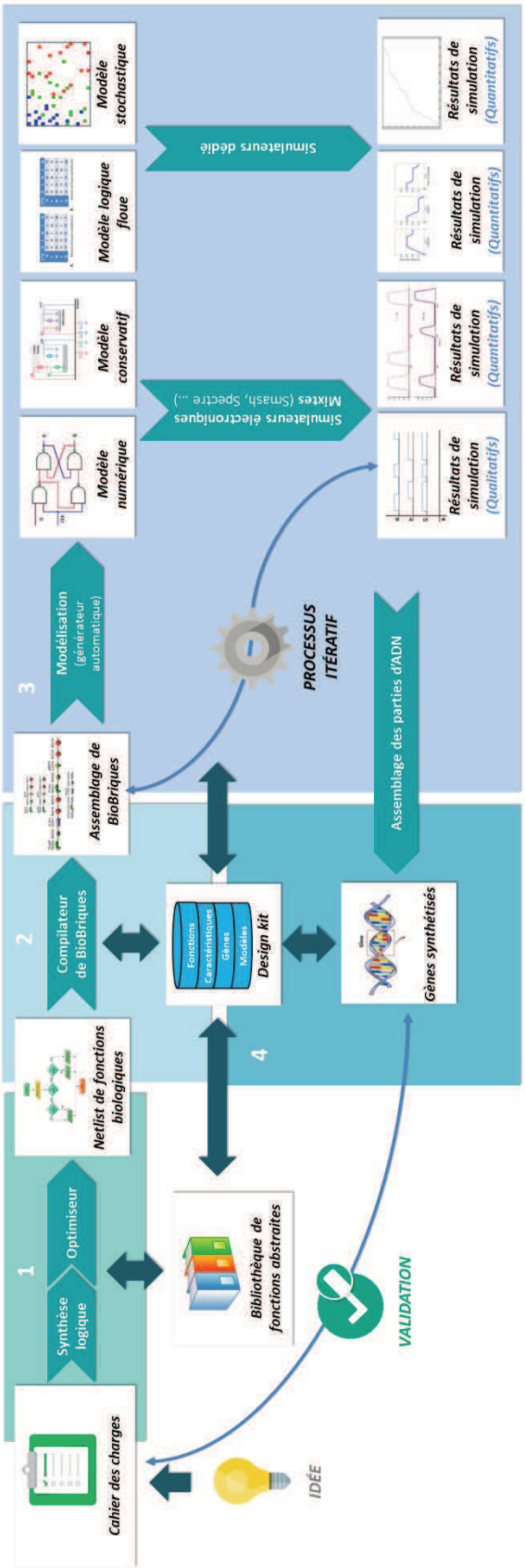


Figure 4.1 : Schéma global du flot de conception destiné à la biologie synthétique, inspiré de la microélectronique.

Le processus débute avec une idée de système biologique à concevoir, qui mène à l'établissement du cahier des charges correspondant. Ce cahier des charges liste les différentes espèces biologiques en entrée et en sortie du système ainsi que les fonctions désirées. Le flot de conception est ensuite découpé en quatre grandes étapes.

4.2 Synthèse fonctionnelle

La première partie, illustrée par le bloc 1 Figure 4.1, concerne la transformation du cahier des charges en une « netlist » de fonctions logiques abstraites. Dans le chapitre 3, il a été démontré que la plupart des fonctions numériques sont réalisables avec des mécanismes biologiques, ce qui permet de réutiliser les premières étapes de conception d'un système d'électronique numérique. Les spécifications sont ainsi réécrites à l'aide de langages de description matérielle (HDL) tels le VHDL ou le Verilog. Ce nouveau cahier des charges, décrit dans un langage compréhensible par les outils existants, va pouvoir être transformé en une description structurale du système, correspondant à une netlist de différents opérateurs numériques élémentaires. Cette étape est réalisée à l'aide des synthétiseurs numériques et des compilateurs RTL.

Jusqu'ici, les étapes de synthèse se font dans un contexte général, sans spécification du matériel utilisé pour sa réalisation (électronique, biologie, pneumatique…) et la netlist RTL générée est la même que pour des systèmes électroniques. Lors de la deuxième étape, la netlist RTL est analysée par un optimisateur afin de l'adapter au mieux aux contraintes biologiques. Cette étape nécessite l'accès à une bibliothèque recensant les différentes fonctions logiques accessibles pour la technologie choisie. L'optimiseur va donc associer la netlist RTL avec les fonctions de la bibliothèque. Le nombre de solutions étant infinies, la bibliothèque doit également contenir, pour chaque fonction biologique, une fonction de coût (dépendant notamment du nombre de protéines nécessaires à sa réalisation, du nombre de gènes utilisés, de l'utilisation préférentielle de certains mécanismes…). L'optimisation consiste donc à trouver l'assemblage de fonctions biologiques formellement équivalente à la netlist RTL passée en entrée et minimisant la fonction de coût globale. A l'issue de cette partie, nous obtenons une liste des différents mécanismes biologiques abstraits (les signaux interconnectés ne sont pas des protéines réelles mais des noms abstraits) ainsi que leurs interconnexions, qui sont fournies aux outils de l'étape suivante. Ce point a été abordé au cours de cette thèse et les résultats obtenus sont présentés dans le chapitre 5.

4.3 Synthèse biologique

L'étape suivante (Figure 4.1 bloc 2) a été intitulée « compilateur de BioBriques » par analogie avec le compilateur silicium des outils de CAO électronique. Il permet de passer d'un assemblage abstrait à un assemblage concret en sélectionnant les BioBriques les plus intéressantes (simplicité ou limitation des interactions) pour la réalisation des fonctions logiques identifiées par la synthèse fonctionnelle. Cet outil doit pouvoir se connecter à une base de données répertoriant les BioBriques et leurs caractéristiques. Cette base de données, que l'on appellera « design kit » en référence aux bases de données similaires existant pour la conception de circuits microélectroniques, constitue l'élément essentiel de ce flot de conception.

A l'issue de la synthèse biologique, les espèces chimiques utilisées pour réaliser les fonctions biologiques sont clairement identifiées et leurs modèles et paramètres associés sont connus. Au cours de cette thèse, ce point n'a été abordé qu'au travers d'un état de l'art des outils existants (présenté dans le chapitre 2) mais n'a pas été approfondi en détail.

4.4 Modélisation, simulation et optimisation

La troisième partie du logiciel (Figure 4.1 bloc 3) est la partie à laquelle nous avons consacré le plus de temps. Le travail effectué sur cet élément a consisté à utiliser l'assemblage des mécanismes biologiques, fournis par le compilateur de BioBriques, pour développer des modèles codés dans des langages compréhensibles par les simulateurs électroniques habituels. Pour les modèles numériques, nous avons employé les analogies, présentées dans le chapitre 3, entre composants électroniques et mécanismes biologiques. Pour les autres modèles développés, les détails seront présentés dans les chapitres associés.

Ainsi, le logiciel peut utiliser la puissance des outils électroniques pour la simulation de systèmes complexes en un temps réduit. Quatre types de modèles peuvent être utilisés dans cette partie. Premièrement, un modèle comportemental qui donne une simulation qualitative et permet de valider à haut-niveau le comportement du système. Le second modèle est un modèle conservatif qui fournit des résultats plus précis. Il permet de suivre l'évolution temporelle moyenne des espèces de manière quantitative pour un ensemble de systèmes. Cette modélisation est décrite plus en détail dans le chapitre 6. Le troisième modèle constitue un niveau de description intermédiaire entre les deux premiers. Cela est nécessaire en biologie car le comportement quantitatif des fonctions est assez éloigné du comportement numérique, ce qui n'est pas le cas en électronique. Cette modélisation à base de logique floue est décrite dans le chapitre 7. Le dernier modèle, un modèle stochastique, est utilisé pour simuler de manière très précise le comportement de chaque espèce du système. Le modèle stochastique peut aussi être décliné en

plusieurs niveaux d'abstraction. Le plus haut niveau consiste à remplacer les équations différentielles par des équations stochastiques (ce qui n'est pas abordé dans cette thèse). Le plus bas niveau consiste à considérer chaque espèce individuellement et à modéliser son déplacement et ses interactions (ce qui est abordé chapitre 10). La plus grosse contribution de notre travail de thèse concerne les différents aspects des modélisations, qui sont abordés en détail dans les chapitres suivants.

Cette étape inclut également une boucle d'optimisation. Réalisée manuellement, elle est utilisée lors de la conception de circuits analogiques. Dans le cas de l'électronique, plusieurs travaux existent dans la littérature pour essayer de l'automatiser [85], [86], [87].

4.5 Finalisation

Enfin, la dernière partie de la suite de conception (Figure 4.1 bloc 4) consiste à passer à la réalisation pratique du biosystème. Elle comprend toutes les étapes permettant de passer de l'assemblage virtuel à l'assemblage réel (choix du châssis, vérification de la compatibilité avec le châssis, génération des séquences génétiques artificielles à intégrer…). Cette partie n'est pas abordée dans ce manuscrit mais l'interface avec des outils existants tels que GenoCAD, Clotho ou encore GeneDesign (présentés au chapitre 2) semble une bonne piste pour réaliser cette étape.

Chapitre 5

Synthèse logique et optimisation

Dans ce chapitre, nous allons passer en revue le travail effectué sur le premier bloc du flot de conception proposé pour la biologie synthétique, comprenant les étapes de synthèse RTL et d'optimisation. Notre approche consistant à adapter les outils déjà existants aux contraintes de la biologie, nous présenterons tout d'abord les options possibles et le choix des logiciels utilisés. Nous détaillerons ensuite le rôle de chaque outil sélectionné pour les étapes de synthèse RTL et de mapping et d'optimisation, et décrirons la bibliothèque de composants biologiques développée. Deux exemples d'application de ces outils seront finalement présentés : une machine d'état permettant la régulation d'espèces biologiques, et la conception d'un microprocesseur biologique. Ce dernier est certes irréalisable concrètement actuellement, mais il permet de bien montrer l'efficacité de la méthode de conception.

Parmi les différents outils de synthèse logique et d'optimisation, les plus couramment utilisés sont des logiciels commerciaux comme Encounter RTL Compiler de Cadence Design Systems [88], intégré dans la suite de conception CAO Cadence, Design Compiler Graphical de Synopsys [89], ainsi que Mentor Graphics EDA [90].

Ces trois logiciels possèdent tous une bibliothèque de composants configurables ainsi que des règles de design adaptables. Ils prennent en charge des fichiers de description comportementale variés, utilisant comme langages de description matérielle le Verilog, le VHDL et même le SystemC. Ils semblent donc être le choix idéal pour l'adaptation à la biologie. Cependant, étant donné que notre but est de proposer une suite complète de conception « open-source » constituée de plusieurs outils, le fait qu'il s'agisse de logiciels commerciaux, dont le code source n'est pas accessible, est rédhibitoire.

Parmi les logiciels libres, la suite Alliance [91] proposée par une équipe du LIP6 se démarque. Il s'agit d'une suite complète d'outils de synthèse de circuits numériques, spécialement adaptée pour les circuits à très forte densité d'intégration de transistors. La description comportementale du système est réalisée en VHDL, mais les fichiers intermédiaires sont rédigés dans un format propriétaire non accessible. La possibilité de rajouter des éléments dans la bibliothèque de cellules est intéressante. Cependant, les outils ne peuvent pas être contraints à l'utilisation d'une seule sorte d'élément, ce qui est important dans notre cas pour imposer certains mécanismes plus avantageux en termes de nombre de gènes par exemple.

Notre choix s'est finalement arrêté sur deux logiciels open-source. Le premier, ODIN II [92], a été développé par une équipe du MIT et nous permet de réaliser l'étape de synthèse logique. La partie optimisation et la génération d'une netlist de mécanismes biologiques sont ensuite gérées par le second logiciel, ABC [93], développé par une équipe de Berkeley.

5.1 Synthèse RTL : ODIN II

L'étape de synthèse RTL est donc réalisée par le logiciel ODIN II. La description comportementale du système à synthétiser est fournie à cet outil en Verilog et ODIN II génère ensuite un fichier contenant une netlist RTL au format BLIF (Berkeley Logic Interchange Format). Ce format de fichier se base sur une description à base de tables de vérité. La synthèse RTL est réalisée de manière automatisée et aucune modification n'a été apportée à ce logiciel dans la mesure où il ne travaille qu'au niveau fonctionnel, sans prise en compte du matériel. ODIN II n'optimise cependant que très peu le résultat de synthèse à partir de la description fournie et rien n'assure que la netlist issue d'ODIN II soit la plus optimale d'un point de vue biologique. Il faut donc intégrer un nouvel outil qui se charge de l'optimisation de la netlist en fonction des contraintes biologiques.

5.2 Mapping abstrait et optimisation : ABC

L'outil choisi pour remplir le rôle d'optimiseur et de générateur de netlist est ABC. Il permet la synthèse et l'optimisation de la logique séquentielle synchrone. ABC part d'une netlist et propose sa réalisation pratique à partir des composants d'une bibliothèque de cellules au format GENLIB. L'optimisation (en électronique en tout cas) se fait en minimisant la taille du circuit, sa consommation, son délai, ou enfin des fonctions des coûts, dépendant de ces trois paramètres.

Le résultat de la synthèse fonctionnelle est une structure de données interne au logiciel et à laquelle nous avons accès dans le code source. Un certain nombre de modules de sortie ont donc été ajoutés à ABC afin de générer des netlists directement compatibles avec VHDL-AMS ou encore SystemC-AMS. Les résultats de simulation obtenus à partir de ces netlists, après modélisation, sont présentés dans le chapitre 6.

5.3 Bibliothèque de composants logiques

La bibliothèque des composants est la clé de la réussite du mapping et de l'optimisation d'ABC. Dans cette section, nous présentons la bibliothèque de composants biologiques développée pour ABC. Nous détaillons ensuite un des composants qui la constitue, la bascule D, qui a demandé un travail d'optimisation préalable. Enfin nous abordons le formalisme GENLIB employé pour la description de la bibliothèque.

5.3.1 Liste des composants

Nous avons standardisé plusieurs composants biologiques qui ont été intégrés dans la bibliothèque des composants. Une liste non-exhaustive de la première version de cette librairie est illustrée Figure 5.1.

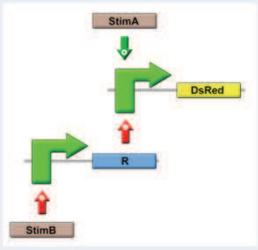
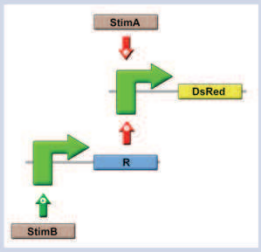
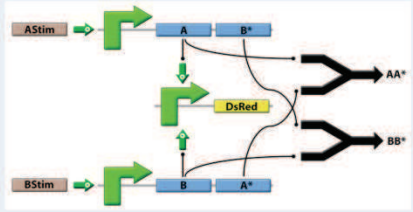
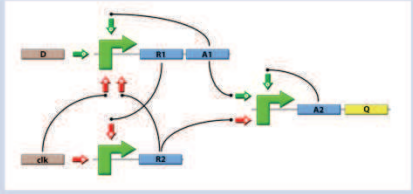
Fonction logique	Schéma	Nombre de gènes	Nombre de protéines
Porte NON-SI		1	1
Porte ET		2	2
Porte NON-ET		2	2
Porte OU		1	1
Porte NON-OU		1	1
Porte OU exclusif		3	6
Bascule D		3	5

Figure 5.1 : Composants biologiques intégrés dans la librairie d'ABC.

Cette liste contient l'ensemble des opérateurs logiques de base ainsi qu'une bascule D, tous réalisés à l'aide de matériel biologique. Le nombre de gènes et de protéines nécessaires à leur réalisation est indiqué pour chaque élément. Pour qu'ABC puisse fonctionner, la bibliothèque doit cependant contenir au minimum un inverseur, un buffer et une porte ET ou NON-ET, ainsi qu'une bascule D pour les systèmes séquentiels. Toutes les protéines mentionnées dans ces composants sont fictives et doivent être sélectionnées par le compilateur de BioBriques à l'étape suivante.

5.3.2 La bascule D

La bascule D est nécessaire dans les systèmes séquentiels. Comme il s'agit d'un composant complexe, formé à partir de composants combinatoires, il a donc été crucial de bien l'optimiser afin de réduire au maximum le nombre de gènes impliqués.

Nous avons commencé par la modélisation d'une D latch. Nous sommes partis du schéma classique, présenté Figure 5.2, basé sur des portes OU-NON plutôt que sur celui constitué de portes ET-NON, plus complexe à réaliser en biologie synthétique.

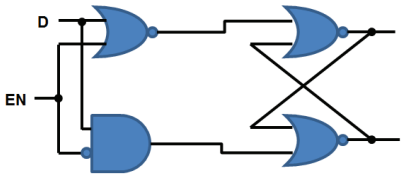


Figure 5.2 : Schéma électronique d'une D latch.

Après transformation de ce schéma en éléments biologiques, nous obtenons le schéma de la Figure 5.3, qui est une D latch biologique, active sur état bas de l'horloge et composée de 4 gènes et de 6 protéines. L'horloge pourrait correspondre à un signal biologique régulier comme l'APC (Anaphase Promoting Complex) qui intervient à chaque division cellulaire.

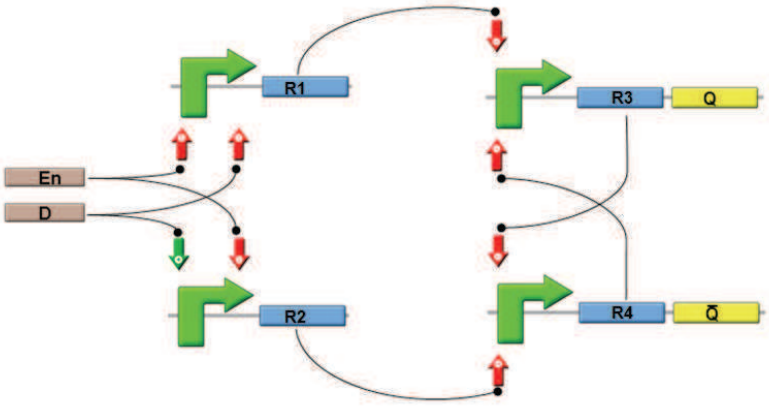


Figure 5.3 : Schéma d'une D latch biologique.

L'approche classique pour obtenir une bascule D active sur fronts montants est de mettre en cascade deux D latchs en structure maitre-esclave. Cependant, celle-ci nécessite 9 gènes pour fonctionner, ce qui est trop important. Nous avons donc développé une solution alternative utilisant moins de gènes. Nous nous sommes basés sur le schéma de la D latch et nous avons amélioré son étage d'entrée pour la rendre synchrone. De plus, la sortie $\bar{Q}$ n'étant pas nécessaire, nous avons également simplifié l'étage de sortie. Le schéma biologique de la bascule D obtenue est illustré Figure 5.4.

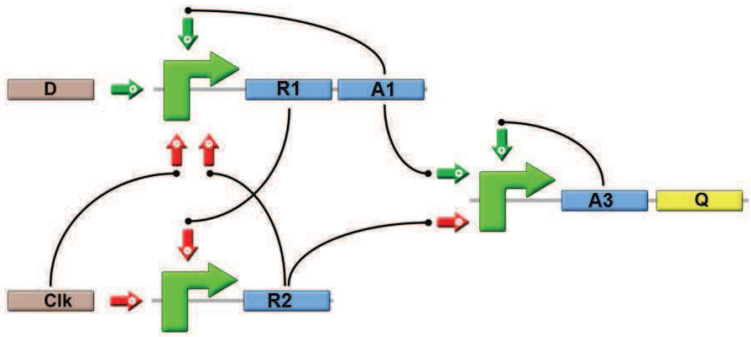


Figure 5.4 : Schéma de la bascule D biologique.

Cette architecture, constituée de seulement 3 gènes, fonctionne selon le principe suivant : le gène activé par D se maintient à un état actif grâce à la protéine A1 qu'il auto-synthétise, tant que l'horloge est à un état bas. La protéine R1, également synthétisée, a pour effet d'empêcher la synthèse de R2. La protéine R2 est synthétisée uniquement quand ni l'horloge ni R1 ne sont présents, ce qui correspond à un front descendant de l'horloge et à un état bas de D. Le gène synthétisant la protéine Q est activé par la molécule A1 et réprimé par la molécule R2. Quand il est activé par la protéine A1, il se maintient actif grâce à la protéine A3 qu'il auto-synthétise. Le comportement de ce système correspond bien à celui d'une bascule D. Les résultats de simulation obtenus avec des modèles bas niveaux tels que présentés au chapitre 6, sont illustrés Figure 5.5.

5.3.3 Le format de la bibliothèque : GENLIB

La bibliothèque a été décrite sous le formalisme GENLIB, géré par ODIN II. Dans ce format GENLIB, il est possible de définir, pour une porte électronique, des paramètres comme la taille de la porte sur le silicium, son temps de propagation… Parmi ces paramètres, nous renseignons la taille de la porte pour chaque élément biologique. La taille correspond en réalité à une fonction de coût calculée pour chaque composant. Cette fonction dépend de plusieurs paramètres comme le nombre de gènes utilisés, le nombre de protéines impliquées dans le système, la complexité de sa réalisation… De fait, en réalisant une synthèse dont l'optimisation porte sur la réduction de l'espace silicium occupé par le système, cela revient à faire une optimisation en

minimisant la complexité des composants du biosystème. Un exemple du fichier GENLIB de la bibliothèque des composants est illustré dans la Figure 5.6.

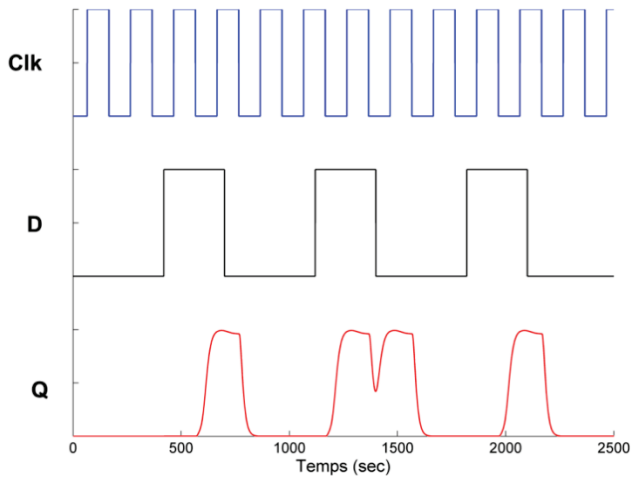


Figure 5.5 : Résultats de simulation de la bascule D à trois gènes.

```
# GATE <cell-name> <cell-area> <cell-logic-function>
# PIN <pin-name> <phase> <input-load> <max-load> <rise-block-delay> <rise-fanout-
delay> <fall-block-delay> <fall-fanout-delay>
GATE    INV     1.00  s=!a;
  PIN   a   INV   0.1  1  1.00  0.00  0.90  0.00
GATE    AND     3.00  s=a*b;
  PIN   a   NONINV    0.1   1  2.50  0.00  1.00  0.00
  PIN   b   NONINV    0.1   1  2.50  0.00  1.00  0.00
GATE    NAND    3.00  s=!(a*b);
  PIN   a   INV   0.1   1  2.50  0.00  1.00  0.00
  PIN   b   INV   0.1   1  2.50  0.00  1.00  0.00
GATE    NOR     1.00  s=!(a+b);
  PIN   a   INV   0.1  1  1.00  0.00  1.40  0.00
  PIN   b   INV   0.1  1  1.00  0.00  1.40  0.00
GATE    OR      1.00  s=a+b;
  PIN   a   NONINV    0.1  1  1.00  0.00  1.40  0.00
  PIN   b   NONINV    0.1  1  1.00  0.00  1.40  0.00
GATE    IF      1.00  s=a*!b;
  PIN   a   NONINV    0.1  1  1.00  0.00  1.40  0.00
  PIN   b   INV   0.1  1  1.00  0.00  1.40  0.00
GATE    BUF     1.00  s=a;
  PIN   a   NONINV    0.1  1  1.00  0.00  1.00  0.00
GATE    ZERO    0.00  s=CONST0;
GATE    ONE     0.00  s=CONST1;
```

Figure 5.6 : Exemple de la bibliothèque GENLIB développée pour ABC.

Enfin, il faut ajouter une autre contrainte lors de la synthèse vers du matériel biologique : la possibilité de différencier la fonction biologique d'une espèce donnée. En effet, une protéine ne peut avoir, par exemple, à la fois la fonction d'activateur et de répresseur. Cela reviendrait à dire que dans certains cas elle accélère l'effet de l'ADN polymérase, et que dans d'autres cas elle la ralentit. Dans le fichier GENLIB, une indication « INV » ou « NONINV » est donnée, afin d'identifier si l'entrée de la fonction biologique nécessite un signal de type activateur ou un répresseur. Deux cas de figure peuvent poser problème : la protéine en entrée est un activateur et nous avons besoin en réalité d'un répresseur ; une protéine activateur ainsi qu'une protéine répresseur sont nécessaires pour la suite du système. Ils sont traités par ABC grâce aux deux composants

illustrés Figure 5.7 que nous avons ajouté dans le fichier GENLIB. Pour le premier cas nous avons intégré un buffer (Figure 5.7.A) et pour le deuxième une synthèse simultanée d'un activateur et d'un répresseur (Figure 5.7.B).

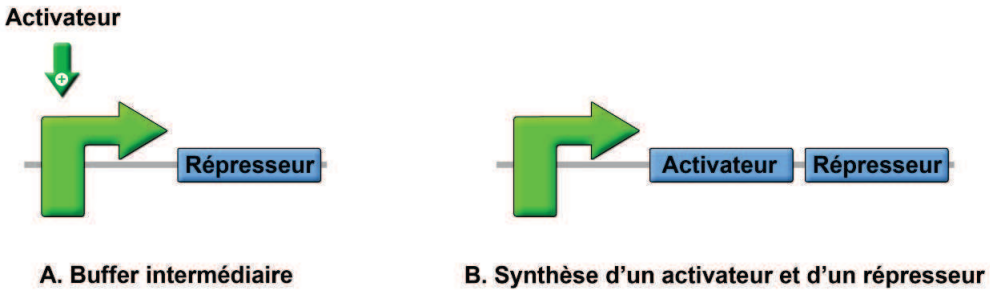


Figure 5.7 : Deux méthodes permettant de connecter une entrée INV sur NONINV et réciproquement.

Les délais sont eux aussi fixés de manière arbitraire en prenant 1.0 pour un mécanisme de synthèse et 0.5 pour un mécanisme de complexation. Cela implique que la synthèse possède un temps de propagation plus long que la complexation car elle se fait en plusieurs étapes. Les paramètres de charges sont également définis arbitrairement à 0.1 pour une entrée d'un mécanisme de synthèse et à 1 pour une entrée d'un mécanisme de complexation. Ce point est important, car il empêche qu'une molécule soit utilisée pour autre chose à partir du moment où elle est impliquée dans un mécanisme de complexation. Ceci a pour but d'éviter l'effet de charge de la complexation. Par exemple, si un gène codant pour la synthèse de la protéine A est exprimé, une certaine quantité de A va être synthétisée, ce qui va avoir pour conséquence d'activer un deuxième gène. Si A est également impliqué dans une réaction de complexation, les A synthétisés par le premier gène vont être immédiatement consommés par le second et la concentration de A baissera, ce qui risque d'empêcher l'activation du deuxième gène.

5.4 Exemples

Pour illustrer la synthèse RTL, le mapping et l'optimisation qui forment la première étape du flot de conception, nous présentons deux exemples. Le premier est celui d'une machine d'état régulant la concentration d'une espèce et le deuxième est la conception d'un processeur biologique.

5.4.1 Machine d'état

Le premier exemple présenté est celui d'une machine d'état permettant de réguler la concentration d'une protéine X. Elle active la synthèse de X lorsque sa concentration passe en dessous d'un seuil et celle d'une protéine Y, permettant de dégrader X par une réaction de type $X + Y \leftrightarrow XY$, quand la concentration de X dépasse un autre seuil. Un système de ce type pourrait

être utilisé, par exemple, dans la régulation de l'insuline chez les diabétiques. Le principe de fonctionnement du système cible est décrit sur la Figure 5.8.A. Il s'agit du pendant biologique d'un thermostat réversible, un exemple classique, utilisé en électronique pour illustrer les notions de logique séquentielle et de machine d'état.

La machine d'état prend comme entrées deux protéines A et B qui vont dépendre de la concentration de la protéine X. Les concentrations des protéines A et B sont gérées par un bloc précédant la machine d'état. Celui-ci compare la concentration de X par rapport à trois seuils, X_F, X_M, et X_E, correspondant respectivement à la concentration minimale, moyenne et élevée de X. L'abstraction numérique de la concentration des protéines A et B est imposée en sortie de ce bloc à :

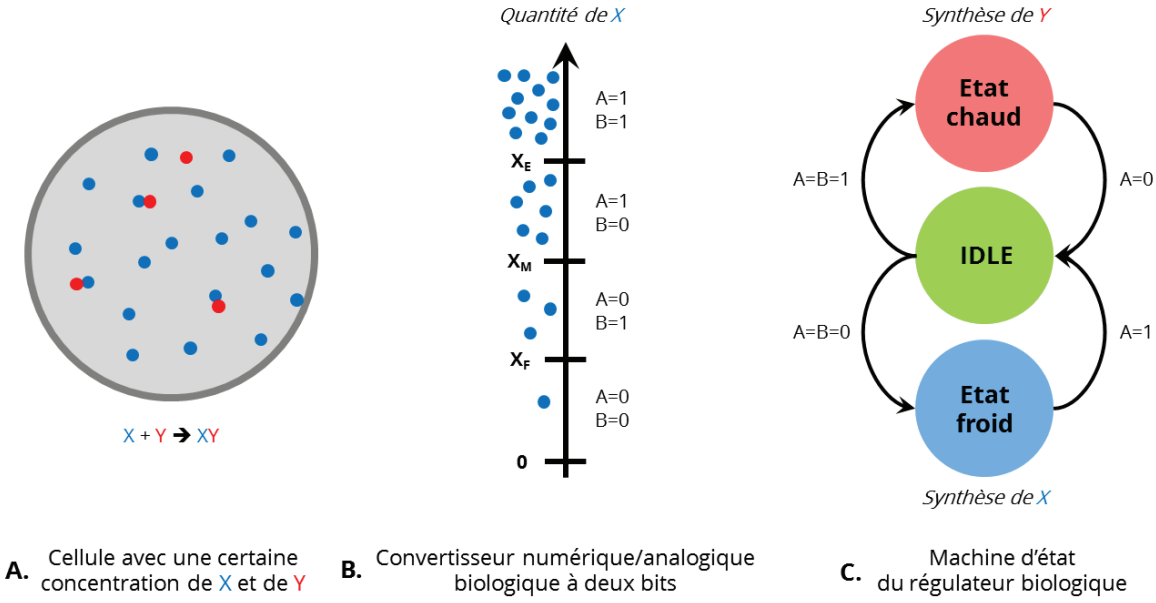
- 0 et 0, si la concentration de X est inférieure à X_F
- 0 et 1, si la concentration de X est entre X_F et X_M
- 1 et 0, si la concentration de X est entre X_M et X_E
- 1 et 1, si la concentration de X est supérieure à X_E.

Ces différentes conditions sont résumées Figures 5.8.B et C. La machine d'état quant à elle dispose de deux sorties : la première sert à activer la synthèse de X et la deuxième sert à activer la synthèse de Y. La machine d'état, illustrée Figure 5.8.C, dispose donc de trois états possibles :

- IDLE, où aucune action n'est effectuée.
- Etat « Froid », où la synthèse de X va être activée.
- Etat « Chaud », où la synthèse de Y va être activée.

Le changement d'état se fait donc de l'état IDLE vers l'état chaud lorsque la concentration de X dépasse X_E (AB=11) et repasse à IDLE dès la diminution de la concentration de X en dessous de X_E (A redescend à 0). De même, le passage de l'état IDLE vers l'état froid est déclenché pour une concentration de X passant sous X_F (AB=00), et le retour à IDLE se fait dès l'augmentation de la concentration de X au-dessus de X_F (A remonte à 1).

Après cette description du système désiré, nous la faisons passer au travers des différentes étapes de la synthèse fonctionnelle, conduisant à la proposition d'un système biologique abstrait, permettant de réaliser la fonction ciblée. Le fichier Verilog (Figure 5.9), fidèle retranscription du cahier des charges, a été fourni à ODIN II.



A. Cellule avec une certaine concentration de X et de Y

B. Convertisseur numérique/analogique biologique à deux bits

C. Machine d'état du régulateur biologique

Figure 5.8 : Illustration de l'action de la machine d'état sur la régulation de concentration de la protéine X.

```
module fsm2 (Clk, Rst, A, B, X, Y);

   input   Clk, Rst, A, B;
   output  X, Y;
   wire    Clk, Rst, A, B, X, Y;
   reg [1:0] state, next_state ;
   parameter idle= 2'b00,hot = 2'b01,cold = 2'b10 ;

// State register
always @ (posedge clock) begin
    if (reset == 1'b1) begin state <= idle; end
    else begin state <= next_state; end
end

// Transition logic
always @ (state or A or B) begin
  case(state)    idle : begin
                          if (A &  B) next_state = hot;
                          else if (!A & !B) next_state = cold;
                        end
                 hot : begin
                          if (!A) next_state = idle;
                       end
                 cold : begin
                          if (A) next_state = idle;
                        end
  endcase
end

// Output logic
assign X = state(1);
assign Y = state(0);

endmodule
```

Figure 5.9 : Code Verilog correspondant à la description comportementale de la machine d'état.

Après la synthèse logique, ODIN II produit un fichier BLIF contenant environs 60 composants, incluant des portes logiques et des bascules synchrones et asynchrones. Le fichier obtenu est ensuite fourni à ABC, qui, après mapping et optimisation avec la bibliothèque décrite sur la Figure 5.1, produit une netlist dont le schéma est illustré Figure 5.10.

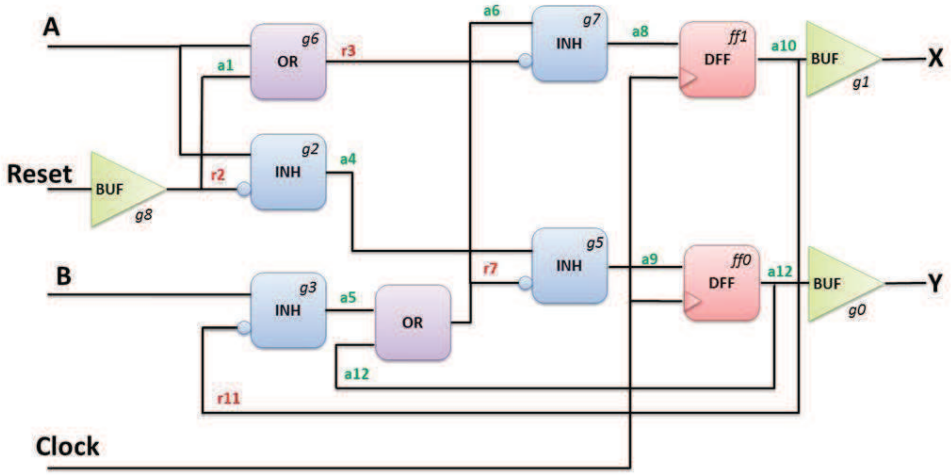


Figure 5.10 : Schéma des composants logiques de la machine d'état, après optimisation d'ABC. Les nœuds indiqués en vert correspondent à des fonctions d'activation et les nœuds rouges à des fonctions de répression.

Cette optimisation conduit à un système totalisant 15 gènes et 26 protéines. Nous remarquons que le synthétiseur a évité d'utiliser des portes ET plus complexes à réaliser d'un point de vue biologique, et pour lesquelles nous avons imposé une pondération importante dans la bibliothèque de composants.

5.4.2 Microprocesseur biologique

Le deuxième exemple présenté est la conception d'un microprocesseur 6 bits biologique. Etant donné sa complexité, la réalisation pratique d'un tel système n'est pas envisageable à court terme. Néanmoins, il constitue un cas d'étude idéal pour tester nos outils. Le schéma général du microprocesseur synthétisé est présenté Figure 5.11.

Ce microprocesseur dispose d'opérations arithmétiques, de branchements conditionnels et non-conditionnels ainsi que des opérations mémoires. Les blocs bleus correspondent aux composants du processeur et les blocs oranges aux entrées/sorties de la machine d'état du processeur.

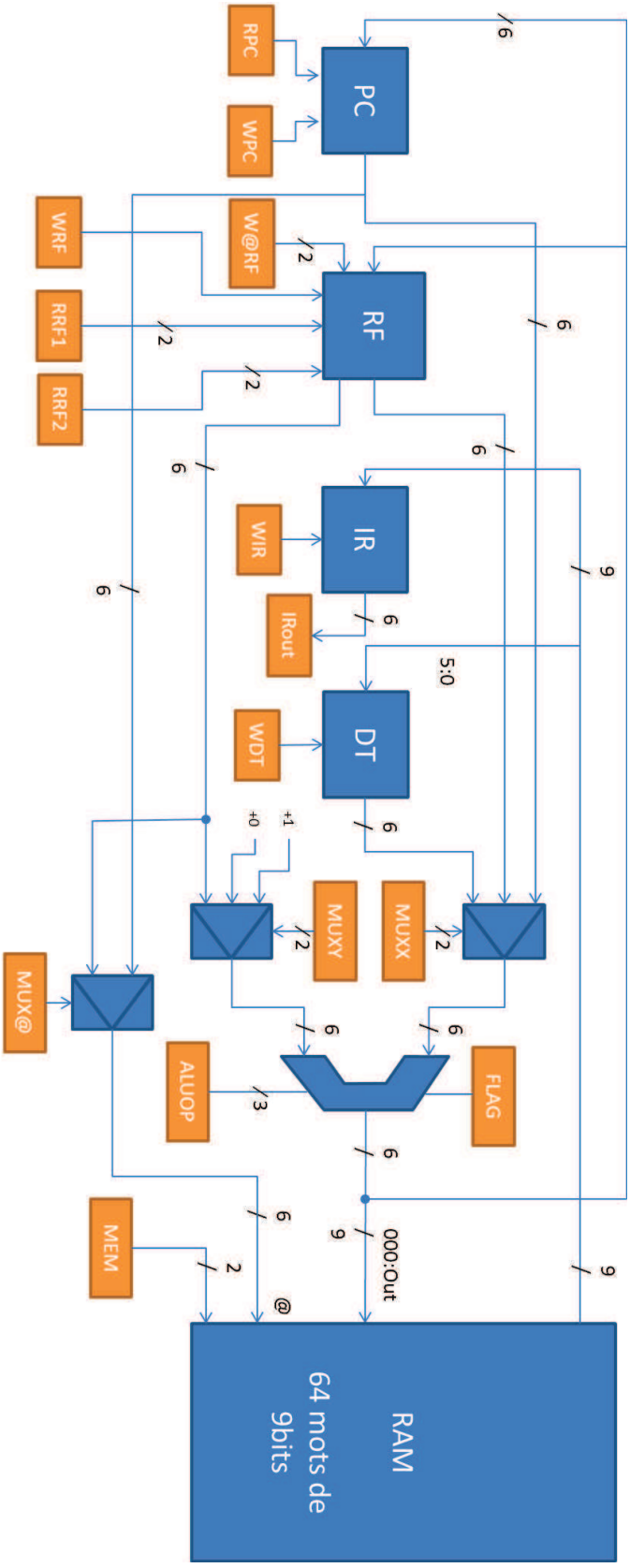


Figure 5.11 : Microprocesseur 6 bits.

Le bloc PC correspond au compteur programme (Program Counter), qui est un registre de 6 bits, contenant l'adresse de l'instruction à exécuter. Les deux entrées associées RPC et WPC correspondent respectivement au Reset, qui sert à mettre le registre à zéro, et au Write du bloc PC, qui actualise le contenu du registre avec le mot de 6 bit présent en entrée.

Le bloc RF est un banc de quatre registres de 6 bits, correspondant au fichier de registres (Register File). Il dispose de deux sorties et des entrées RRF1 et RRF2, qui permettent de choisir l'adresse qui est recopiée vers chacune des sorties. Les entrées W@RF et WRF servent quant à elles à mettre à jour le contenu des registres.

Le bloc IR correspond au registre d'instruction (Instruction Register). Il s'agit d'un registre de 9 bits contenant l'instruction à exécuter. Il est commandé par WIR, qui permet d'activer la copie de l'instruction présente en entrée vers le registre. IRout est la sortie du registre contenant l'instruction à exécuter.

Le bloc DT est un registre de 6 bits, contenant les six derniers bits d'un mot de la RAM. Les trois premiers bits non stockés correspondent à l'opérande et les six bits stockés dans ce registre à la donnée sur laquelle sont faites les opérations. Le signal WDT sert à mettre à jour le registre.

Les multiplexeurs MUXX et MUXY permettent de sélectionner les opérandes de l'unité logique et arithmétique. Le multiplexeur MUX@ permet de sélectionner l'adresse qui va être lue ou écrite dans la RAM.

Enfin, l'entrée ALUOP de l'unité logique et arithmétique permet de sélectionner l'opération effectuée par l'ALU, au choix entre l'addition, la soustraction et les fonctions logiques ET, OU, OU exclusif, ou NON-ET.

Une fois décrit en Verilog, ce microprocesseur a été simulé. Tous les composants, excepté la machine d'état, ont été simulés séparément. Le programme suivant a été mis en œuvre afin d'utiliser toutes les opérations possibles du processeur, et de vérifier son bon fonctionnement avec la machine d'état.

```
0: JMP 8          // Saut à l'adresse 8
1: LDIMM R1 2     // Chargement de 2 dans le registre R1
2: LDIMM R2 63    // Chargement de 63 dans le registre R2
3: LOAD R3 R2     // Chargement dans R3 la valeur qui est à l'adresse  pointé
                  par R2
4: LDIMM R2 3     // Chargement de 3 dans le registre R2
5: BC R0 R3 R2    // Si R0=R3 saut à PC+R2
6: ADD R0 R1      //R0=R0+R1
7: JMP 5          // Saut à l'adresse 5
8: LDIMM R0 32    // Chargement de 32 dans le registre R0
9: JMP 1          // Saut à l'adresse 1
63: 40            // constante 40
```

Ce programme est une boucle qui, à chaque itération, incrémente de 2 la valeur d'une variable initialisée à 32. Une fois que la variable atteint 40, le programme se réinitialise et recommence. Le fonctionnement du processeur a été validé par des simulations numériques. Nous sommes ensuite passés à la synthèse de celui-ci. Dans un premier temps, tous les modules ont été synthétisés séparément. La Table 5.2 liste le nombre de gènes utilisés pour les différents blocs constituant le microprocesseur.

Module	Nombre de gènes
ULA	112
Banc de registre	219
Machine d'états	136
Registre d'instructions	45
Multiplexeurs	120
Program counter	38
Registre DT	30
Total	700

Table 5.2 : Nombre de gènes pour chaque module de processeur.

Une deuxième synthèse a été réalisée avec tous les composants du processeur en une seule fois. Le nombre total de gènes utilisé passe à 618, soit une différence de 82, correspondant à une réduction de plus de 10% du nombre de gènes. En effet, comme l'optimiseur dispose d'une visibilité sur la totalité du système, il a plus de libertés pour le simplifier.

La bibliothèque présentée Figure 5.1 correspond à la première version de la bibliothèque des composants. Afin d'évaluer l'influence des composants présents dans la bibliothèque sur l'optimisation générée par ABC, nous avons développé d'autres bibliothèques intégrant d'autres composants basés sur un seul gène sensible à plusieurs activateurs et répresseurs. Cette idée provient de l'électronique numérique, où des composants sensibles à plusieurs entrées peuvent dans certains cas simplifier grandement un système (portes ET à quatre entrées…). Cette simplification opère sur le nombre de composants utilisés, ce qui serait très bénéfique en biologie, et sur les temps de propagation, qui ne sont pas un point bloquant. Ces bibliothèques sont constituées des éléments suivants :

- Bibliothèque 0 : Bibliothèque de base de la Figure 5.1.
- Bibliothèque 1 : Bibliothèque 0 avec en plus un gène sensible à deux activateurs A1 et A2 et un répresseur R1 (correspondant à la fonction logique $A1.A2.\overline{R1}$)

- Bibliothèque 2 : Bibliothèque 1 avec en plus un gène sensible à un activateur A1 et deux répresseurs R1 et R2 (correspondant à la fonction logique $A1.\overline{R1}.\overline{R2}$)
- Bibliothèque 3 : Bibliothèque 2 avec en plus un gène sensible à trois activateurs A1, A2 et A3 (correspondant à la fonction logique OU à trois entrées $A1.A2.A3$)
- Bibliothèque 4 : Bibliothèque 3 avec en plus un gène sensible à deux activateurs A1 et A2 et deux répresseurs R1 et R2 (correspondant à la fonction logique $A1.A2.\overline{R1}.\overline{R2}$)

Nous avons testé ces différentes bibliothèques sur l'optimisation du microprocesseur biologique. Les résultats du nombre de protéines impliquées dans ce système, en fonction des bibliothèques utilisées, sont résumés Table 5.1.

Bibliothèque	Nombre de protéines
Numéro 0	569
Numéro 1	530
Numéro 2	498
Numéro 3	497
Numéro 4	489

Table 5.1 : Influence de la bibliothèque sur le nombre de protéines utilisées.

Nous constatons que plus la bibliothèque est fournie en composants plus le nombre de protéines nécessaires diminue. Cela vient de la plus grande liberté laissée à l'optimiseur pour réaliser les fonctions logiques. Ces résultats prouvent qu'avec l'augmentation future de la complexité des biosystèmes, il sera très intéressant de disposer de gènes multi-activateurs et multi-répresseurs, bien qu'ils soient très complexes à concevoir actuellement, afin de réduire le nombre de protéines impliquées dans un système.

5.5 Conclusion

Dans ce chapitre, nous avons pu voir les adaptations réalisées sur le couple ODIN II/ABC, qui permet de répondre aux besoins de la première étape du flot de conception, correspondant à la synthèse RTL, au mapping des fonctions abstraites et des opérations d'optimisation.

La bibliothèque de composants servant de base à ABC a été développée sur mesure, en intégrant différents éléments combinatoires biologiques ainsi qu'une bascule D particulièrement optimisée. L'étude réalisée sur cette bibliothèque montre que des gènes multi-activateurs et multi-répresseurs sont très intéressants pour diminuer le nombre de protéines utilisées.

Nous avons enfin pu tester cette étape sur deux exemples, une machine d'état dont les résultats de modélisation en SystemC-AMS sont présentés au sein du chapitre suivant dans la section de

génération des modèles, et un microprocesseur biologique, qui permet de montrer l'efficacité de notre approche sur des systèmes très complexes.

Chapitre 6

Modélisation bas-niveau

Dans le chapitre précédent, nous avons vu que les biosystèmes peuvent être décrits de manière comportementale. Le concepteur dispose ainsi d'une abstraction numérique de son système qui lui permet de valider l'assemblage de BioBriques de manière qualitative. Cette modélisation utilise uniquement des fonctions logiques (portes ET, OU, NON...) qui peuvent être éventuellement complétées par des temps de propagation entre l'entrée et la sortie de chaque opérateur. Le schéma équivalent d'une modélisation sous ce degré d'abstraction est illustré Figure 6.1.A. Cependant, ce degré d'abstraction n'est pas suffisant pour obtenir des résultats de simulation quantitatifs.

Pour pallier ce manque, nous avons développé des modèles bas-niveau, reposant sur des équations différentielles ordinaires (ODEs). Elles nous permettent de décrire plus finement les mécanismes biologiques. Après simulation de ces modèles, nous obtenons des résultats quantitatifs prédictifs sur lesquels le concepteur peut se baser pour sélectionner les différents éléments de son biosystème.

Nous allons dans un premier temps présenter les équations modélisant les mécanismes de complexation et de synthèse des protéines. Ensuite, nous détaillerons les différents modèles développés : le modèle flux de signal (Figure 6.1.B) et le modèle conservatif simple (Figure 6.1.C) ainsi que son évolution intégrant un châssis biologique sur lequel les différents blocs du système viennent se connecter (Figure 6.1.D). Ensuite, nous aborderons leur implémentation dans deux langages utilisés en microélectronique pour la simulation multi-domaines : le VHDL-AMS et le SystemC-AMS. Enfin, nous présenterons plusieurs méthodes de génération automatique des modèles que nous avons développées.

6.1 Equations différentielles ordinaires

Les mécanismes biologiques utilisés dans la conception de systèmes peuvent être représentés sous la forme d'équations biochimiques reliant les espèces biologiques impliquées [94]. Nous pouvons ensuite transformer ces équations chimiques en équations différentielles ordinaires (ODEs). L'évolution moyenne des différentes espèces dans une cellule peut ainsi être évaluée de manière temporelle et quantitative. Nous allons donc passer en revue les équations utilisées pour modéliser les mécanismes de complexation et de synthèse des protéines.

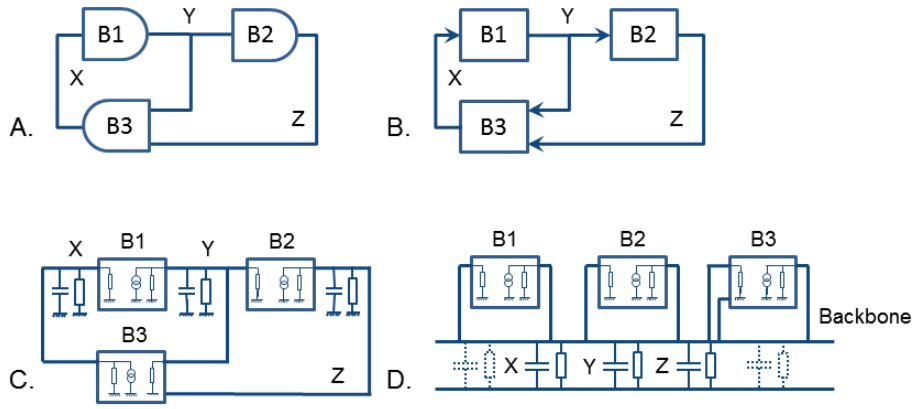


Figure 6.1 : Représentation des différents modèles pour un ensemble de BioBriques. A. le modèle logique, B. le modèle flux de signal, C. le modèle conservatif simple, et D. le modèle conservatif avec le châssis biologique, liant tous les blocs.

6.1.1 Mécanisme de complexation

Le mécanisme de complexation, présenté plus en détail dans le chapitre 3, consiste en la liaison entre une macromolécule A et un ligand B. Ce mécanisme est illustré Figure 6.2.

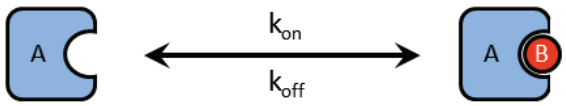


Figure 6.2 : Mécanisme de complexation d'une macromolécule A avec un ligand B.

Ce mécanisme dépend de deux constantes : k_{on}, le taux de liaison entre les deux molécules, et k_{off}, le taux de dissociation. Il est ainsi représenté par l'équation biochimique (6.1).

$$A + B \underset{k_{off}}{\overset{k_{on}}{\longleftrightarrow}} AB \tag{6.1}$$

Nous transformons ensuite cette équation en ODE (équation (6.2)), nous permettant d'obtenir la concentration du complexe AB en fonction de la concentration des espèces A et B.

$$\frac{\partial [AB]}{\partial t} = k_{on} \cdot [A] \cdot [B] - k_{off} \cdot [AB] - k_{degr} \cdot [AB] \tag{6.2}$$

Le dernier terme qui vient compléter cette équation représente la dégradation du complexe dans le milieu et dépend de la concentration de AB, ainsi que d'un coefficient de dégradation k_{degr}. Il

représente tous les mécanismes conduisant à la dégradation des protéines et il est nécessaire pour éviter de voir certaines concentrations diverger vers l'infini.

6.1.2 Mécanisme de synthèse des protéines

Le deuxième mécanisme que nous avons modélisé, décrit en détail dans le chapitre 3, est la synthèse des protéines. Ce mécanisme peut être décomposé en deux étapes : la synthèse de l'ARN messager à partir de l'ADN, appelée transcription, qui peut être contrôlée par deux espèces, l'activateur et le répresseur, et la synthèse de la protéine à partir de l'ARNm, appelée traduction. Ces deux étapes sont illustrées Figure 6.3.

Figure 6.3 : Mécanisme de la synthèse d'une protéine, réalisée en deux étapes, la transcription de l'ADN en ARN messager, puis la traduction de l'ARNm en protéine.

L'étape de la transcription peut impliquer la liaison des espèces activatrices et répressives sur l'ADN. Elle est modélisée à l'aide d'une équation différentielle dont le terme principal est la modulation du taux de transcription en fonction des protéines régulatrices fixées autour du promoteur. Cette fixation est régie par une équation de type équation de Hill [95]. Cette équation qui fournit la concentration mX de l'ARNm est donnée par :

$$\frac{\partial [mX]}{\partial t} = k_{tr} \cdot \left(a + \frac{1}{\prod_p \left(1 + \left(\frac{K_j}{[X_p]} \right)^{n_j} \right)} \right) - d_{ARNm} \cdot [mX] \tag{6.3}$$

Nous retrouvons dans cette équation les différents paramètres suivants :

- k_{tr}, la constante de la cinétique de transcription,
- a, la constante de transcription libre, qui est très faible et la plupart du temps négligeable,
- K_j, la constante de Hill, représentant la force de l'activateur ou du répresseur,
- n_j, le coefficient de Hill, positif pour un activateur et négatif pour un répresseur,
- d_{ARNm}, le coefficient de dégradation de l'ARNm.

La deuxième étape, la traduction, est ensuite modélisée par l'ODE suivante, qui donne la concentration de la protéine X en fonction de la concentration d'ARNm mX :

$$\frac{\partial [X]}{\partial t} = k_{tl} \cdot [mX] - d_X \cdot [X] \tag{6.4}$$

où k_{tl} est la constante de la cinétique de la traduction et d_X le coefficient de dégradation de la protéine.

En combinant ces deux équations, nous obtenons la modélisation complète du mécanisme de synthèse des protéines. Nous pouvons résoudre ces deux équations en statique et les différents cas de l'étude aux limites obtenus sont présentés Figure 6.4.

[A]	[R]	[mX]	[X]
$\ll K_A$	$\ll K_R$	0	0
$\ll K_A$	$\gg K_R$	0	0
$\gg K_A$	$\ll K_R$	$\frac{k_{tr}}{d_{ARNm}}$	$\frac{k_{tr} \times k_{tl}}{d_{ARNm} \times d_X}$
$\gg K_A$	$\gg K_R$	0	0

→

A	R	X
0	0	0
0	1	0
1	0	1
1	1	0

Figure 6.4 : Etude aux limites des ODEs de la synthèse de protéines.

Quand la concentration de l'activateur est très faible, ou quand la concentration du répresseur est très élevée, la concentration d'ARNm et de protéine tend vers 0. Le seul cas où la protéine est synthétisée correspond à une concentration élevée d'activateur et faible de répresseur. Avec cette étude, nous retrouvons bien la table de vérité de la porte SI-NON, utilisée comme modèle logique de ce mécanisme.

6.2 Modèle flux de signal

La première modélisation effectuée est l'encapsulation des équations des mécanismes dans des blocs qui sont des sortes de boîtes noires. Chaque bloc représente un mécanisme et possède des entrées/sorties. Nous retrouvons l'exemple de la complexation illustré Figure 6.5.

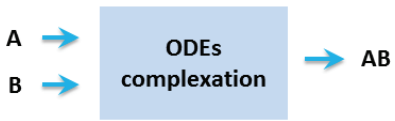


Figure 6.5 : Modèle flux de signal du mécanisme de complexation.

De cette façon, le biosystème peut être représenté par un assemblage de blocs, sous forme de diagramme, illustré Figure 6.1.B. Le modèle tient son nom de modèle flux de signal de cette représentation : une modification sur une entrée d'un bloc va induire le calcul des ODEs qui le composent et le résultat est ensuite répercuté sur les sorties du bloc.

Cependant, cette approche non-conservative présente deux inconvénients principaux. Tout d'abord, l'utilisation d'un diagramme de blocs repose sur l'hypothèse de base selon laquelle l'impédance d'entrée de chaque bloc est considérée comme infinie et l'impédance de sortie comme nulle. Cela impose qu'un bloc *n+1* ne modifie pas la concentration calculée par le bloc *n*. Or cela n'est pas le cas dans plusieurs mécanismes biologiques.

Le mécanisme de complexation est un bon exemple pour illustrer ce problème. La concentration du complexe AB en sortie du bloc de complexation est calculée à partir de la concentration de A et B. Toutefois, la réaction de complexation consomme des protéines A et B pour former le complexe AB. Par conséquent, les concentrations de A et B diminuent lorsque la concentration de AB augmente. Or les concentrations en entrée du bloc de la complexation ne peuvent pas être modifiées, ce qui conduit à des erreurs de modélisation et de simulation. Pour corriger ce problème, il est nécessaire de rajouter des rétroactions entre les blocs, afin d'informer le bloc *n-1* des concentrations calculées au bloc *n*. Cette solution est illustrée Figure 6.6 pour le mécanisme de complexation.

Figure 6.6 : Addition des entrées/sorties pour les rétroactions dans le modèle flux de signal.

L'ajout de rétroactions entre les blocs a trois conséquences principales. La première est le risque de non-convergence d'un système contre-réactionné. La deuxième est un effet de charge lié à l'interconnexion entre les blocs qui devient beaucoup plus complexe en fonction du nombre d'entrées et de sorties des blocs, puisque celles-ci sont doublées, et le nombre d'ODEs à calculer est lui aussi augmenté en fonction du nombre d'entrées. En reprenant l'exemple de la complexation, deux ODEs supplémentaires, une pour chaque entrée A et B, doivent être intégrées dans le bloc :

$$\frac{\partial [A]}{\partial t} = k_{off} \cdot [AB] - k_{on} \cdot [A] \cdot [B] - k_{degr} \cdot [A]$$
$$\frac{\partial [B]}{\partial t} = k_{off} \cdot [AB] - k_{on} \cdot [A] \cdot [B] - k_{degr} \cdot [B] \quad (6.5)$$

Dans le bloc suivant, utilisant le complexe AB, il sera également nécessaire de rajouter une ODE correspondant à AB*. Cela a pour conséquence d'augmenter le temps de simulation proportionnellement au nombre de rétroactions à rajouter et aux ODEs correspondantes.

Le troisième problème de ce type de modélisation concerne la sécurité dans l'élaboration du modèle. Une fois la boîte noire du mécanisme créée, le concepteur n'a plus aucune information sur la nature des entrées et sorties. Par conséquent, avec une modélisation de type flux de signal, il est possible de faire des erreurs de connexion en branchant les entrées/sorties des blocs sur le mauvais type de protéines. Ce problème peut être évité en rajoutant un système vérifiant les compatibilités entre les entrées et les sorties des blocs mais cela alourdi le travail de modélisation. Le modèle conservatif a donc été développé pour répondre à ces deux problèmes.

6.3 Modèle conservatif

Pour résoudre ce problème d'effet de charge des espèces chimiques dans le système biologique modélisé, nous nous appuyons sur des concepts existants et tentons de trouver des équivalences aux lois électriques générales de Kirchhoff (General Kirchhoff Law : GKL) dans le domaine biologique [96]. Chaque partie élémentaire du modèle doit inclure des informations supplémentaires sur la façon dont elle se comporte vis-à-vis des autres éléments. Dans le cas d'une structure hiérarchique, cela revient à définir un ensemble de composants interconnectés qui communiquent et partagent des données via des éléments d'interface appelés ports. La nature de ces ports peut être de type flux de signal directionnels, pour les modèles à temps discret et continu, ou satisfaisant des lois conservatives entre les quantités, pour les modèles à temps continu uniquement. Ces lois de conservation supposent l'existence de deux sortes de quantités : les quantités d'effort (en anglais « across »), correspondant par exemple à la tension pour les systèmes électriques, et les quantités de flux (en anglais « through »), correspondant par exemple à l'intensité pour les systèmes électriques. Ces lois impliquent également que les lois de Kirchhoff sur les potentiels (General Potential Law : GPL) et sur les flux (General Flow Law : GFL) soient respectées. Ces deux lois sont les lois de Kirchhoff utilisées pour les circuits électriques, généralisées à tous types de systèmes de conservation d'énergie, comme les systèmes mécaniques, thermiques ou fluidiques.

En biologie, nous pouvons considérer que chaque molécule impliquée dans le système peut être équivalente à un seul électron. En microélectronique, nous retrouvons la source de courant qui est une source d'électrons. En biologie, un gène, ou tout autre mécanisme pouvant conduire à la synthèse d'une espèce, peut être vu comme une source de protéines. Par conséquent, ce mécanisme peut être représenté par une source de courant contrôlée. De la même manière, un potentiel peut être associé à l'accumulation de charges dans un condensateur en électronique, dont l'une des bornes est reliée à la masse. Le mécanisme équivalent en biologie correspond aux protéines synthétisées s'accumulant dans la cellule. Enfin, une résistance, dont l'une des bornes est reliée à la masse, qui correspond à la consommation ou la fuite d'un électron, peut être utilisée pour représenter le processus de dégradation de la protéine en biologie.

En utilisant cette approche, chaque nœud du réseau électrique correspond à une protéine dont la concentration peut être calculée comme un équilibre entre les différents composants électriques connectés au nœud :

- les sources de courant positives (pour la synthèse de protéines),
- les sources de courant négatives ou des résistances variables (pour la consommation de protéines),
- des résistances parallèles (pour la dégradation naturelle des protéines).

La liste des équivalences entre les composants électriques et les mécanismes biologiques est donnée dans la Table 6.1. L'aspect transitoire est également pris en compte par une telle approche car les capacités introduisent les termes de dérivée temporelle présents dans les différentes ODEs. Le biosystème total peut ainsi être modélisé par un schéma bloc équivalent à celui de la Figure 6.1.C.

Composant électronique	Equivalence biologique
Tension	Concentration des espèces
Source de tension	Concentration constante des espèces
Courant	Flux des espèces
Source de courant	Synthèse/consommation des espèces
Résistance	Dégradation des espèces
Condensateur	Stockage des espèces dans la cellule

Table 6.1 : Equivalences entre composants électroniques et mécanismes biologiques.

6.3.1 Mécanisme de complexation

Pour représenter le mécanisme de complexation à l'aide du modèle conservatif, nous sommes partis des équations (6.2) et (6.5) et nous avons identifié chaque terme grâce au formalisme présenté précédemment. Nous obtenons le schéma électrique équivalent de ce mécanisme, illustré Figure 6.7.

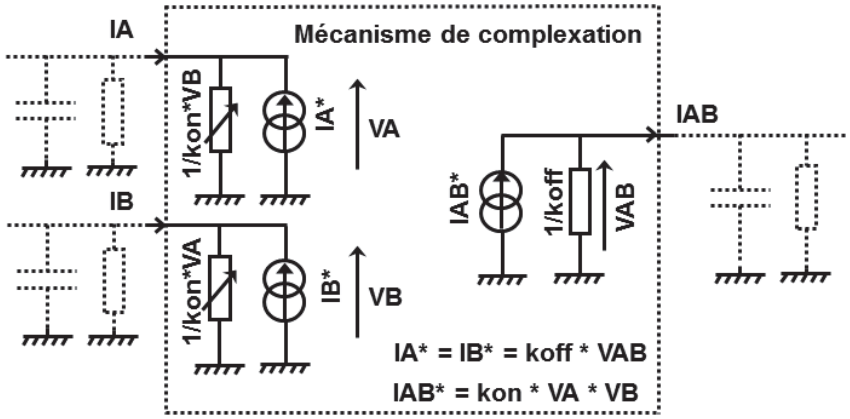


Figure 6.7 : Modèle conservatif du mécanisme de complexation.

A partir de l'équation (6.2), nous identifions un générateur de courant IAB contrôlé par les tensions VA et VB, avec une résistance 1/koff en parallèle, correspondant à la probabilité de dissociation du complexe AB. Pour les espèces A et B, nous retrouvons deux générateurs de courant IA et IB, contrôlés par la tension VAB, avec une résistance variable en parallèle, tous identifiés à partir de l'équation (6.5).

6.3.2 Mécanisme de synthèse

De la même manière, nous pouvons représenter le mécanisme de synthèse des protéines par un modèle conservatif. Les équations (6.3) et (6.4) sont ainsi transformées par indentification en différents composants électriques pour former le schéma électrique équivalent de la Figure 6.8.

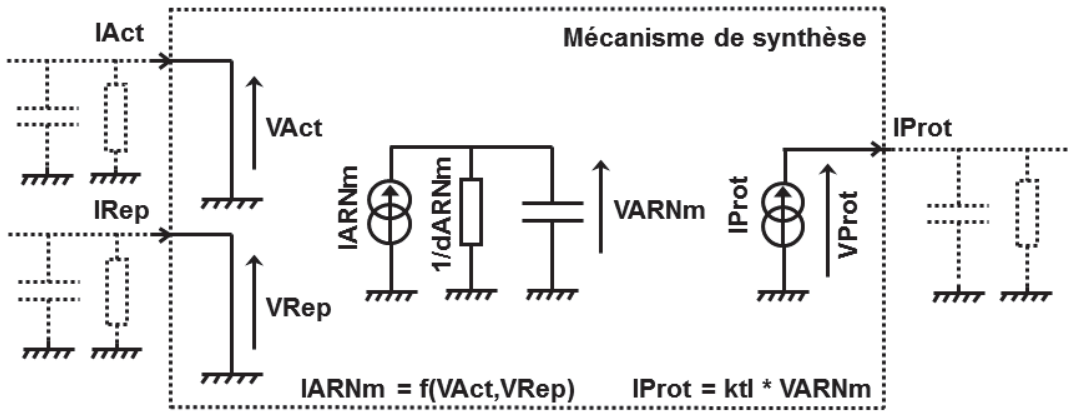


Figure 6.8 : Modèle conservatif du mécanisme de synthèse des protéines.

La synthèse de l'ARNm est ainsi modélisée comme une source de courant I_{ARNm} commandée en tension (VCCS), caractérisée par l'équation suivante déduite de l'équation (6.3) :

$$IARNm = \frac{k_{tr}}{\left(1 + \left(\frac{K_A}{VAct}\right)^{n_A}\right) \cdot \left(1 + \left(\frac{K_R}{VRep}\right)^{n_R}\right)} \quad (6.6)$$

Ce courant vient charger une résistance en parallèle, correspondant à la modélisation de la dégradation naturelle de l'ARNm (de valeur 1/dARNm, équivalent à une résistance de fuite), et un condensateur, correspondant à la modélisation de l'accumulation de l'ARNm synthétisé dans la cellule. Cette partie du schéma correspond ainsi à la modélisation de la transcription.

Le même principe est utilisé pour modéliser le processus de traduction, en utilisant une autre source de courant IProt contrôlée en tension sur la base de l'équation (6.4). Les résistances correspondant à la dégradation naturelle des espèces Act, Rep et Prot, ainsi que les condensateurs, correspondant au stockage de ces espèces dans la cellule, sont placés en dehors du bloc. Ainsi, elles ne sont instanciées qu'une seule fois, quel que soit le nombre de blocs connectés aux nœuds de ces espèces. Nous avons fait le choix de représenter les deux étapes de ce mécanisme en un seul bloc. Si toutefois un autre mécanisme venait interagir avec l'ARNm qui est une espèce interne au bloc, il faudrait utiliser une représentation alternative séparant ce bloc en deux.

6.3.3 Evolution du modèle conservatif

Le premier modèle conservatif présenté permet de répondre au problème du modèle flux de signal quant à la conservation des espèces. Cependant le deuxième problème du modèle flux de signal, qui concerne les erreurs possibles d'interconnexion entre les protéines, n'est pas réglé par le modèle conservatif. En effet, plusieurs espèces différentes peuvent être connectées sur le même nœud ce qui provoque des erreurs de modélisation et de simulation. De toute évidence, l'utilisation des ports électriques pour interconnecter les espèces n'est pas suffisante.

Pour tenir compte de ce problème, la solution réside dans l'utilisation de ports spécifiques à chaque type de protéines utilisées. De cette façon, il n'est plus possible de connecter deux blocs élémentaires avec des entrées/sorties de différents types. En outre, cette fonctionnalité simplifie la représentation d'un modèle, ce qui est illustré Figure 6.1.D. La capacité de stockage de la cellule et la dégradation naturelle des protéines sont modélisées par un ensemble de couples de condensateurs et de résistances pour chaque type de protéine, tous connectés à un bus biologique appelé châssis biologique. Chaque bloc des mécanismes utilisés dans le biosystème est ensuite connecté à ce châssis. Comme les ports des protéines sont maintenant différenciés, il n'y a plus de problème d'erreur d'interconnexion et le couple condensateur/résistance qui sera chargé ou déchargé par un mécanisme est automatiquement choisi en fonction du type des ports des entrées et des sorties des blocs élémentaires.

6.4 Approche fonction versus approche espèce

L'approche de modélisation présentée dans ce chapitre est appelée approche « fonction » car elle consiste à rassembler les éléments du biosystème par blocs remplissant une fonction donnée. Cette approche a été privilégiée car elle a été pensée dans le cadre d'un flot de conception. En effet, l'étape de synthèse logique du flot de conception, présentée au chapitre 5, fournie une netlist des différentes fonctions biologiques utilisées dans le biosystème. La transformation de cette netlist en modèles se fait donc naturellement en gardant cette répartition par fonction. L'exemple du modèle conservatif d'un biosystème utilisant trois fonctions B1, B2 et B3, est illustré grâce à cette approche « fonction » à la Figure 6.9.A.

La deuxième approche consiste à regrouper tous les composants élémentaires du modèle conservatif par espèce. Cette approche est appelée approche « espèce », et est illustrée Figure 6.9.B. Nous constatons que le passage de l'approche « fonction » à l'approche « espèce » est assez simple puisqu'il suffit d'identifier tous les composants connectés sur un même nœud pour une protéine donnée. Le lien entre ces deux approches est illustré sur le modèle conservatif. Dans ce cas, ce sont toutes les ODEs concernant une même espèce qui seront regroupées en une seule ODE. Dans l'exemple de la Figure 6.9, nous n'aurons plus que trois ODEs, pour les espèces X, Y et Z, ce qui simplifie la simulation du modèle. Cependant, ce regroupement n'est pas forcément aisé à réaliser dans le cas des biosystèmes où de nombreux blocs modifient une même espèce.

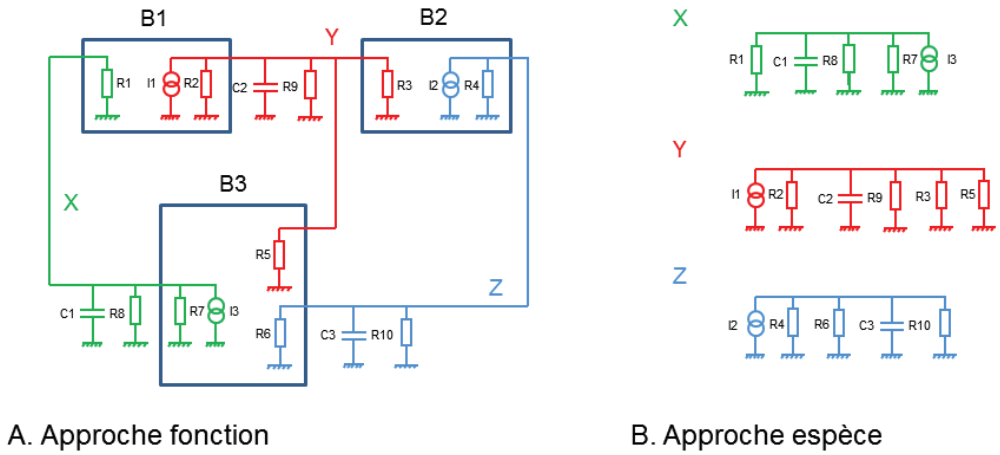


Figure 6.9 : Schémas de l'approche « fonction » et de l'approche « espèce ».

Nous pouvons aussi constater que l'évolution proposée pour le modèle conservatif est en réalité un mélange de ces deux approches. Nous conservons la répartition des mécanismes du système en blocs représentant les différentes fonctions, mais tous ces blocs sont connectés au châssis

biologique qui est un bloc pensé sur une approche « espèce ». Il intègre toutes les espèces présentes dans le biosystème et permet ainsi une connexion simplifiée de nouvelles fonctions.

6.5 Implémentation en VHDL-AMS

Le premier langage de description matériel utilisé en microélectronique dans lequel les modèles ont été implémentés est le VHDL-AMS [97]. Il s'agit d'une extension du VHDL comprenant la gestion de signaux analogiques et de signaux mixtes. Il permet la modélisation de systèmes complexes multi-domaines sous plusieurs niveaux d'abstraction. Sa modularité en fait donc un langage de choix pour notre approche de modélisation bas niveau du vivant. Pour réaliser les simulations des modèles développés, nous avons utilisé Dolphin Smash [98] qui est l'un des simulateurs VHDL-AMS les plus aboutis actuellement.

Le modèle « flux de signal » est le premier modèle à avoir été implémenté. VHDL-AMS permet de décrire et d'instancier structurellement des systèmes de manière aisée, par l'utilisation de ports d'entrées/sorties. Nous avons utilisé le type de données *quantity* de VHDL-AMS pour représenter les concentrations des espèces. La description des ODEs n'a pas posé de problème non plus, le langage permettant l'écriture directe d'équations différentielles grâce à l'opérateur *dot*. Un exemple de code du modèle flux de signal pour une complexation et pour une synthèse de protéine est donné en Annexes B.1 et B.2.

Le VHDL-AMS est un langage qui supporte de manière native la modélisation conservative et qui permet de gérer la description à l'aide des quantités *across* et *through* des lois de Kirchhoff. Pour le modèle conservatif simple, nous avons d'abord utilisé la nature *electrical* de VHDL-AMS pour décrire les différents ports. Cette nature possède une tension comme quantité *across*, et une intensité comme quantité *through*. Ces deux quantités ne sont pas très parlantes dans l'application biologique, où nous avons respectivement des concentrations et des flux de protéines qui correspondent à ces quantités. Le langage permet cependant de créer des natures personnalisées. Pour le modèle conservatif simple, nous avons donc créé une nature *biological* possédant une concentration comme quantité *across* et un flux comme quantité *through*.

L'évolution du modèle conservatif nécessite quant à lui la création d'une nature spécifique pour chaque espèce impliquée dans le biosystème. Cela rend donc possible l'intégration de ce modèle en VHDL-AMS. En contrepartie, il est nécessaire de décrire chaque jeu de composants (la résistance et la capacité) pour chaque type de nature créée, alors que dans le modèle conservatif simple, il suffisait de décrire un seul jeu pouvant ensuite être instancié pour les différentes espèces. Cela complexifie le développement du modèle conservatif d'un biosystème par des tâches répétitives, et c'est entre autres pour cette raison que nous avons choisi de développer un logiciel de génération automatique de modèles qui est abordé plus loin dans ce chapitre. Un

exemple du code du modèle conservatif est présenté en Annexes C.1 et C.2 pour le mécanisme de complexation et de synthèse des protéines. Un exemple du code du paquetage biological_systems est présenté en Annexe C.3.

6.6 Implémentation en SystemC-AMS

La bibliothèque SystemC-AMS est une extension de SystemC développée en C++ [99]. Elle gère plusieurs formalismes de modélisation appelés modèles de calcul (en anglais Models of Computation : MoC) pour la conception des systèmes analogiques et mixtes [100]. Les parties AMS d'un système peuvent être modélisées en utilisant trois modèles de calculs : le flux de signal temporisé (ou Timed Data Flow en anglais : TDF), le flux de signal linéaire (ou Linear Signal Flow en anglais : LSF) et des réseaux électriques linéaires (ou Electrical Linear Network en anglais : ELN). Ces modèles de calcul peuvent interagir entre eux ou avec d'autres MoC (comme les événements discrets de SystemC) à travers une couche de synchronisation présente dans SystemC-AMS. Les deux modèles de calcul qui nous intéressent pour la modélisation du vivant sont donc le TDF et l'ELN.

6.6.1 Modèle TDF

Le modèle de calcul TDF permet de modéliser des comportements complexes non conservatifs. Il consiste en une approche de modélisation à temps discret qui considère les données comme des signaux échantillonnés dans le temps. Ces signaux sont mis à jour à des points discrets dans le temps et possèdent des valeurs discrètes ou continues, comme les amplitudes.

Ce modèle de calcul correspond parfaitement à notre approche de modélisation flux de signal. Nous avons donc développé notre modèle flux de signal sous le modèle de calcul TDF de SystemC-AMS. La principale difficulté a été l'intégration des équations différentielles du modèle. Contrairement au VHDL-AMS où cette opération a été aisée grâce aux opérateurs disponibles dans le langage, en SystemC-AMS ceux-ci ne sont pas disponibles, ce qui a rendu la tâche plus compliquée. La solution a été de travailler avec les pas de temps de simulation au lieu d'utiliser directement le temps de simulation. Pour retrouver les dérivées temporelles, il a suffi de multiplier les équations par ce pas de temps.

Le deuxième problème rencontré a été une limitation du langage, qui ne permet pas de connecter plusieurs entrées sur une seule sortie, contrairement à la réciproque. Il a fallu développer un bloc intermédiaire qui permet de sommer N entrées et de restituer cette somme sur une seule sortie. La même limitation a été rencontrée lors du branchement de plusieurs blocs sur une seule sortie. Un deuxième bloc a donc été développé pour recopier une entrée sur N sorties. Ces deux blocs n'ont aucune signification physique mais servent à combler les lacunes

du langage. Un exemple du modèle flux de signal codé en TDF pour les mécanismes de complexation et de synthèse des protéines est présenté en Annexes D.1 et D.2.

6.6.2 Modèle ELN

Le modèle de calcul ELN permet la description d'un réseau électrique linéaire et dispose d'une bibliothèque d'éléments prédéfinis comme des résistances, des capacités, etc. Il inclut aussi plusieurs modules de conversion avec le modèle de calcul TDF. Le module ELN correspond ainsi à notre approche de modélisation conservative. La description de ce modèle en ELN est facilitée par l'instanciation des différents composants qui est très simple à réaliser. Cependant, nous nous sommes heurtés à quelques problèmes.

Comme ce modèle de calcul n'est composé que de composants linéaires, il a été nécessaire de passer par des blocs TDF pour décrire les générateurs de courant modélisant la synthèse des espèces, ainsi que par des résistances variables modélisant la dégradation contrôlée. Ce problème peut entraîner des erreurs de calcul aux interfaces TDF/ELN, et ce, malgré les blocs de conversion spécifiques utilisés. Il entraine aussi un allongement du temps de simulation, à cause de l'utilisation conjointe des deux solveurs (ELN et TDF) pour simuler un système.

La notion de nature présentée dans la section sur le VHDL-AMS n'est pas existante en SystemC-AMS. Tous les nœuds sont d'un type équivalent au type *electrical* de VHDL-AMS et ne peuvent pas être différenciés par des attributs. L'implémentation de l'évolution du modèle conservatif possédant un bus biologique connecté à un châssis biologique nécessiterait donc des développements importants dans la version actuelle du langage. Dans les Annexes E.1 et E.2, le code correspondant au mécanisme de complexation et à la synthèse des protéines est présenté pour le modèle conservatif développé en ELN.

6.7 Comparaison des deux langages

La comparaison entre le VHDL-AMS et le SystemC-AMS effectuée dans cette section porte sur plusieurs points. Le premier concerne le modèle flux de signal. Le simulateur VHDL-AMS possède un pas de temps adaptatif qui peut être contraint entre un minimum et un maximum, mais c'est le simulateur qui sélectionnera le pas en fonction des résultats de simulation obtenus. Le modèle de calcul TDF possède quant à lui un pas de temps fixe pour un bloc donné mais il est adaptable d'un bloc à l'autre pour éviter de fixer tous les pas des différents blocs sur le processus le plus rapide. De plus, le modèle de calcul TDF est procédural, ce qui permet d'éviter de résoudre des équations, une étape gourmande en temps, mais seulement de réaliser des affectations. L'ordonnancement statistique des différents modules apporte également un gain de temps. Cela

a pour conséquence de réduire le temps de calcul pour l'implémentation d'un modèle en SystemC-AMS par rapport à son implémentation en VHDL-AMS.

Le modèle conservatif bénéficie par contre de la personnalisation des natures correspondant aux quantités des lois de Kirchhoff en VHDL-AMS. Même si elle n'implique aucun changement au niveau des résultats de calcul du modèle, cette fonctionnalité offre un confort d'utilisation au concepteur car elle évite des erreurs de connections. De plus, l'implémentation en SytemC-AMS requiert l'ajout de blocs TDF dans le modèle ELN pour les composants non linéaires, ce qui alourdit le modèle.

En ce qui concerne les aspects moins techniques, le VHDL-AMS n'est pas open-source, contrairement au SystemC-AMS. De ce fait, les simulateurs efficaces pour le VHDL-AMS, comme Dolphin Smash, sont payants, et ne peuvent pas être redistribués librement, alors que SystemC-AMS intègre directement un simulateur open-source performant développé par Fraunhofer-Gesellschaft. De plus, SystemC-AMS est une librairie développée en C++, ce qui rend possible l'intégration conjointe de n'importe quelle autre librairie C++ aux modèles développés. Cela pourra être très pratique pour la suite du projet, en permettant notamment l'intégration d'une librairie de gestion de connexion à des bases de données pour aller récupérer les paramètres des différentes BioBriques dont le modèle à besoin. A cause de l'aspect plus fermé de VHDL-AMS et des simulateurs associés, cette fonctionnalité n'est pas possible sans outil externe.

Des études comparatives concernant l'évolution des temps de simulation entre les deux langages avec l'augmentation du nombre de mécanismes vont être réalisées afin de confirmer ou d'infirmer l'avantage de l'un ou de l'autre en fonction des modèles. Les différentes fonctionnalités résultantes de la comparaison entre les deux langages sont résumées Table 6.2.

Fonctionnalité	VHDL-AMS	SystemC-AMS
Modèle flux de signal	✓	✓
Modèle conservatif	✓	✓
Châssis biologique	✓	—
Ajout de fonctions tierces	—	✓
Liens outils de microélec.	✓	✗
Simulateur intégré	✗	✓
Open-source	✗	✓

Table 6.2 : Récapitulatif de la comparaison des deux implémentations des modèles.

En conclusion de cette étude, le langage SystemC-AMS semble être plus intéressant pour l'application que nous souhaitons faire des modèles. La distribution de l'ensemble de notre suite de conception automatisée pour la biologie synthétique n'est pas compatible avec le VHDL-AMS. Cependant ce dernier présente des avantages sur le SystemC-AMS au niveau de l'élaboration des modèles.

6.8 Générateurs de modèles automatiques

Afin de générer automatiquement les modèles présentés dans ce chapitre, nous avons développé plusieurs approches. La première a été de créer un logiciel de génération automatique de modèles totalement indépendant qui puisse être utilisé facilement par un biotechnologue. La seconde a été de concevoir un module de sortie pour le logiciel ABC, permettant de transformer directement la netlist générée par cet outil en modèles. Enfin, la troisième, encore à l'étude, est de proposer un convertisseur du langage utilisé par les biologistes pour la description de biosystèmes vers les langages de description matériel de la microélectronique.

6.8.1 Générateur de modèles indépendant

Deux raisons principales ont mené au développement de ce générateur automatique de modèles. La première est le nombre important de fichiers à générer pour un biosystème donné. La seconde est de faciliter l'accessibilité des bio-ingénieurs aux langages de modélisation électronique. L'outil est codé en C++ et le framework de Qt a été utilisé pour développer l'interface graphique. Le simulateur utilisé est Dolphin Smash. Nous allons voir comment sont générés les modèles au niveau du logiciel, puis son application sur plusieurs exemples.

L'architecture du logiciel est basée sur deux parties principales : l'interface graphique et l'algorithme, illustrées Figure 6.10. L'interface graphique fournit à l'utilisateur une interface claire, lui permettant de définir l'ensemble de son système. Depuis cette fenêtre, il peut interagir avec l'algorithme pour saisir les différentes espèces, les mécanismes ainsi que le système entier, mais aussi modifier les paramètres. Un guide d'utilisation détaillé est présenté en Annexe F.

Ensuite, l'algorithme de génération des modèles récupère la structure du système défini par l'utilisateur dans l'interface graphique et génère quatre sortes de fichiers :

- un ensemble de fichiers d'espèces, un par espèce concernée, incluant les différents fichiers nécessaires ;
- le fichier du système qui contient la description structurelle du système biologique ;

- le fichier testbench, qui contient une instanciation du système biologique ainsi que les mécanismes associés ;
- le fichier de compilation requis par le simulateur pour compiler les différents fichiers du modèle ainsi que pour exécuter les simulations et pour afficher les résultats.

La stratégie utilisée dans l'algorithme consiste à remplir des fichiers gabarits avec les informations fournies par l'utilisateur via l'interface graphique. Le formalisme des modèles permet la génération de tous les modèles nécessaires à partir d'un ensemble réduit de fichiers gabarits.

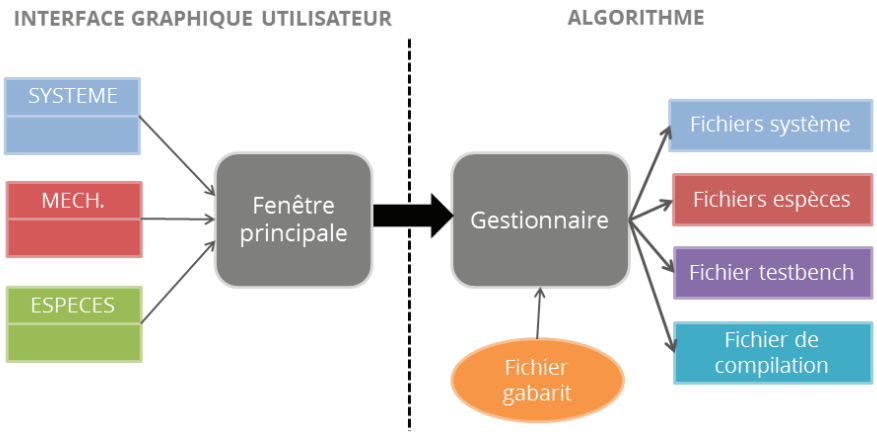


Figure 6.10 : Architecture du générateur de modèles automatique.

6.8.2 Exemples

Nous avons appliqué le générateur de modèles sur trois exemples de biosystèmes synthétiques dont les résultats ont été publiés récemment dans la littérature. Le premier est un des biosystèmes combinatoires les plus complexes à ce jour. Le deuxième exemple est un demi-additionneur biologique, constitué d'une porte ET et d'une porte OU exclusif qui implique des rétroactions. Le dernier est un oscillateur synthétique, nativement rebouclé et donc potentiellement instable. Les paramètres par défaut des modèles des différents mécanismes biologiques utilisés sont résumés Figure 6.11.

6.8.2.1 Détecteur de cellules cancéreuses

Pour ce premier exemple, nous considérons un biosystème synthétique conçu par Xie *et al.* [101] dont le but est de différencier les cellules cancéreuses des cellules saines. Cette tâche est réalisée en régulant la synthèse d'une protéine fluorescente rouge DsRed, en fonction de la concentration d'espèces spécifiques, les microARNs. Ce sont des ARNs très courts, présents chez les organismes eucaryotes. Ils ont le pouvoir de réprimer l'expression d'un gène en se fixant à l'ARN

messager produit par le gène, ce qui conduit à la dégradation de cet ARN messager et empêche ainsi la traduction. Leur rôle exact dans la cellule reste encore méconnu, mais des études ont montré que l'expression ou l'absence de certains de ces microARNs peut constituer un moyen d'identifier si une cellule est cancéreuse ou non [102]. C'est à partir de cette propriété que Xie *et al.* ont bâti un système permettant de détecter la signature en termes de microARNs dans une cellule HeLA. Il s'agit de la plus ancienne lignée de cellules cancéreuses étudiée, et elle tient son nom d'Henrietta Lacks, la patiente sur lesquelles ces cellules ont été prélevées à l'origine.

Paramètres	Valeurs
K_{xa}	0.2
K_{xr}	0.02
k_{tr}	0.1
k_{tl}	0.1
d_{ARNm}	0.1
d_p	0.1
a	0
n	2.0

A. Synthèses de protéines

Paramètres	Valeurs
k_{on}	0.9
k_{off}	0.1
k_{degr}	0.5

B. Complexation

Figure 6.11 : Paramètres par défaut des modèles des mécanismes biologiques.

Une cellule est considérée comme maligne en présence des microARNs miR-21, miR-17 et miR-30a et en l'absence des microARNs miR-141, miR-142 et miR-146a. Si nous considérons ce système sous une abstraction numérique, nous obtenons l'équation logique suivante :

$$DsRed = miR\text{-}21 \,.\, miR\text{-}17 \,.\, miR\text{-}30a \,.\, \overline{miR\text{-}141} \,.\, \overline{miR\text{-}142} \,.\, \overline{miR\text{-}146a} \tag{6.7}$$

La détection de l'absence des microARNs est assez facile à réaliser en raison de leur nature. Le gène codant pour DsRed est ainsi conçu pour être une cible pour miR-141, miR-142 et miR-146a. En présence de ces microARNs, DsRed n'est pas synthétisé. En revanche, la présence des microARNs est plus difficile à détecter. La solution proposée par Xie *et al.* repose sur un système possédant une double inhibition, illustrée Figure 6.12.

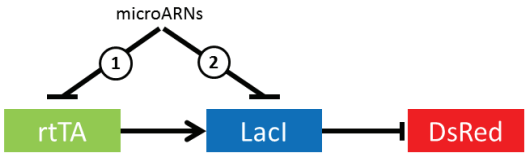


Figure 6.12 : Sous-système de détection de la présence de microARNs, avec l'illustration de leur action répressive sur rtTA et/ou LacI.

Ce sous-système est constitué de trois gènes : le premier synthétise rtTA, le deuxième synthétise LacI et est activé par rtTA, et le troisième synthétise DsRed et est réprimé par LacI. L'expression des deux premiers gènes peut être inhibée par les microARNs. Quand les microARNs sont présents, ni rtTA, ni LacI ne sont synthétisés, ce qui permet la synthèse de DsRed.

D'un point de vue numérique, le second mécanisme d'inhibition est redondant et il suffirait en théorie d'inhiber la synthèse de rtTA pour que le système fonctionne. Cependant, Xie *et al.* ont démontré expérimentalement la nécessité de cette action combinée des microARNs, dans le but d'atteindre des niveaux de fluorescence suffisamment discriminants pour une grande concentration de microARNs. In fine, le système complet est décrit par la figure 6.13.

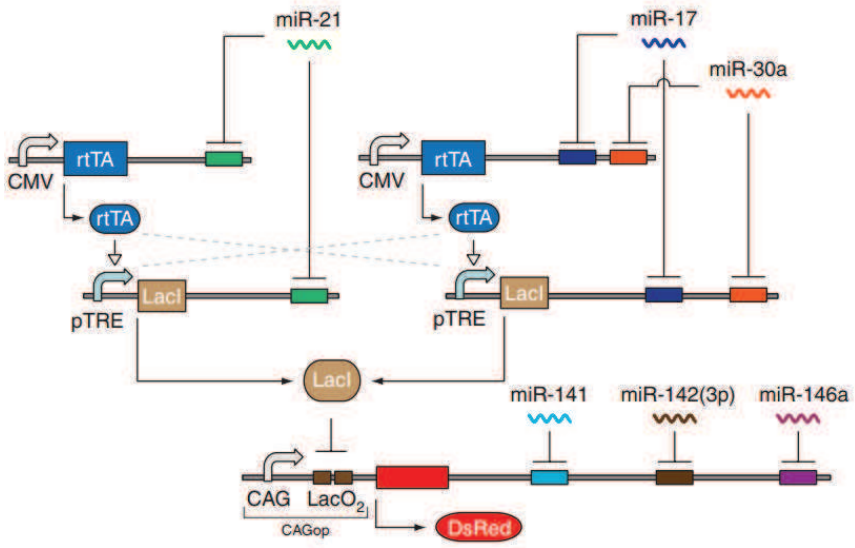


Figure 6.13 : Système complet de détection de cellules cancéreuses [101].

Nous avons saisi ce système dans le générateur de modèle avec les deux cas de figure où les microARNs n'ont une action que sur rtTA puis sur rtTA et sur LacI. Après ajustement des paramètres sur les données expérimentales, nous obtenons les résultats de simulations Figure 6.14.

Les résultats représentés en rouge, correspondent à la double action des microARNs sur rtTA et sur LacI, et ceux en bleu, à la simple action des microARNs sur rtTA, les croix représentant les résultats expérimentaux et les courbes les résultats de simulation. Dans les deux cas, les modèles générés correspondent bien aux résultats expérimentaux.

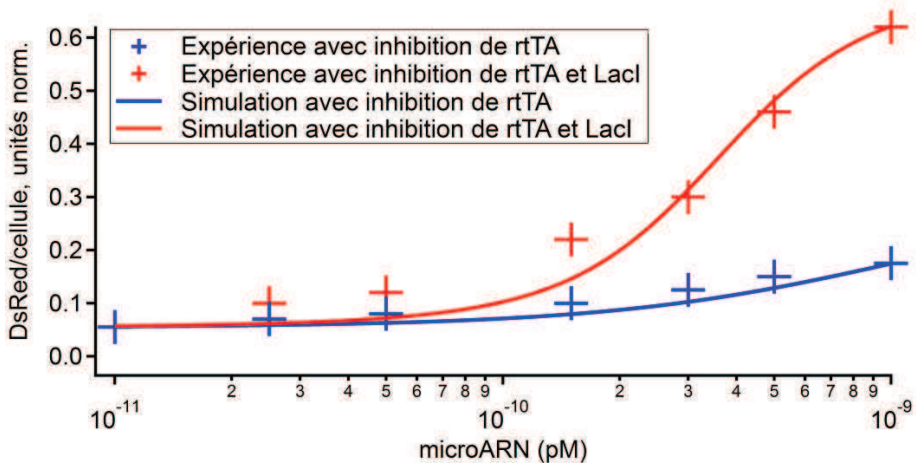


Figure 6.14 : Résultats de simulation comparés aux résultats expérimentaux du système Figure 4.8.

Cependant, au voisinage d'une concentration de 10^{-10} pM de microARNs, nous pouvons observer un défaut d'ajustement entre la simulation et les résultats expérimentaux. Ce défaut provient de la limitation des modèles utilisés pour représenter l'expression du gène. En effet, l'équation de Hill utilisée est connue pour être trop limitée pour modéliser avec précision des comportements complexes. Entre une concentration de 10^{-9} et 10^{-8} pM de microARNs, les courbes expérimentales diminuent également. L'origine de ce phénomène n'a pas été expliquée par les auteurs de la revue et provient sûrement d'un autre mécanisme mis en jeu mais non décrit. Une fois identifié, ce mécanisme pourra être intégré dans la modélisation de ce biosystème. La logique floue, présentée au chapitre 7, permet de régler ces défauts. Le développement de modèles alternatifs plus précis est présenté dans les chapitres de la partie 3 et pourront être intégrés dans le flot de conception par la suite.

Le système entier est modélisé en se basant sur le sous-système présenté. Il est constitué de deux sous-systèmes permettant de détecter respectivement la présence de miR-21, de miR-17 et de miR-30a, et du gène synthétisant DsRed sensible aux microARNs miR-141, miR-142 et miR-146a, en accord avec l'équation (4.1). Une simulation transitoire de l'ensemble du système a été réalisée en présentant les différents cas de figure possibles, illustrée Figure 6.15.

Quand miR-21 et miR-17 ou 30a sont présents en même temps, rtTA et LacI sont complètement dégradés et DsRed est synthétisée. Cependant, quand il y a présence de miR-21 ou 17 ou 30a séparément, rtTA et LacI sont synthétisés dans une concentration moindre mais toujours trop élevée pour activer la synthèse de DsRed. La présence de miR-141, miR-142 et miR-146a, ensemble ou séparément, montre la dégradation de DsRed.

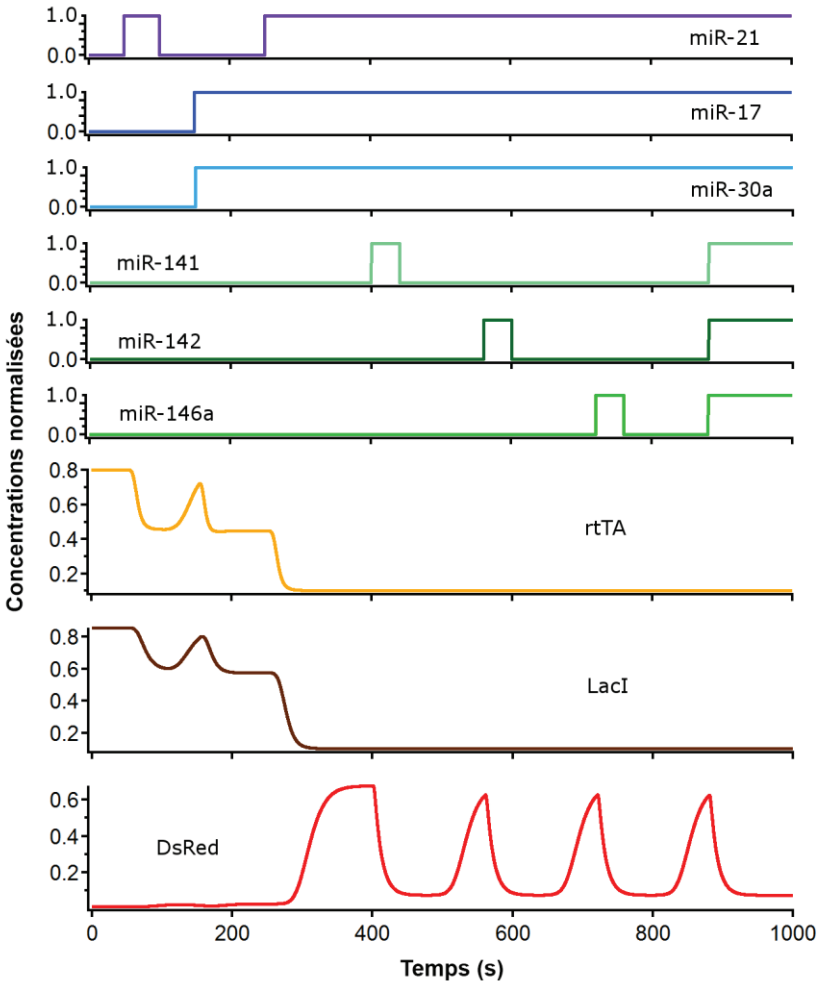


Figure 6.15 : Simulation temporelle du système complet de Xie et al. *[101].*

Les modèles générés de ce système sont constitués de plus de 60 fichiers et de plus de 300 quantités. Un tel système montre que la biologie synthétique a maintenant atteint un niveau où l'utilisation d'un logiciel de génération automatique de modèles est nécessaire.

6.8.2.2 Demi-additionneur biologique

Le deuxième exemple est un demi-additionneur développé par S. Ausländer *et al.* [103]. Il est constitué d'une porte ET et d'une porte OU exclusif et utilise plusieurs gènes dont les différentes interactions sont illustrées Figure 6.16, où les symboles « + » correspondent à des complexations.

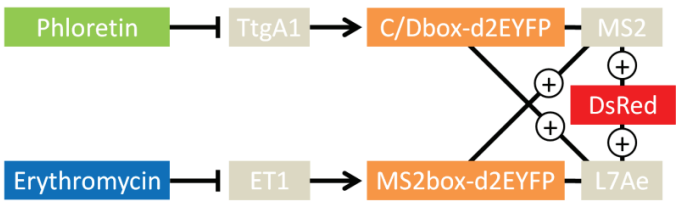


Figure 6.16 : Schéma d'un demi-additionneur biologique.

Les deux entrées du demi-additionneur sont la Phlorétine (Ph) et l'Erythromycine (Er) et les sorties sont la protéine fluorescente jaune d2EYFP, pour la sortie de la porte OU exclusif, et la protéine fluorescente rouge DsRed, pour la sortie de la porte ET. Le comportement du système peut être résumé par une table de vérité, présentée Table 6.3 et les mécanismes biologiques utilisés sont les suivants :

- La Phlorétine et l'Erythromycine sont des répresseurs de la synthèse de TtgA1 et d'ET1. Par conséquent, TtgA1 et ET1 sont synthétisés uniquement si Ph et Er ne sont pas présents dans la cellule.
- La présence de TtgA1 dans la cellule induit la synthèse de C/Dbox-d2EYFP et MS2. La présence d'ET1 a le même comportement sur MS2box-d2EYFP et L7Ae
- MS2 et MS2box-d2EYFP ainsi que L7Ae et C/Dbox-d2EYFP peuvent se lier, ce qui rend la protéine fluorescente d2EYFP inactive.
- DsRed est toujours synthétisée dans la cellule, mais devient inactive lorsqu'elle est liée à L7Ae ou MS2.

Ph	Er	TtagA1	ET1	C/Dbox-MS2	MS2box-L7Ae	YFP	DsRed
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	0	1	1	0
1	1	0	0	0	0	0	1

Table 6.3 : Table de vérité du demi-additionneur illustré Figure 6.16.

Le modèle de ce système a été obtenu à l'aide du générateur automatique de modèles et après ajustement des paramètres, les résultats de la simulation temporelle sont illustrés Figure 6.17.

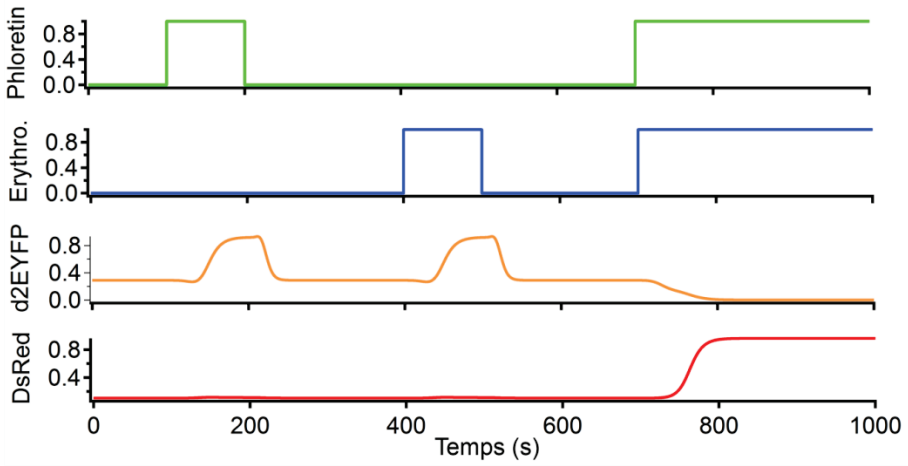


Figure 6.17 : Résultats de simulation temporelle du demi-additionneur.

Les résultats de simulation correspondent bien au comportement attendu ainsi qu'aux résultats expérimentaux. Quand la phlorétine et l'érythromycine sont absentes, les deux sorties sont à une concentration faible. Quand une seule des deux espèces est présente, d2EYFP (correspondant au résultat de l'addition) est à une concentration élevée et DsRed (la retenue de l'addition) est à une concentration faible. Enfin, lorsque les deux espèces sont présentes, DsRed est à une concentration élevée alors que d2EYFP est à une concentration faible.

Des comparaisons quantitatives n'ont cependant pas pu être réalisées, car les valeurs expérimentales de la fluorescence ne sont pas fournies pour les différents cas. Nous observons tout de même un comportement similaire aux images de fluorescence publiées : une fluorescence faible de d2EYFP est observée lorsque les deux entrées sont absentes et le même phénomène est observé sur DsRed lorsqu'une seule des deux entrées est présente.

Cet exemple a été choisi pour montrer la pertinence des modèles générés, dans le cas d'un système nécessitant une approche conservative. En raison des quatre réactions de liaison, cet exemple est plus complexe que le premier du point de vue de la modélisation et de la simulation. Le modèle complet du système est constitué d'environ 60 fichiers de modèles et de plus de 250 quantités.

6.8.2.3 Oscillateur biologique

Le dernier exemple de l'utilisation du logiciel présenté est son application pour la modélisation de l'oscillateur biologique développé par J. Stricker *et al.* [104]. Le schéma du système est illustré Figure 6.18. Ce qui différencie cet exemple des deux premiers est qu'il s'agit d'un système rebouclé, bien connu pour poser des problèmes de stabilité et de convergence des simulateurs.

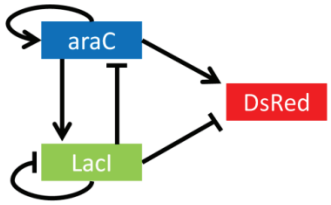


Figure 6.18 : Schéma de l'oscillateur synthétique modélisé.

Le système est composé de trois gènes, codant respectivement pour la synthèse de trois protéines : un activateur araC, un répresseur LacI et une protéine fluorescente DsRed, servant de rapporteur pour observer les oscillations. L'expression de chaque gène est activée ou inhibée en fonction de la concentration d'araC et de LacI. Les oscillations du système sont démarrées par une injection d'arabinose, qui, en se fixant sur araC, en fait un activateur. Cela conduit ensuite à la synthèse de LacI et de DsRed. A partir de là, le système se met à osciller entre deux états : l'état inhibé, où la cellule contient une quantité suffisante de LacI pour réprimer l'expression des trois

gènes, et l'état activé, où la concentration de LacI diminue suffisamment sous une valeur seuil, ce qui permet l'activation de la synthèse des trois protéines. La période des oscillations ne dépend que du retard entre l'activation des gènes et le début effectif de la synthèse des protéines, ainsi que de la vitesse de dégradation des protéines synthétisées.

Ce système est modélisé avec notre logiciel et les résultats de simulation sont présentés Figures 6.19 et 6.20. L'oscillation des trois protéines est bien observée.

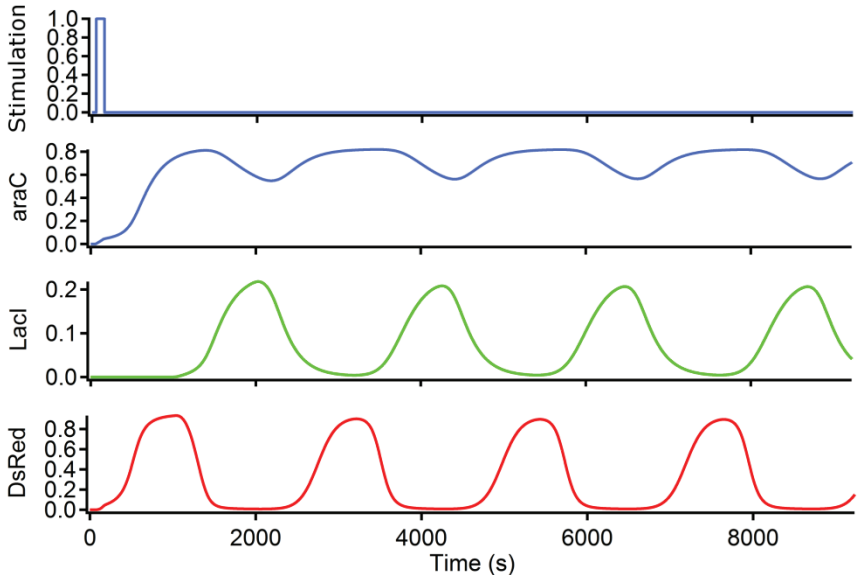


Figure 6.19 : Simulation temporelle de l'oscillateur.

Nous avons aussi comparé les résultats de simulation aux résultats expérimentaux obtenus par J. Stricker *et al.* Cette comparaison est illustrée Figure 6.20, avec la courbe représentant la simulation et les croix les point expérimentaux.

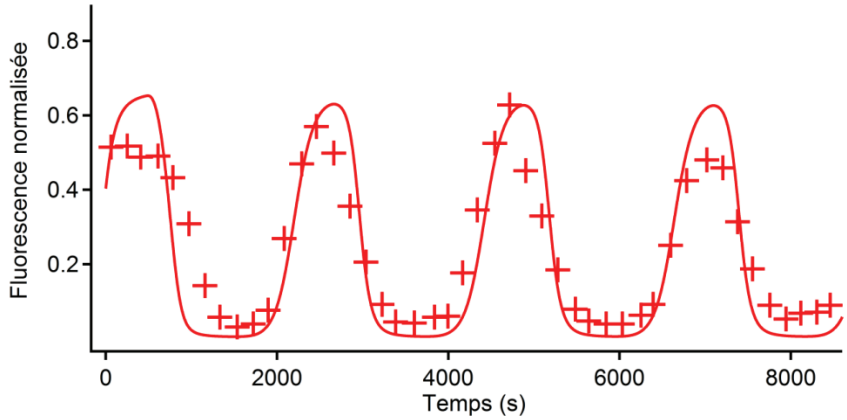


Figure 6.20 : Simulation temporelle de l'oscillateur comparée aux résultats expérimentaux.

Nous constatons que les résultats de simulation correspondent bien aux résultats expérimentaux avec une période d'oscillation similaire. Le système est composé de 20 fichiers de

modélisation et a généré environ 100 quantités VHDL-AMS (*across* et *through*). Bien que la complexité du système soit plus faible en comparaison avec les deux autres exemples, il prouve la robustesse de notre approche dans le cas de la simulation d'un système en boucle fermée.

Nous avons résumé la complexité des trois systèmes présentés dans la Table 6.4 en indiquant pour chacun le nombre de gènes et de protéines utilisés, ainsi que le nombre de fichiers de modélisation et les quantités créées. Ce tableau récapitulatif illustre bien la nécessité d'un outil de génération automatique de modèles, en raison du nombre important de quantités et de fichiers à créer.

Exemple	Détecteur de cellules cancéreuses	Demi-additionneur	Oscillateur
Nb de protéines	16	13	6
Nb de gènes	5	4	3
Nb de liaisons	0	4	0
Nb de quantités VHDL-AMS	300	250	100
Nb de fichiers	60	50	20

Table 6.4 : Tableau récapitulatif des différents systèmes modélisés à l'aide du générateur automatique de modèles.

6.8.3 Interface avec les outils en amonts

L'inconvénient de la méthode présentée dans la section précédente est de devoir ressaisir manuellement tout le système dans un logiciel spécifique. La deuxième approche permettant de générer les modèles automatiquement consiste à repartir des netlist fournies par les logiciels en amont (ABC ou un compilateur de BioBriques) et à les convertir directement en modèles. Pour ce faire nous avons développé un module de sortie pour ABC qui synthétise les modèles correspondant au système dans le langage SystemC-AMS. Le but est de tendre vers un module de sortie générique qui permettrait de générer différents types de modèles directement à partir d'ABC.

6.8.3.1 Module de génération de modèles SystemC-AMS

Nous avons développé un nouveau module de sortie pour ABC ayant la capacité de transformer le résultat de synthèse logique et de l'optimisation en modèles SystemC-AMS. Nous ne rentrons pas ici dans le détail, car ces modèles sont passés en revue dans le chapitre 5.

Nous avons repris l'exemple de la machine d'état de la section 5.4.1, optimisée par ABC, et nous avons généré les modèles correspondants. Leur simulation est illustrée Figure 6.21. Nous constatons que le comportement est bien conforme à ce qui est attendu, avec cependant un retard sur les sorties non prévu dans la simulation numérique mais résultant du temps mis par les protéines pour être synthétisées.

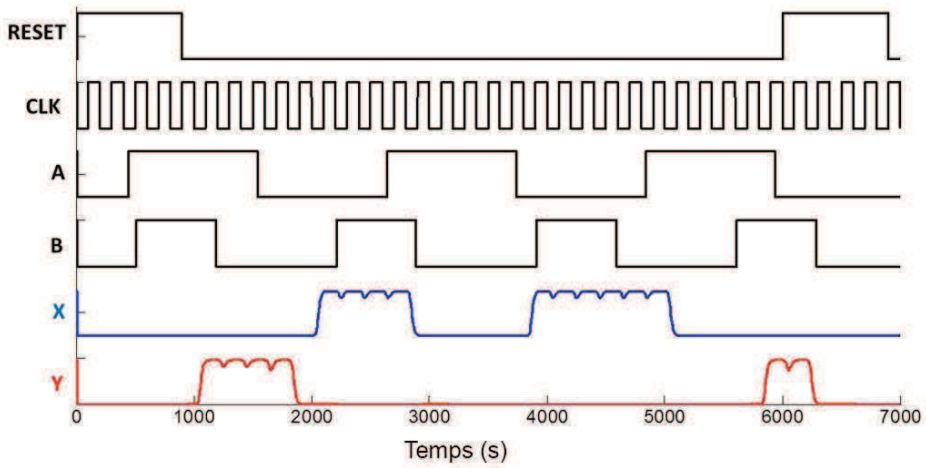


Figure 6.21 : Simulation temporelle de la machine d'état de la section 5.2.4.

Nous avons également réalisé une étude plus poussée sur ce régulateur, en le plaçant en situation réelle. Nous avons donc considéré son implémentation chez un homme pour une application de régulation d'une protéine X. La simulation se déroule sur une journée entière et nous avons élaboré le scénario suivant :

- la concentration de la protéine X à réguler augmente après un repas et diminue continuellement, et ce, de façon plus importante pendant un effort physique ;
- trois horaires correspondant aux repas ont été fixés à 7h, 12h et 19h, et le repas de 12h a été réglé pour apporter une concentration plus importante de X ;
- deux horaires correspondant à des efforts physiques ont été fixés dans l'après-midi.

Nous obtenons les résultats de simulation, dans le cas d'une protéine X régulée et non-régulée, illustrés Figure 6.22. Nous pouvons constater que la concentration de la protéine régulée présente de légères fluctuations autour d'une valeur moyenne, alors que la concentration de la protéine non-régulée présente des fluctuations d'amplitude très importante. Ce comportement correspond bien aux résultats attendus de régulation.

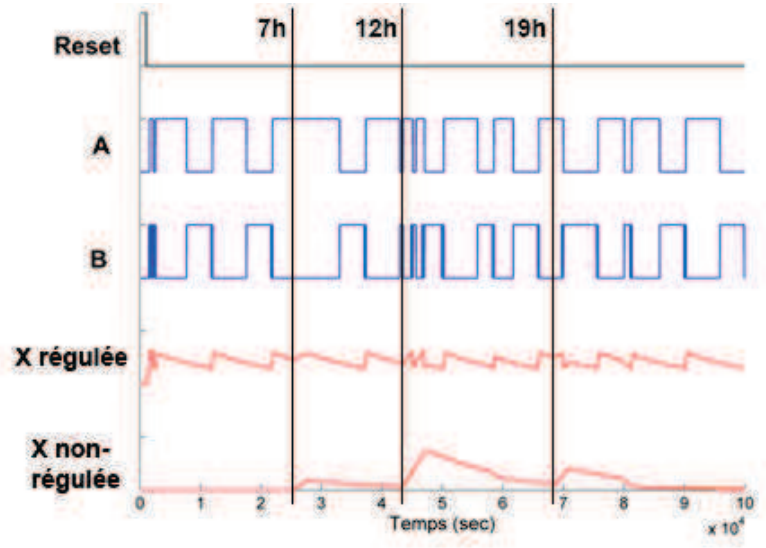


Figure 6.22 : Simulation du régulateur en situation réelle durant une journée.

6.8.3.2 Vers une netlist standardisée

L'idée est de ne pas synthétiser directement les modèles dans le module de sortie d'ABC mais de passer par une netlist standardisée intermédiaire, intégrant les mécanismes et leurs interconnexions, qui seront ensuite transformés en modèles. Le niveau d'abstraction pourra être spécifié, ce qui déterminera les modèles à générer et permettra de disposer d'un seul module de sortie pour tous les types de modèles au lieu d'un par niveau d'abstraction.

La description d'un système serait réalisée selon la nomenclature suivante :

```
#SYSTEM SYSTEM_LABEL {

                #SPECIES {
                …
                }

                #MECHANISMS {
                …
                }

                #CELLS {
                …
                }
        }
```

A l'intérieur du bloc #SPECIES, toutes les espèces biologiques sont listées, les mécanismes utilisés et leur interconnexion avec les espèces sont référencés dans le bloc #MECHANISMS et enfin les interconnexions entre les différents mécanismes formant les sous-systèmes sont saisies dans le bloc #CELLS. Une description plus avancée du formalisme proposé est fournie en Annexe G et son application sur le détecteur de cellules cancéreuses présenté dans la section 6.8.2.1 est illustrée en Annexe G.5.

6.8.4 Import d'une description SBML

La dernière méthode envisagée pour générer des modèles de manière automatique serait l'implémentation d'un convertisseur des langages utilisés par les biologistes pour la description des biosystèmes, comme le SBML (Systems Biology Markup Language) [67]. Les systèmes décrits dans ces langages pourraient être transformés à l'aide du même formalisme employé pour la netlist standardisée et ainsi être modélisés avec les langages de la microélectronique.

Cet outil permettrait de court-circuiter toute l'étape de synthèse logique et de transformer directement le système en modèles. Cette technique empêcherait de ce fait les étapes d'optimisation mais permettrait de bénéficier tout de même de l'efficacité de la modélisation et de la simulation reposant sur les langages électroniques.

Le principal intérêt de cette méthode est de servir aux biologistes disposant déjà de la description de leur système ou à ceux dont le système dispose de certaines spécificités qui ne sont pas prises en charge actuellement par les outils de synthèse logique.

Pour conclure, nous avons résumé le lien entre les modules permettant la génération des modèles (numérique, bas-niveau, logique floue…) avec la netlist de BioBriques dans la Figure 6.23. Nous y retrouvons aussi les différents modules évoqués, permettant de générer la netlist de BioBriques, avec d'une part le couple ODIN II/ABC, l'interface graphique correspondant au logiciel présenté à la section 6.8.1 et enfin la description SBML.

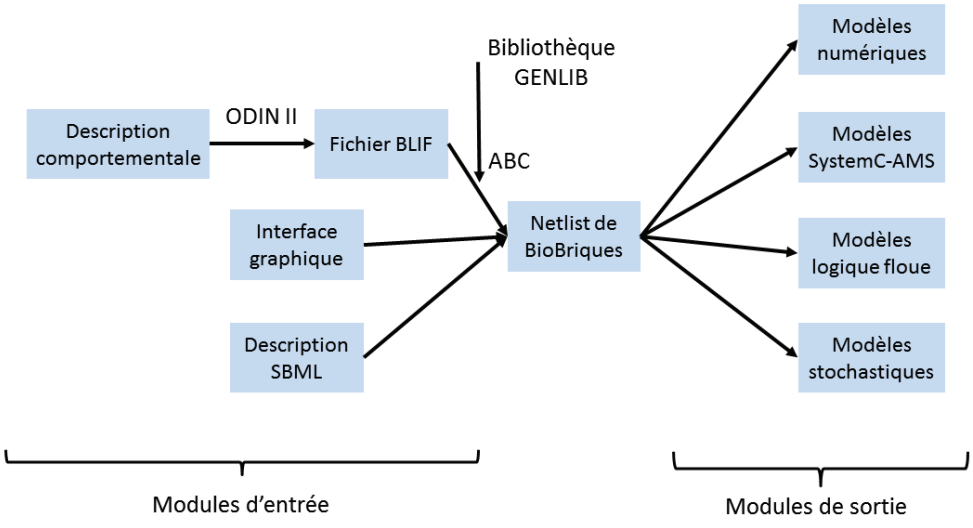


Figure 6.23 : Interaction des différents modules d'entrée/sortie avec la netlist de BioBriques.

6.9 Conclusion

Dans ce chapitre, nous avons présenté la modélisation bas niveau du vivant à travers deux modèles : le modèle flux de signal, directement déduit des équations différentielles ordinaires représentant les réactions biochimiques, et le modèle conservatif, où une identification entre des composants électroniques standards et les différents termes des ODEs a été effectuée. Ils présentent chacun leurs avantages et leurs inconvénients : les rétroactions nécessaires au modèle flux de signal le rendent plus compliqué à instancier, mais il présente des temps de simulation plus faibles, en fonction de l'augmentation de la complexité du biosystème par rapport au modèle conservatif, plus proche de la représentation et du comportement de la cellule, mais nécessitant la résolution des lois de Kirchhoff à chaque nœud.

L'implémentation de ces deux types de modèle a été réalisée dans deux langages de description matérielle AMS, utilisés traditionnellement en microélectronique. Entre ces deux langages, notre préférence va vers le SystemC-AMS pour le modèle flux de signal à cause de sa rapidité d'exécution et au VHDL-AMS pour le modèle conservatif pour ses natures sélectives.

Le développement d'un générateur automatique de modèles a permis de régler le problème du nombre important de fichiers de modèles à générer avec la complexité grandissante du biosystème. Son utilisation sur des cas concrets issus de la littérature a aussi permis de valider les modèles sur des biosystèmes nécessitant la prise en charge de la conservation des espèces ainsi que sur des systèmes rebouclés. Néanmoins, ces modèles dépendent de paramètres biologiques qui doivent être obtenus par une caractérisation expérimentale des BioBriques, ce qui fait encore défaut actuellement.

En regroupant le travail du chapitre précédent effectué sur la synthèse logique avec le module de sortie du logiciel ABC, générant automatiquement les modèles SystemC-AMS, nous avons créé une machine virtuelle sur laquelle tous ces outils sont présents, et qui permet de servir de preuve de concept de notre approche.

Chapitre 7

Modélisation intermédiaire à l'aide de logique floue

Nous avons présenté dans les deux chapitres précédents deux niveaux d'abstraction extrêmes des mécanismes biologiques, mais tous deux nécessaires au flot de conception. Le premier est comportemental mais permet de manipuler des signaux discret ce qui est très utile pour les phases de conception en amont. Le second est quantitatif, ce qui est nécessaire pour la validation et l'optimisation quantitative. En microélectronique, dans les technologies standard, le lien entre ces deux niveaux d'abstraction est assez direct. En effet, pour une technologie donnée, il n'y a qu'une seule cellule standard qui est associée à une fonction logique élémentaire. De plus, les niveaux de tensions d'entrée et de sortie sont standardisés, de sorte à ce que toutes les cellules standard soient compatibles au sein d'une même famille logique ou d'une même technologie. La marge de bruit est telle qu'il est très peu probable qu'un '0' logique (respectivement '1') en sortie d'une porte soit interprété comme un '1' logique (respectivement '0') en entrée de la suivante [105]. Le couplage potentiel entre les comportements des différentes portes logiques instanciées dans un circuit est a fortiori très faible sauf dans le cas d'applications très spécifiques.

Ce n'est malheureusement pas le cas en biologie, car tous les signaux biologiques se retrouvent en contact dans le même milieu. De ce fait, chaque instance de fonctions logiques doit être réalisée à l'aide de signaux biologiques différents pour éviter le couplage indésirable entre les différentes portes et l'ensemble des espèces utilisées dans le système. Celles-ci doivent être chimiquement indépendantes les unes des autres de sorte à ne pas perturber le comportement global du système par des réactions inattendues. Par conséquent, il n'y a pas une BioBrique par fonction mais plusieurs, et leurs propriétés changent en fonction des espèces chimiques impliquées, ce qui éloigne la description comportementale de la description quantitative. Un niveau de description intermédiaire semble donc nécessaire.

Le lien entre la description numérique et la description analogique d'une porte logique électronique se fait grâce à des valeurs seuil et des notions de gabarit. Celles-ci sont illustrées Figure 7.1, avec l'exemple d'un gabarit d'une porte NON. Nous pouvons spécifier que toutes les portes NON d'une même famille ou d'une même technologie respectent ce gabarit si les conditions suivantes sont remplies : si la tension d'entrée est supérieure à V_{IH}, la tension de sortie est inférieure à V_{OL} et inversement si la tension d'entrée est inférieure à V_{IL}, la tension de sortie

est supérieure à V_{OH}. Avec ces valeurs données pour une technologie CMOS standard, nous vérifions bien sur la Figure 7.1, les règles de compatibilité ($V_{OH} > V_{IH}$ et $V_{OL} < V_{IL}$), qui assurent qu'une porte possède en entrée le même niveau logique que la précédente en sortie.

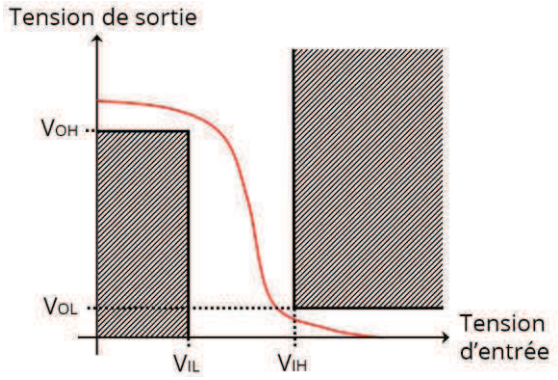


Figure 7.1 : Gabarit d'une porte NON.

Nous avons dans un premier temps exploré une approche équivalente pour la biologie synthétique [106]. La principale difficulté rencontrée est que, pour une même fonction logique, la variété d'espèces biologiques utilisables, et donc la variation des paramètres biochimiques, nous oblige à prendre des marges trop importantes sur les paramètres V_{IL}, V_{IH}, V_{OH} et V_{OL}. De ce fait, la règle de compatibilité énoncée ci-dessus est impossible à vérifier. Ceci implique que l'approche gabarit n'est plus valable et démontre qu'il est impossible de décrire efficacement les propriétés d'une porte biologique avec seulement 2 seuils. L'utilisation de la logique floue (ou fuzzy logic en anglais) [107] pourrait apporter des solutions à ce problème.

La logique floue permet la description d'un système par le biais d'une approche discrète (le comportement du système se résume à un nombre fini de règles), mais permet d'obtenir des résultats quantitatifs, ce qui rend possible la distinction entre deux comportements qui sont a priori identiques, sous une approche purement numérique.

Dans la première partie du chapitre, les principes importants de la logique floue sont rappelés. L'implémentation de l'algorithme est ensuite décrite. Enfin, l'application de l'algorithme est effectuée sur la modélisation de différents mécanismes par le biais de deux exemples concrets, afin d'illustrer l'intérêt de notre travail.

7.1 La logique floue

La logique floue est un domaine des mathématiques dont le concept a été développé par Zadeh en 1965 [107], et qui se rapproche du domaine des probabilités. En logique classique, nous ne considérons que deux états possibles : l'état booléen « vrai » et l'état booléen « faux ». En logique

floue, la plage des données est découpée en plusieurs intervalles et nous considérons qu'une valeur possède un taux d'appartenance à ces différents intervalles.

A partir des bases introduites par Zadeh, la logique floue s'est rapidement répandue et est principalement utilisée pour résoudre des problèmes d'ingénierie de contrôle. Elle est utilisée dans de nombreux domaines tels que l'imagerie et le traitement du signal [108], la gestion du contrôle de la température [109], ainsi que dans des systèmes de contrôle de tension ou de courant [110]. Le principal avantage de la logique floue est qu'elle offre une bonne précision de calcul tout en étant rapide et facile à utiliser.

Le cœur de calcul effectuant les opérations de logique floue est divisé en trois parties principales. Tout d'abord, le but de la première étape, appelée fuzzyfication ou pondération, est de convertir les quantités d'entrée en un ensemble de données adaptées à l'algorithme de la logique floue. Ensuite, les données constituant la sortie du système sont calculées à partir des données d'entrée, selon un ensemble de règles fournies par l'utilisateur. Cette deuxième étape est appelée évaluation des règles. Enfin, la sortie quantitative est calculée à partir des données de sortie pendant la troisième étape : la défuzzyfication ou concrétisation. Ces différentes étapes sont illustrées Figure 7.2.

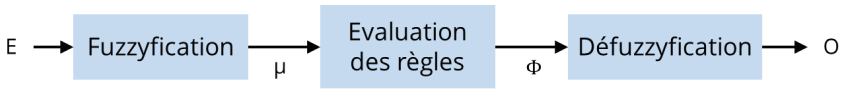


Figure 7.2 : Les trois étapes de la logique floue.

7.1.1 Fuzzyfication

Les entrées-sorties du modèle sont des signaux analogiques mais le cœur du modèle est décrit à l'aide de propositions logiques sur des signaux discrets. La première étape de l'algorithme consiste donc à convertir les signaux d'entrées en signaux discrets. C'est la fuzzyfication. Prenons l'exemple d'un signal analogique d'entrée E borné à un intervalle. Cet intervalle est d'abord divisée en N sous-intervalles, et nous définissons N fonctions d'appartenance (ou en anglais Membership Functions), $MF_1(x)$, ..., $MF_N(x)$, qui indiquent le taux d'appartenance d'une valeur x à chaque intervalle. Les différentes MF_i sont le plus souvent des fonctions triangles, mais d'autres formes existent (exponentielles, gaussienne, ...) [111], [112]. Sur l'exemple de la Figure 7.3, le nombre N de fonctions d'appartenance est fixé à cinq. Le plus souvent, une variable linguistique est attribuée à chaque MF, ce qui permet de définir explicitement l'état auquel la MF correspond. Dans l'exemple Figure 7.3, nous avons 5 variables linguistiques qui sont «très faible», «faible», «moyen», «élevé», et «très élevé».

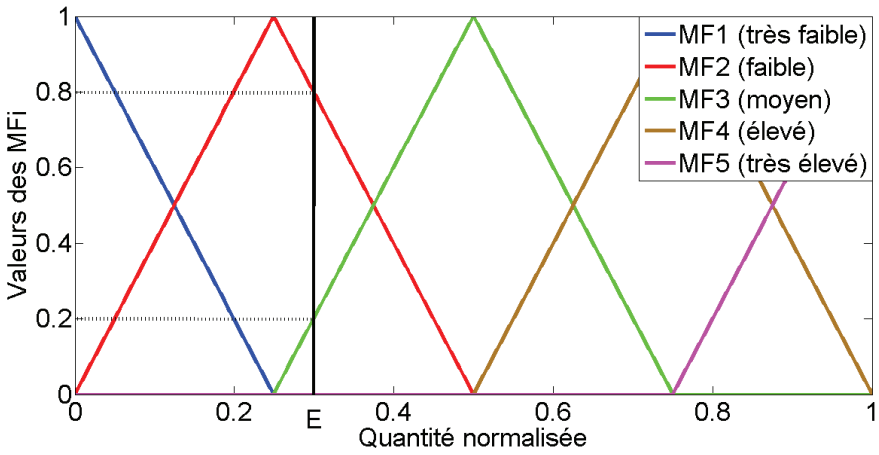


Figure 7.3 : Représentation de l'étape de fuzzyfication avec cinq fonctions d'appartenance et l'entrée E valant 0.3.

La fuzzyfication consiste à donner à l'entrée *E* un degré d'appartenance à chaque MF_i. Ces valeurs constituent un vecteur d'entrée $\mu=(MF_1(E),..., MF_N(E))$. Par exemple, dans le cas illustré Figure 7.3, si la quantité d'entrée *E* vaut 0.3, le vecteur d'entrée vaut μ_E = (0, 0.8, 0.2, 0, 0), ce qui indique que l'entrée appartient à 80% à la MF2 et à 20% à la MF3. Pour un système à *p* variables, la fuzzyfication est appliquée indépendamment sur chacune des variables, avec pour chaque entrée des MF qui peuvent être différentes, et l'ensemble des vecteurs μ_p constitue la structure de données d'entrée qui est ensuite utilisée pour l'évaluation des règles.

7.1.2 Evaluation des règles

L'évaluation des règles consiste à calculer les données de sortie en fonction de la structure des données d'entrée. Cette étape est donc la partie descriptive du modèle. Il s'agit d'une série de propositions logiques, appelée matrice de règles, indiquant à quelle MF chaque sortie appartient, en fonction des MF auxquelles les entrées appartiennent. Par exemple, pour une variable d'entrée E et une variable de sortie O, une règle peut être :

$$Si\ E\ est\ "faible"\ Alors\ O\ est\ "très\ élevé"\qquad (7.1)$$

Avec plusieurs variables, les conditions sur les entrées peuvent être combinées à l'aide de deux opérateurs flous : ET et OU. Notons Ω l'ensemble contenant les *K* règles régissant le fonctionnement d'un système. Pour un même système, Ω peut prendre diverses formes dont deux sont normalisées.

La première consiste à exprimer les règles uniquement avec l'opérateur ET. De ce fait, pour avoir une description exhaustive, l'ensemble des combinaisons d'entrées doit être balayé, ce qui peut conduire à un nombre de règles important : $N_1*...*N_p$ où N_k est le nombre de MF pour l'entrée *k*.

L'autre consiste à exprimer la condition d'appartenance à chaque MF de sortie en fonction des MF d'entrées à l'aide d'une combinaison d'opérateurs ET et OU. Cette alternative réduit considérablement le nombre de règles (1 par MF de sortie), mais complexifie le calcul des taux de validité de chacune des règles. Il s'agit simplement d'une convention d'écriture car d'un point de vue logique, les deux descriptions sont formellement équivalentes.

A titre d'exemple, considérons un système à deux entrées I1 et I2, et, pour chaque entrée et pour la sortie, 3 MF dont les variables linguistiques sont « Faible », « Moyen » et « Elevé ». Dans le premier formalisme, les règles peuvent être représentées par une matrice 3x3. Si le système décrit correspond à une porte logique OU, il peut être décrit par la matrice A sur la Figure 7.4. S'il s'agit d'une porte ET, il peut être décrit par la matrice B. Néanmoins, nous nous apercevons que pour décrire un même comportement logique, nous pouvons définir plusieurs ensembles de règles puisque la matrice C de la Figure 7.4 correspond également à une porte ET plus « tolérante ».

A.

I1\I2	F	M	E
F	F	M	E
M	M	E	E
E	E	E	E

B.

I1\I2	F	M	E
F	F	F	F
M	F	M	M
E	F	M	E

C.

I1\I2	F	M	E
F	F	F	M
M	F	M	E
E	M	E	E

Figure 7.4 : Matrices des règles pour A. une porte OU, B. une porte ET forte et C. une porte ET faible. F correspond à la MF « Faible », M correspond à la MF « Moyen » et E correspond à la MF « Elevé ».

L'évaluation des règles peut se faire selon plusieurs méthodes [111], [112]. Seule la méthode dite « max / min » sera décrite dans ce chapitre. Soit S la sortie du système et σ_j les N_S fonctions d'appartenance définies pour la sortie du système. Pour plus de simplicité, considérons que les règles sont énoncées selon le premier formalisme. Chaque règle est donc constituée de p propositions de type « l'entrée x_j appartient à B_R », dont le taux de validité est noté $\alpha_j = \mu_j(k)$. Ces propositions sont reliées par l'opérateur flou ET. Soit $\alpha_{j,k}$ le taux de validité de la proposition concernant l'entrée x_j pour la $k^{\text{ième}}$ règle. L'algorithme d'évaluation des règles consiste à calculer tour à tour :

- le taux de validité total pour chaque règle : soit β_k le taux de validité total de la règle k. β_k est défini comme étant la valeur minimale des taux de validité des propositions qui la compose :

$$\beta_k = \min_j \alpha_{j,k} \tag{7.2}$$

- le taux de réalisation de chaque fonction d'appartenance en sortie : soit φ_i ce taux de réalisation pour la $i^{ème}$ fonction d'appartenance, et E_i l'ensemble des numéros de règles

conduisant à une sortie égale à la $i^{ème}$ fonction d'appartenance. μ_i est défini comme la valeur maximale des taux de validité des règles appartenant à E_i :

$$\varphi_i = \max_{t \in E_i} \beta_t \tag{7.3}$$

- La fonction de sortie définie comme le maximum du minimum entre les fonctions d'appartenance et le taux de réalisation :

$$s(x) = \max_t \big(min(\sigma_t(x), \varphi_t)\big) \tag{7.4}$$

Prenons l'exemple des matrices de règles de la figure 7.2. On fixe I1 = 0.7 et I2 = 0.1. Après fuzzyfication avec 3 fonctions d'appartenance triangulaires dans l'intervalle 0 à 1, nous obtenons les vecteurs d'entrée : μ_1 = (0, 0.6, 0.4) et μ_2 = (0.8, 0.2, 0). Les taux de validité des règles, représentés sous forme matricielles, afin de correspondre à la matrice de règles, sont :

$$\begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \\ \beta_4 & \beta_5 & \beta_6 \\ \beta_7 & \beta_8 & \beta_9 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0.6 & 0.2 & 0 \\ 0.4 & 0.2 & 0 \end{bmatrix} \tag{7.5}$$

Par exemple, $\beta_1 = \min(\alpha_{1,1}, \alpha_{2,1}) = \min(0{,}0.8) = 0$. Nous supposons que la sortie est également fuzzyfiée avec 3 fonctions d'appartenance triangulaires dans l'intervalle 0 à 1. Les taux de réalisation de chacune des trois fonctions d'appartenance Φ_A, Φ_B et Φ_C pour les 3 matrices de règles de la Figure 7.4 sont :

$$\begin{aligned} \Phi_A &= [0 \quad 0.6 \quad 0.4] \\ \Phi_B &= [0.6 \quad 0.2 \quad 0] \\ \Phi_C &= [0.6 \quad 0.4 \quad 0.2] \end{aligned} \tag{7.6}$$

Pour A : $\Phi_A = [\varphi_{A1} \quad \varphi_{A2} \quad \varphi_{A3}] = [\max(\beta_1) \quad \max(\beta_2, \beta_4) \quad \max(\beta_3, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9)] = [0 \quad 0.6 \quad 0.4]$. Finalement, les fonctions de sortie pour les trois cas de figure sont représentées par la Figure 7.5.

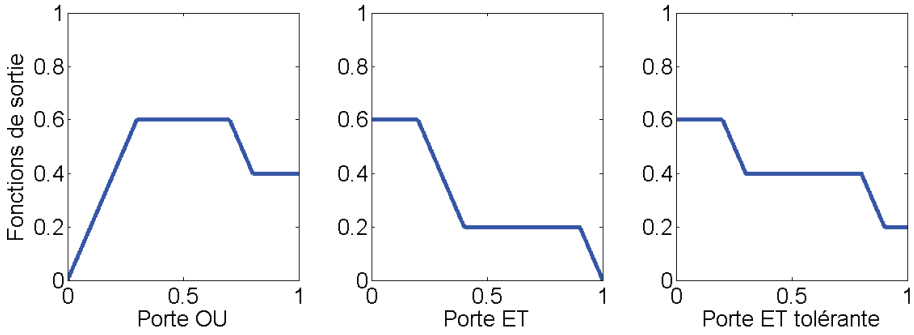


Figure 7.5 : Fonctions de sortie pour les 3 matrices de règles de la Figure 7.4.

Une autre méthode de calcul classique, appelée « somme / produit », est basée sur le même algorithme. La différence se situe au niveau du taux de validité d'une règle, qui est calculé par le

produit entre les taux de validité des propositions. Le taux de réalisation d'une fonction d'appartenance de sortie est calculé comme la somme des taux de réalisation des règles.

7.1.3 Défuzzyfication

La défuzzyfication est la dernière étape qui permet de quantifier la sortie S du modèle à partir de la fonction de sortie, calculée précédemment. S est le plus souvent calculée par intégration numérique, par dichotomie ou directement à partir des résultats de l'évaluation des règles [111], mais la méthode la plus commune est celle du centroïde, aussi appelée centre de gravité. Mathématiquement, S est simplement définie comme le centre de gravité de la fonction de sortie $s(x)$:

$$S = \frac{\int_0^1 x \cdot s(x) \cdot dx}{\int_0^1 s(x) \cdot dx} \tag{7.7}$$

Géométriquement, elle correspond donc à la valeur de l'abscisse pour laquelle les aires sous la fonction de sortie de part et d'autre de S sont égales. Ceci est illustré Figure 7.6, où la barre noire représente le centroïde, qui sépare l'aire sous la fonction de sortie en deux aires égales, l'aire bleue et l'aire verte. Cette méthode est appliquée sur l'exemple précédent et les résultats pour les trois matrices de règles proposées sur la Figure 7.4 sont : S_A = 0.54, S_B = 0.348 et S_C = 0.424.

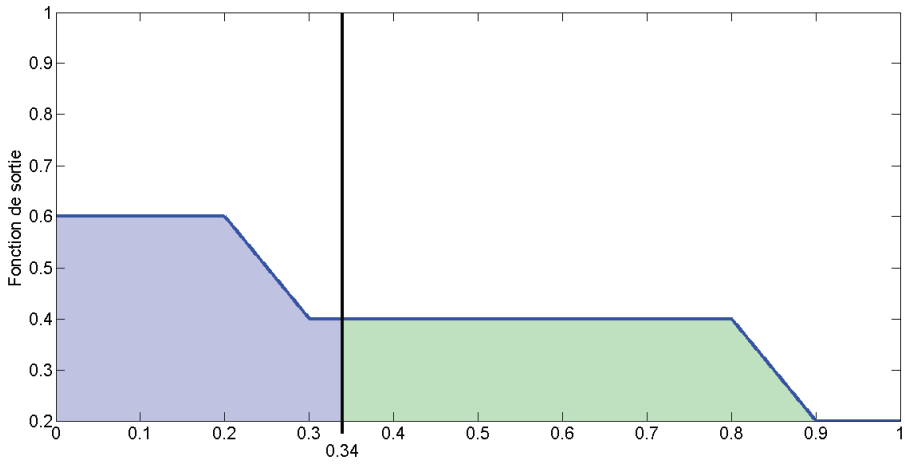


Figure 7.6 : Illustration du calcul du centroïde, ici valant 0.34, séparant l'aire sous la fonction de sortie en deux aires égales, l'aire bleue et l'aire verte.

7.2 Implémentation

Dans la littérature, plusieurs ouvrages traitent des différentes méthodes employées pour implémenter les algorithmes des étapes utilisées en logique floue [112], [113], [114]. Parmi les

implémentations, nous retrouvons la boîte à outils Fuzzy de Matlab [115] qui est très complète et permet de visualiser graphiquement ses différentes étapes. Elle permet également de modifier dynamiquement les règles, leurs méthodes d'évaluation, et les algorithmes de défuzzyfication. D'autres implémentations ont été réalisées dans les langages de programmation les plus courants, comme Java [116], Python [117], C++ [118], tout comme dans les langages de description matériel comme le VHDL [119].

Ces deux dernières implémentations sont particulièrement intéressantes. La première étant une librairie en C++, elle pourrait être instanciée dans notre bibliothèque de modèles en SystemC. La deuxième est réalisée directement dans un des langages de modélisation utilisé dans le flot de conception standard de la microélectronique, donc son intégration en est encore plus aisée. Ce travail n'a pas été mené dans le cadre de la thèse, car en plus de l'intégration, un certain nombre d'adaptations devrait être apporté pour rendre les simulateurs compatibles avec les contraintes de la biologie et la finalité que nous souhaitons. Cela implique donc un travail d'ingénierie logicielle important, par rapport auquel nous avons privilégié une étude plus approfondie de la faisabilité et de l'intérêt de l'utilisation de modèles en logique floue pour la biologie. Pour mener cette étude, un cœur de calcul dédié aux applications biologiques a d'abord été développé sous Matlab [120]. Les paragraphes suivants permettent de discuter des choix qui ont été fait pour le développement de ce cœur, le code complet étant disponible en Annexe H.

Les différentes implémentations acceptent la plupart du temps une plage de données d'entrée dont les bornes peuvent être librement fixées par l'utilisateur. Cependant, elles ne permettent qu'une évolution linaire entre ces bornes. Or les concentrations expérimentales évoluent souvent de manière logarithmique. Dans notre algorithme, nous avons le choix des bornes ainsi que du type d'échelle disponible pour la définition des MF. Nous travaillerons la plupart du temps en échelle logarithmique, en ayant normalisé les entrées par une valeur de concentration maximale. Ceci permet une définition plus efficace des règles mais le recours au logarithme reste transparent du point de vue de l'utilisateur.

Nous avons choisi de représenter les fonctions MFs par des fonctions triangles, qui sont plus faciles à intégrer dans l'algorithme de calcul car elles comportent des transitions linéaires. Nous obtenons donc de meilleures performances en matière de vitesse de calcul.

Dans les implémentations classiques, les opérations de la logique floue sont réalisées sur un ensemble de points couvrant l'intervalle des valeurs d'entrée. Plus ce nombre de points est élevé, plus l'algorithme de logique floue donne des résultats précis. Cependant, les opérations de calcul peuvent être simplifiées, pour ne garder que les points intéressants, correspondant aux intersections des fonctions d'appartenance. Nous avons ainsi repensé toute la méthodologie de

calcul afin de l'alléger. Les résultats sont obtenus environ dix fois plus rapidement avec cette technique, présentée en Annexe H, par rapport à la méthodologie classique.

Pour la défuzzyfication, plusieurs méthodes existent mais le résultat est généralement calculé par la méthode du centroïde, présentée équation (7.7). Nous avons adapté cette fonction à la méthodologie de calcul employée, afin de déterminer avec précision l'abscisse correspondant à l'endroit séparant en deux l'aire sous la fonction de sortie. Malgré le peu de points disponibles, la méthode de calcul du centroïde implémentée, présentée en Annexe H, donne une précision similaire à la fonction de calcul traditionnelle sur beaucoup de points.

Néanmoins, cette nouvelle méthode, ainsi que celle du centroïde, présentent des défauts pour les bornes inférieure et supérieure. La valeur de sortie ne varie pas entre 0 et 1, en valeur normalisée, mais entre l'abscisse correspondant à la moitié de l'aire du triangle de la première MF et l'abscisse correspondant à la moitié de l'aire du triangle de la dernière MFs. En temps normal, cela n'est pas gênant, mais pour l'utilisation que nous voulons faire des modèles en logique floue, il sera nécessaire d'interconnecter plusieurs modèles entre eux. Nous risquons donc d'avoir, au fur et à mesure des étages de modèles, des plages de variation de plus en plus resserrées autour de la valeur médiane, ce qui n'est pas favorable. Pour corriger ce problème, nous avons mis en place une méthode qui permet de renormaliser les résultats correctement entre des valeurs inférieure et supérieure, données par l'utilisateur.

7.3 Application à la biologie

La logique floue est de plus en plus utilisée dans le domaine de la bio-informatique, comme pour l'analyse de certaines données, ainsi que pour étudier le comportement de certaines espèces [121], [122]. C'est toutefois une tout autre application qui est visée ici puisqu'il s'agit de la modélisation de mécanismes biologiques. Plus exactement, le but est de se servir au mieux du compromis qu'offre la logique floue, entre une description comportementale pouvant être résumée à un nombre fini de propositions et des modèles permettant d'obtenir des résultats quantitatifs. Un des rares groupes ayant travaillé sur cette application est celui de Woolf et Wang [108], qui a posé les bases d'une modélisation en logique floue de la régulation génétique. Ils se sont servis de leur algorithme pour déterminer les triplets activateurs/répresseur/protéine, à partir de données expérimentales d'expression génétique chez les levures. Leur algorithme a ensuite été amélioré par H. Ressom *et al.* [123], qui ont pu réduire le temps de calcul par deux.

Nous allons maintenant présenter deux de nos travaux sur des applications biologiques de la logique floue. Le premier a pour objectif de montrer que nous pouvons modéliser efficacement des phénomènes biologiques et obtenir des résultats de simulation pertinents pour un temps de

calcul réduit. Le second est une illustration du potentiel de la logique floue pour des opérations automatisées de conception.

7.3.1 Modélisation d'un biosystème de la littérature

Pour ce premier exemple, nous reprenons le biosystème synthétique conçu par Xie *et al.* [101] présenté dans le chapitre 6. D'un point de vue numérique, nous pouvons constater que la sortie DsRed du système peut être exprimée par l'équation logique simplifiée suivante :

$$DsRed = \overline{\overline{microARN}} \tag{7.8}$$

L'action des microARNs sur rtTA est donc redondante avec leur action sur LacI et n'apparaît pas dans l'équation (7.8). Leur action répressive sur rtTA n'est donc pas nécessaire d'un point de vue numérique.

Cependant, les expériences montrent que DsRed n'est pas synthétisé en présence des microARNs, quand cette action sur rtTA n'est pas présente. Cette propriété ne peut donc pas être prise en considération avec le modèle numérique, mais peut être intégrée grâce à la logique floue. La Figure 7.7 illustre les résultats obtenus pour la simulation du modèle en logique floue, comparés aux résultats expérimentaux de la publication de Xie *et al.* [101].

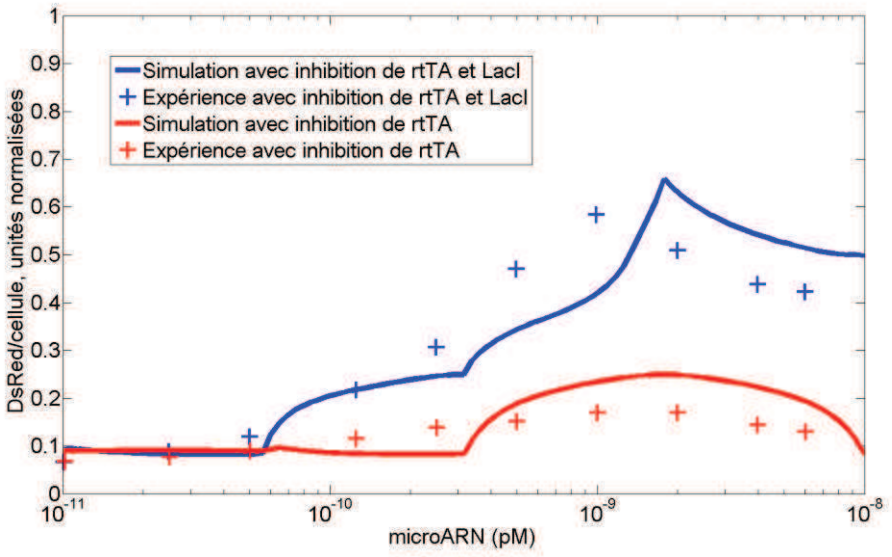


Figure 7.7 : Les résultats de simulation du modèle en logique floue comparés aux résultats expérimentaux de Xie et al. [101].

Nous retrouvons les résultats expérimentaux représentés par des croix, et les résultats de simulation du modèle en logique floue en traits pleins. Deux systèmes sont présentés, le premier

en bleu, où LacI et rtTA sont sensibles tous les deux aux microARNs, et le deuxième en rouge, où seulement LacI est sensible aux microARNs.

Les modèles en logiques floues possèdent cinq MFs pour chaque entrée et sont composés de trois éléments : la synthèse de rtTA, de LacI et de DsRed. La modélisation à l'aide de la logique floue permet de décrire la diminution de la synthèse DsRed à partir d'une concentration des microARNs de 10^{-9} pM, ce qui est impossible avec le modèle numérique car cela nécessiterait de modifier les modèles standards utilisés pour les descriptions analogiques. L'ajustement du modèle avec les données expérimentales pourrait encore être amélioré en augmentant le nombre de MFs pour chaque entrée, mais cela conduirait à des temps de simulation supérieurs, ce qui n'est pas le but de ce modèle. Un compromis entre rapidité de simulation et précision du modèle peut être trouvé en choisissant le bon nombre de MFs par entrée. Ici, le système possède deux entrées, donc la complexité de l'algorithme va évoluer en N^2, où N est le nombre de MFs. La logique floue ayant pour but de fournir le comportement quantifié d'un système et non de remplacer une simulation bas niveau, le choix de N=5 semble être le meilleur compromis précision/rapidité.

Le système entier est ensuite modélisé en conservant cette précision de cinq MFs. Nous avons effectué une analyse statistique du comportement du système. Pour cela, 1000 combinaisons aléatoires des valeurs des concentrations des différents microARNs (miR-21, miR-17 ou 30a et miR-141 ou 142 ou 146a) ont été générées, avec une distribution de probabilité uniforme sur des valeurs normalisées allant de 0 à 1. Chaque échantillon est représenté sur le graphique 3D de la figure 7.8.A. Les croix bleues correspondent à la combinaison des concentrations des microARNs des cellules considérées comme saines, alors que les croix rouges correspondent à la combinaison des concentrations des microARNs des cellules considérées comme cancéreuses. Le seuil de détection, permettant de déterminer si une cellule est cancéreuse, a été fixé à la valeur normalisée de 0,3 pour la concentration de DsRed, afin d'être cohérent avec les données expérimentales publiées. La zone hachurée correspond à la zone où les modèles analogiques prédisent des cellules saines. Il y a quelques faux positifs mais la détection avec le modèle en logique floue est efficace à plus de 99%. Les résultats de simulation sont obtenus presque instantanément avec le modèle flou, tandis que le temps de calcul est beaucoup plus important avec le modèle bas niveau, dont la simulation dure environ 10 secondes pour chaque cas de figure.

Une autre étude statistique a été réalisée sur ces 1000 échantillons, en comparant les résultats du modèle en logique floue à cinq MFs à ceux du modèle en logique floue à deux MFs, sensés tendre vers le comportement numérique. La répartition des concentrations de DsRed calculées ainsi que le pourcentage de cellules considérées comme cancéreuses pour ces deux modèles

sont illustrés Figure 7.8. Nous constatons bien que le modèle tendant vers un comportement numérique présente des résultats erronés par rapport à celui à cinq MFs, ce qui montre l'intérêt de la logique floue par rapport à une simple abstraction numérique.

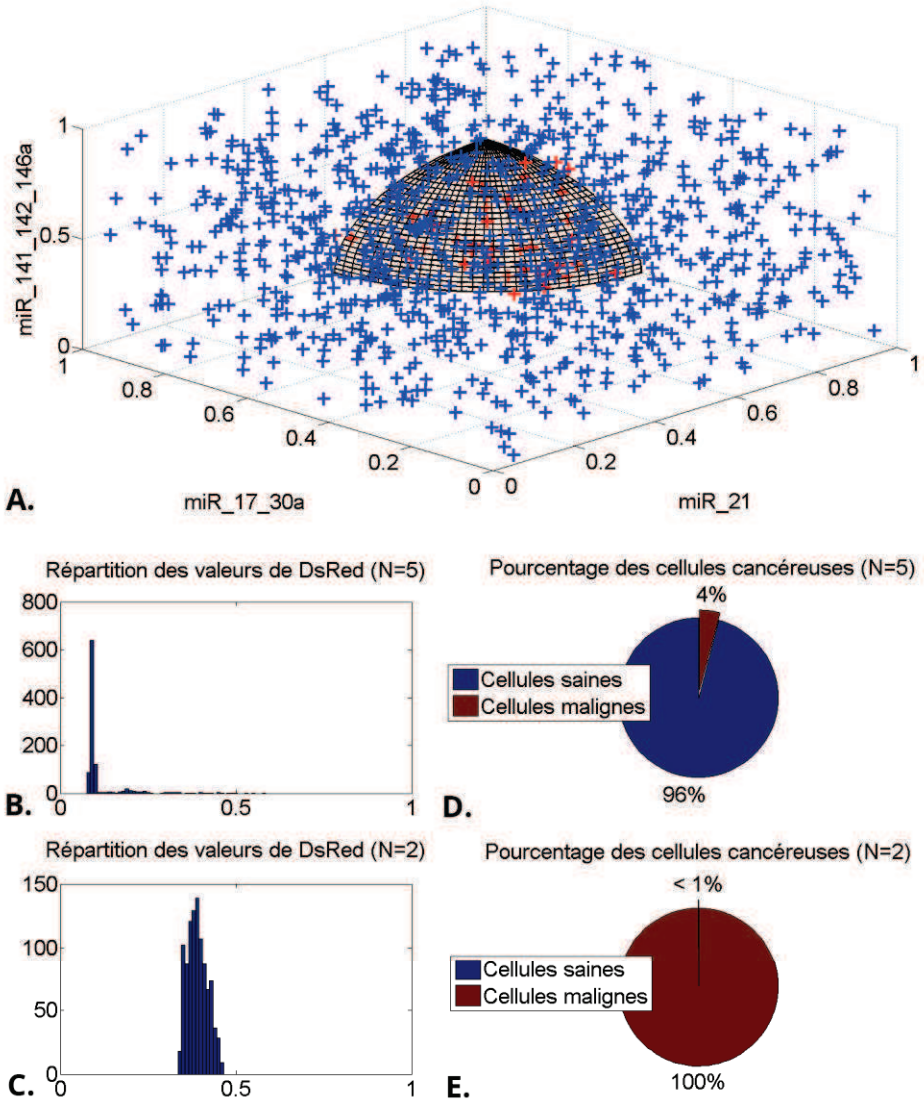


Figure 7.8 : A. Simulation statistique sur 1000 échantillons de cellules avec des concentrations des différents microARNs tirées aléatoirement. B. et C. Histogrammes de la répartition des concentrations de DsRed calculées, pour un modèle avec cinq et deux MFs respectivement. D. et E. Pourcentages des cellules considérées comme saines et cancéreuses avec un seuil de détection à 0.3 pour DsRed, pour les modèles avec cinq et deux MFs respectivement.

7.3.2 Conception d'une porte OU exclusif biologique

Le deuxième exemple présenté est celui d'une porte OU exclusif biologique, illustrée Figure 7.9. Elle correspond à une version abstraite de la porte OU exclusif, imaginée par une équipe de l'ETH

Zürich dans le cadre du concours iGEM 2006 [124]. Elle permet de montrer l'intérêt que peut avoir la modélisation à l'aide de la logique floue dans le processus de conception. Cette porte OU exclusif comprend trois gènes et sept protéines. L'entrée *Astim* régule l'activité du premier gène (*Bstim* le deuxième) et par conséquent la quantité de *A* et *B** (respectivement de *B* et *A**) synthétisés dans la cellule. Ensuite, *A* et *A** (respectivement *B* et *B**) peuvent se lier pour former un complexe moléculaire *AA** (respectivement *BB**). Enfin, un troisième gène code pour la synthèse d'un reporteur fluorescent, la GFP. Ce gène possède deux activateurs : *A* et *B*. Trois cas de figure peuvent se présenter : les deux entrées *Astim* et *Bstim* sont absentes, *Astim* ou *Bstim* sont présents seuls, ou les deux entrées sont présentes en même temps. Dans le premier cas, la GFP n'est pas synthétisée car ni *A* ni *B* ne sont produits. Dans le deuxième cas, la GFP est bien synthétisée car soit *A* soit *B* est produit. Dans le troisième cas, *A* et *B* sont tous les deux produits, mais leurs ligands *A** et *B** aussi. Une certaine quantité de *A* et *B* est donc immédiatement consommée pour former les complexes *AA** et *BB** et il n'y en a plus suffisamment pour activer l'expression du gène de sortie codant pour la GFP. Ce comportement correspond bien à celui d'une porte OU exclusif, n'étant active qu'en présence d'une des deux entrées.

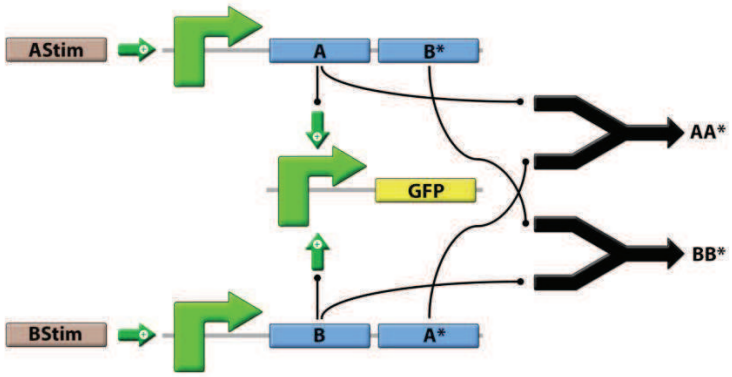


Figure 7.9 : Porte OU exclusif fictive composée de 3 gènes et 7 protéines.

Ce système a été modélisé avec la logique floue en utilisant, pour chaque signal, cinq MFs. Il est constitué de cinq blocs comme le montre la Figure 7.10.

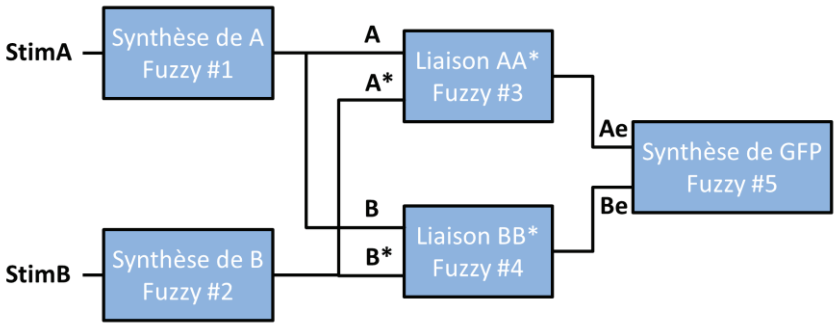
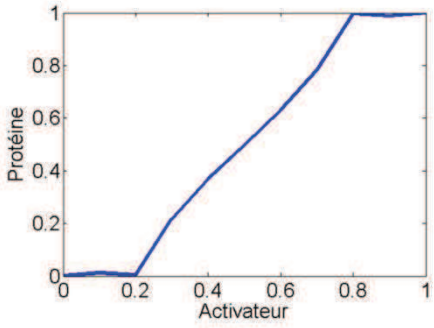


Figure 7.10 : Schéma des blocs de logique floue correspondant à la porte OU exclusif Figure 7.9.

Les blocs #1 et #2 correspondent à des règles d'un mécanisme de synthèse de protéines. La matrice de règles ainsi que les résultats de simulation correspondant à ces blocs sont donnés Figure 7.11.

Activateur	TF	F	M	E	TE
Protéine	TF	TF	M	TE	TE

A. Matrice de règles pour la synthèse



D. Protéine synthétisée en fonction de l'activateur

Figure 7.11 : Matrice de règle (A.) et résultats de simulation correspondant (B.), pour le mécanisme de synthèse de blocs #1 et #2.

Les blocs #3 et #4 correspondent aux règles d'une réaction de liaison. Pour ces deux blocs, deux règles différentes donnant *Xe*, la quantité effective de *X* après la réaction de liaison, ont été comparées. Celles-ci sont données Figure 7.12.A et 7.12.B. La matrice de la Figure 7.12.A correspond à une réaction de liaison presque complète consommant la totalité des réactifs, tandis que la Figure 7.12.B correspond à une réaction de liaison partielle (tous les réactifs ne sont pas consommés). De la même façon, le bloc #5, qui correspond aux règles de l'activation du gène synthétisant GFP est également testé avec deux matrices de règles différentes. La première correspond à un mécanisme avec un activateur faible, donnée Figure 7.12.C (une quantité importante d'activateur est nécessaire pour obtenir beaucoup de protéine synthétisée) et la deuxième à un mécanisme avec un activateur fort, donnée Figure 7.12.D.

Le système est ensuite simulé pour plusieurs cas énumérés dans le tableau de la Figure 7.12.E, correspondant à différents choix pour les mécanismes de liaison des blocs #3 et #4 et pour les mécanismes de synthèse du bloc #5. Nous constatons que le comportement est très différent selon les matrices de règles sélectionnées. L'étude montre que la seule combinaison qui semble présenter un comportement de porte OU exclusif est celle qui utilise une réaction de liaison presque complète pour les blocs #3 et #4, et un activateur faible pour la synthèse du bloc #5.

Cette conclusion est indispensable au concepteur dans la phase de synthèse biologique où il devra sélectionner des protéines réelles pour *A*, *A**, *B* et *B**. Il sait qu'il faudra trouver des protéines *A* et *A** (respectivement *B* et *B**) possédant une constante d'association forte. Les espèces *A* et *B* devront de plus correspondre à des activateurs « faibles ». Le modèle numérique des mécanismes mis en jeu n'aurait pas été assez détaillé pour permettre d'établir cette

conclusion en amont. Il aurait donc fallu choisir des paramètres standards pour les mécanismes biologiques utilisés, puis effectuer une série de simulations à l'aide du modèle analogique pour converger de manière itérative vers des conditions sur les paramètres du modèle (constantes des réactions de complexation, paramètres des équations de Hill) permettant de retrouver le comportement d'une porte OU exclusif. Ces conditions auraient mené à la même conclusion que celles obtenues avec le modèle en logique flou en 4 simulations.

A. Réaction de liaison complète (LC)

X*\X	TF	F	M	E	TE
TF	TF	F	M	E	TE
F	TF	TF	F	M	E
M	TF	TF	F	F	M
E	TF	TF	TF	F	F
TE	TF	TF	TF	TF	F

B. Réaction de liaison partielle (LP)

X*\X	TF	F	M	E	TE
TF	TF	F	M	E	TE
F	TF	F	M	E	TE
M	TF	TF	F	M	E
E	TF	TF	F	M	E
TE	TF	TF	TF	F	M

C. Synthèse avec un activateur faible (fa)

B\A	TF	F	M	E	TE
TF	TF	TF	TF	E	TE
F	TF	TF	F	E	TE
M	TF	F	M	E	TE
E	E	E	E	TE	TE
TE	TE	TE	TE	TE	TE

D. Synthèse avec un activateur fort (FA)

B\A	TF	F	M	E	TE
TF	TF	F	M	E	TE
F	F	M	E	TE	TE
M	M	E	TE	TE	TE
E	E	TE	TE	TE	TE
TE	TE	TE	TE	TE	TE

E.

		Cas #1	Cas #2	Cas #3	Cas #4
Blocs Fuzzy #3 et #4		LC	LP	LC	LP
Bloc Fuzzy #5		fa	fa	FA	FA
StimA	StimB	GFP			
0	0	0	0	0	0
1	0	1	1	1	1
0	1	1	1	1	1
1	1	0.003	0.496	0.430	1

Figure 7.12 : A. et B. sont deux matrices de règles possibles pour la réaction de liaison des blocs #3 et #4 de la Figure 7.10, C. et D. sont les matrices du mécanisme de synthèse du bloc #5, et E. présente les résultats de simulation obtenus avec les différents choix pour ces mécanismes (TF = très faible, F = faible, M = moyen, E = élevé, TE = très élevé).

7.4 Conclusion

Après avoir présenté les bases de la logique floue ainsi que son implémentation, ce chapitre montre qu'elle est une alternative intéressante pour la modélisation des systèmes biologiques à un niveau d'abstraction intermédiaire entre le comportement numérique et analogique. La logique floue est très simple à utiliser, ce qui facilite sa mise en œuvre et rend la simulation de modèles beaucoup plus rapide que dans le cas des équations complexes. Malgré cette rapidité de simulation, la logique floue permet d'obtenir des résultats quantifiables au lieu d'une valeur

tout-ou-rien comme pour les modèles numériques, ce qui la rend beaucoup plus précise que ces derniers. Le premier exemple présenté montre que les modèles en logique floue peuvent être utilisés pour décrire des mécanismes biologiques complexes tels que des systèmes les plus récemment développés.

L'exemple de la porte OU exclusif met aussi en évidence l'intérêt de ce niveau de modélisation pour les phases de conception d'un biosystème. En introduction, nous avions vu que le comportement de deux BioBriques, reposant sur le même mécanisme biologique mais utilisant des espèces différentes, n'est pas forcément identique. Une classification des BioBriques uniquement à partir de leur fonction numérique est donc trop limitée et l'utilisation de paramètres issus des modèles bas-niveau n'est pas possible lors des étapes de conception. Cela revient à faire de la conception analogique, ce qui, malgré de nombreux efforts, a montré ses limites [85], [86], [87]. En revanche, la classification des BioBriques à l'aide de la logique floue permet de faire un compromis entre abstraction comportementale et bas-niveau. Nous gardons un nombre fini d'instances pour une fonction donnée mais celles-ci décrivent les BioBriques assez précisément pour qu'elles puissent être discriminées.

Ainsi, un outil de synthèse automatique pourra se baser sur cette classification afin de guider et d'optimiser le choix des BioBriques utilisées pour effectuer une fonction logique ciblée. L'intégration de ces modèles et du cœur de calcul dans le flot de conception présenté chapitre 4 est en cours d'étude. Son intégration dans la partie amont des étapes de conception est en revanche beaucoup plus complexe et fera l'objet de travaux ultérieurs.

Troisième partie

Amélioration des modèles et modélisation avancée

Chapitre 8

Modélisation avancée de la liaison entre une macromolécule et des ligands

Les modèles bas-niveau présentés jusque-là utilisent des équations mathématiques empiriques de premier ordre. Parallèlement au travail de formalisation des modèles destinés au flot de conception, nous avons également mené des travaux sur l'amélioration des modèles eux-mêmes. Une des premières améliorations qui peut être apportée à la description des mécanismes biologiques est une modélisation plus fine des interactions entre les macromolécules et les ligands. Jusqu'à présent, ces mécanismes ont été décrits à l'aide de modèles mathématiques simples, construits autour de l'équation de Hill, décrivant le taux moyen d'occupation des sites de liaison de la macromolécule en fonction de la quantité de ligand. Son principal avantage est de n'être constituée que de deux paramètres par type de ligand, qui peuvent donc être facilement extraits de courbes expérimentales. Il a toutefois été démontré que son utilisation présentait un certain nombre de limitations et ne permettait pas de couvrir efficacement l'ensemble des interactions intervenant dans un système macromolécules-ligands [125]. Ce type de problématique est à la base de tous les mécanismes biologiques utilisés pour décrire des biosystèmes existants ou synthétiser des biosystèmes artificiels. Nous la retrouvons dans le cas de la complexation entre plusieurs molécules, mais aussi dans le cas de l'expression des gènes, où la liaison des activateurs et des répresseurs sur l'ADN est étudiée, en présence ou en absence de l'ARN polymérase, l'enzyme qui catalyse le négatif de l'ADN en ARN. Il est donc nécessaire d'étudier les modèles permettant de décrire ces interactions de manière plus complète et fidèle.

La modélisation avancée de l'interaction entre une protéine et un ligand s'est construite historiquement sur l'analogie entre une clef (le ligand) et une serrure (la macromolécule) [126]. Les conformations du ligand et de la serrure sont rigides et ne peuvent s'emboiter que lorsque la conformation de la clef est rigoureusement complémentaire à la conformation de la serrure. Ce modèle ne permet pas d'expliciter la plasticité des systèmes biologiques, où la conformation de la macromolécule est modifiée par la complexation du ligand. Ce phénomène, décrit sous le nom d'allostérie (autre forme) a nécessité le développement de nouveaux modèles. Deux grandes classes de modèles sont alors apparues dans les années 50 :

- Le modèle de Wymann-Monod-Changeux [127], ou modèle allostérique, où l'on considère que la macromolécule existe sous forme d'un équilibre entre au moins deux conformations différentes, l'une de ces conformations étant le complémentaire de la conformation du ligand.
- Le modèle d'ajustement induit [128], ou modèle de Koshland où l'on considère que la liaison du ligand entraine la modification conformationnelle de la macromolécule.

Quelle que soit l'approche, l'étude d'une interaction ligand-macromolécule repose sur un principe itératif entre modèles et expériences, illustré Figure 8.1.

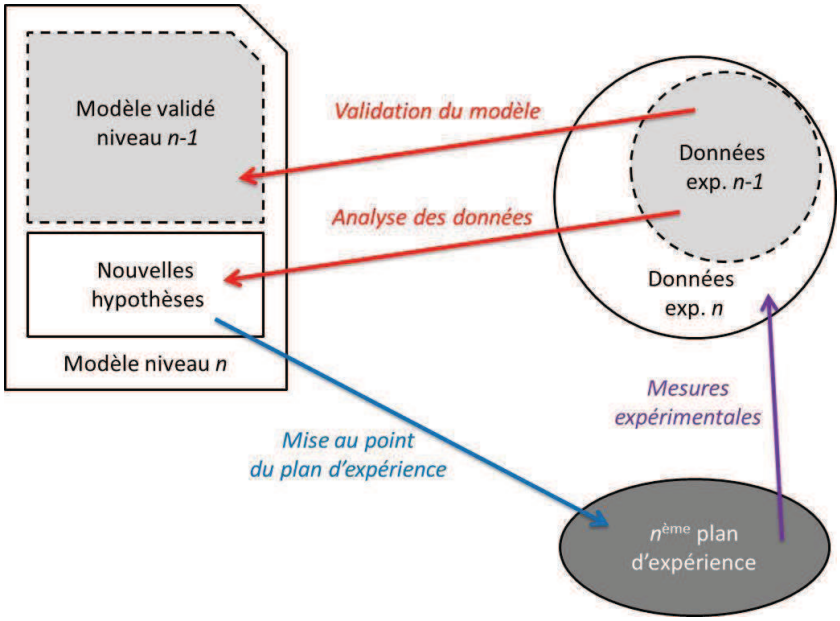


Figure 8.1 : Principe itératif entre modèles et expériences.

Le point de départ est un modèle simple, ou modèle de niveau 0, auquel nous associons une première série de mesures expérimentales. Cette première série de mesures expérimentales permet d'extraire un ensemble de paramètres de ce modèle et de mettre en avant les limites de celui-ci. Nous tentons donc ensuite d'améliorer le modèle (modèle au niveau 1) et nous procédons de la même manière. Le passage du modèle de niveau $n-1$ au modèle de niveau n se fait en trois étapes élémentaires :

- La mise en place de nouvelles hypothèses, plus fines, sur le fonctionnement réel de l'interaction (mécanisme de changement de conformation, ordre des liaisons, conditions environnementales) à partir des données expérimentales recueillies au niveau $n-1$.
- La construction du modèle mathématique au niveau n, à partir du modèle au niveau $n-1$ et les nouvelles hypothèses.

- La mise au point d'un plan d'expérience, permettant de valider ces hypothèses, ainsi que le modèle associé et d'en extraire les paramètres.

Nous reproduisons ce mécanisme de manière itérative jusqu'à obtenir un modèle satisfaisant l'ensemble des données expérimentales produites. Les méthodes expérimentales permettant d'obtenir des résultats de mesures sont de trois types :

- Les méthodes permettant de suivre la fixation du ligand de manière directe, par exemple la résonance plasmonique de surface et, dans certains cas précis, la spectrométrie de masse supramoléculaire pour mesurer les modifications de masse des molécules [129], [130], [131], [132].
- Les méthodes permettant de suivre les changements de conformations de la macromolécules, ou dans certains cas du ligand, par exemple les spectroscopies de fluorescence, la spectroscopie RMN, ou encore la microcalorimétrie pour suivre l'évolution des conformation des macromolécules [133], [134], [135].
- Les méthodes permettant de suivre l'activité induite par la fixation du ligand, ou les modifications cellulaires qui peuvent en découler, par exemple les mesures de l'activité enzymatique, d'une modification de la morphologie cellulaire ou de la production d'un signal intermédiaire (chacun de ces effets pouvant être suivi par des techniques *ad hoc*) pour l'observation des effets biologiques d'une interaction [136].

L'approche de modélisation décrite sur la Figure 8.1 entraine irrémédiablement des modèles spécifiques pour chaque interaction, dépendant du type d'interaction, de l'effet biologique observable, des moyens expérimentaux mis en œuvre, et surtout des besoins et/ou des habitudes des équipes mettant au point les modèles. Cette diversité des modèles va à l'encontre des idées de standardisation et de généricité que nous défendons dans ce travail de thèse et qui pourraient permettre la structuration des outils de conception en biologie synthétique. Néanmoins, nous allons démontrer dans ce chapitre qu'il est possible d'unifier les modèles que l'on peut trouver dans la littérature sous un seul et unique « modèle générique », terme qui sera utilisé dans toute la suite pour désigner notre approche.

Nous commençons par présenter cette méthodologie, en traitant la liaison d'un seul type de ligand à une macromolécule avant de la généraliser à plusieurs types de ligands pouvant se lier sur plusieurs sites. Nous faisons ensuite les liens entre ce modèle générique et les autres descriptions classiques que nous pouvons trouver dans la littérature. Dans la troisième partie, nous présentons un logiciel développé sous Matlab permettant la génération automatique des équations du modèle. Enfin, nous illustrons ces travaux par l'exemple de la liaison des ions calcium à la calmoduline.

8.1 Modélisation de l'interaction ligand-macromolécule biologique

Dans cette première section nous présentons la méthodologie conduisant au modèle générique. Contrairement à tous les modèles évoqués dans les chapitres précédents, ce nouveau modèle est basé sur une approche microscopique qui est explicitée dans la première section de ce chapitre.

8.1.1 Approche microscopique

Considérons une macromolécule P, constituée de n sites susceptibles de lier n ligands L (par exemple sur la Figure 8.2, $n = 3$). Nous pouvons considérer que dans ce cas de figure, il n'existe que quatre espèces possibles : P (la macromolécule non liée), PL (la macromolécule avec un ligand quel que soit le site qu'il occupe), PL_2 (la macromolécule avec 2 des 3 sites occupés) et PL_3 (la macromolécule avec 3 sites occupés). Il s'agit de l'approche macroscopique et les équilibres entre ces quatre espèces sont définis à partir de constantes d'association, appelées aussi constantes d'Adair-Klotz, définies en début de partie 8.1.2.

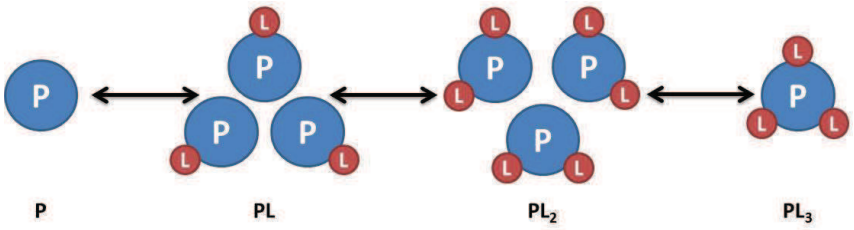


Figure 8.2 : Approche macroscopique de la liaison de trois ligands L sur une macromolécule P.

A l'inverse, dans l'approche microscopique (Figure 8.3), nous considérons que les propriétés de l'espèce chimique issue d'une liaison macromolécule-ligands ne dépendent pas que du nombre de sites occupés, mais aussi des numéros des sites occupés. Ainsi, nous considérons qu'il existe trois espèces PL différentes pouvant avoir potentiellement des propriétés différentes. Nous les notons $PL_1L_0L_0$, $PL_0L_1L_0$ et $PL_0L_0L_1$, en fonction du site occupé par le ligand (site n°1, n°2 ou n°3 respectivement). Ainsi, dans l'approche microscopique, nous distinguons 2^n espèces différentes, soit 12 constantes d'association dans le cas $n = 3$, au lieu de seulement 3 constantes dans l'approche macroscopique. Dans la partie suivante, nous montrerons que ces constantes peuvent être remplacées par un jeu constitué de deux types de paramètres : la constante d'association propre à un site (correspondant à l'affinité d'un site à fixer un ligand) ainsi que des coefficients de couplage entre les sites (modifiant l'affinité d'un site à fixer un ligand lorsqu'un ou plusieurs autres sites sont occupés).

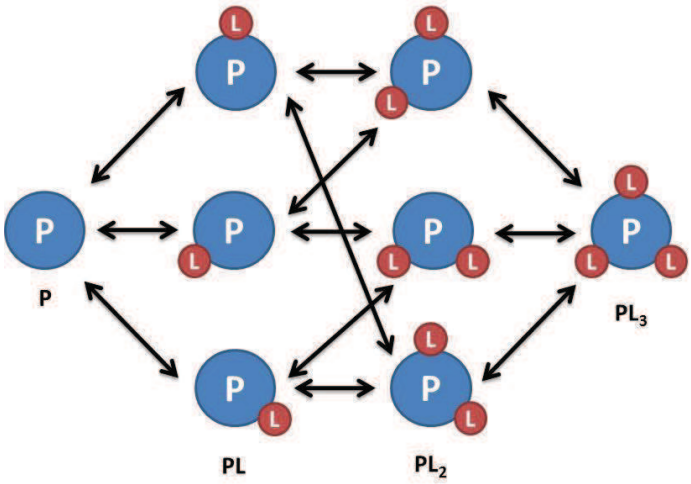


Figure 8.3 : Approche microscopique de la liaison de trois ligands L sur une macromolécule P.

Le modèle générique repose sur l'approche microscopique. Il est décrit dans un premier temps dans le cadre d'une macromolécule à *n* sites capables chacun de lier un seul type de ligand, puis dans le cas plus général d'une macromolécule susceptible de lier *m* ligands (identiques ou différents) sur *m* sites (virtuels ou réels)

8.1.2 Liaison d'un seul type de ligand à une macromolécule

Pour cette partie, nous considérons une macromolécule P qui possède *n* sites de liaison pour un ligand L. La représentation macroscopique définie ci-dessus nous amène à étudier les équilibres entre les espèces chimiques présentes, en considérant l'ensemble des réactions de type :

$$PL_{i-1} + L \overset{K_i}{\Leftrightarrow} PL_i \tag{8.1}$$

pour *i* variant de 1 à *n.* Nous définissons alors les constantes macroscopiques K_i ou constantes d'association d'Adair-Klotz par la relation :

$$K_i = \frac{[PL_i]}{[PL_{i-1}][L]} \tag{8.2}$$

Nous pouvons caractériser l'équilibre entre les espèces par un polynôme de liaison $p(x)$, défini comme la somme des concentrations des différentes espèces, normalisées par la concentration de macromolécule non liée :

$$p(x) = \sum_{i=0}^{n} \frac{[PL_i]}{[P]} \tag{8.3}$$

Cette équation peut être réécrite à l'aide des constantes d'Adair-Klotz :

$$p(x) = 1 + K_1 \cdot x + K_1 \cdot K_2 \cdot x^2 + \cdots + K_1 \dots K_n \cdot x^n \tag{8.4}$$

où x est la concentration du ligand L. Cette représentation conduit à un modèle permettant de s'ajuster aux mesures expérimentales à l'aide de n paramètres. Elle ne permet néanmoins pas de modéliser efficacement des systèmes, pour lesquels le comportement varie plus en fonction de l'occupation individuelle des sites, que du nombre total de sites occupés. Pour ce faire, nous devons avoir recours à la représentation microscopique définie dans la partie 8.1.1. Dans cette nouvelle représentation, la formation du complexe $PL_{i_1} \dots L_{i_n}$ (avec L représentant les n sites de liaison pour la macromolécule P et i_m un indice qui vaut 1 ou 0 selon que le $m^{\text{ième}}$ site est occupé ou non) peut être décrite par une constante d'équilibre $\beta_{i_1,i_2,\dots,i_n}$ définie par :

$$\beta_{i_1,i_2,\dots,i_n} = \frac{[PL_{i_1} \dots L_{i_n}]}{[P]\,[L]^{i_1} \dots [L]^{i_n}} = \frac{[PL_{i_1} \dots L_{i_n}][P]^{n-1}}{[PL_{i_1}][PL_{i_2}] \dots [PL_{i_n}]} \cdot \frac{[PL_{i_1}]}{[P][L]^{i_1}} \cdots \frac{[PL_{i_n}]}{[P][L]^{i_n}} \tag{8.5}$$

Dans l'approche microscopique, en lieu et place des constantes d'Adair-Klotz, nous définissons des constantes individuelles d'association pour chaque site et des facteurs de couplage entre les sites. Ces paramètres sont déduits directement des paramètres $\beta_{i_1,i_2,\dots,i_n}$. Premièrement, nous définissons le facteur de couplage $c_{i_1,i_2,\dots,i_n}$ associé au complexe $PL_{i_1} \dots L_{i_n}$ par :

$$c_{i_1,i_2,\dots,i_n} = \frac{[PL_{i_1} \dots L_{i_n}][P]^{n-1}}{[PL_{i_1}][PL_{i_2}] \cdots [PL_{i_n}]} \tag{8.6}$$

Par définition, $c_{i_1,i_2,\dots,i_n} = 1$, lorsqu'au maximum un seul indice est à 1. Avec les facteurs de couplage, l'équation (8.5) devient :

$$\beta_{i_1,i_2,\dots,i_n} = c_{i_1,i_2,\dots,i_n} \cdot \beta_{i_1,0,\dots,0} \cdot \dots \cdot \beta_{0,0,\dots,i_n} \tag{8.7}$$

avec i_m valant toujours 0 ou 1, en fonction de l'occupation des sites. Si les différents sites impliqués dans la formation d'un complexe sont indépendants et que l'occupation de l'un de ces sites n'affecte pas les autres, alors le facteur de couplage associé à ce complexe sera égal à 1. L'inverse n'est pas nécessairement vrai.

De plus, nous définissons la constante d'association individuelle pour le $j^{\text{ième}}$ site $k_j = \beta_{0,\dots 1,\dots,0}$ (où seul le $j^{\text{ième}}$ indice i_j de β est à 1). L'équation (8.7) peut maintenant s'écrire :

$$\beta_{i_1,i_2,\dots,i_n} = c_{i_1,i_2,\dots,i_n} \cdot k_1^{i_1} \cdot \dots \cdot k_n^{i_n} \tag{8.8}$$

Le polynôme de liaison (8.3) dans la représentation microscopique est une extension de sa version macroscopique. Il s'exprime à l'aide des paramètres microscopiques (k_j et $c_{i_1,\ldots,i_n}$) de la manière suivante :

$$p(x) = \sum_{s=0}^{n} x^s \sum_{\sum i_q = s} c_{i_1,\ldots,i_n} \cdot k_1^{i_1} \cdot \ldots \cdot k_n^{i_n} \tag{8.9}$$

Par identification entre les polynômes de liaison des deux représentations (macroscopique, équation (8.4) et microscopique, équation (8.9)), il est possible d'établir les liens entre les paramètres macroscopiques (constantes d'Adair-Klotz) et les paramètres microscopiques (constantes d'association propres aux sites et coefficients de couplage).

$$K_p = \sum_{\sum i_q = p} c_{i_1,\ldots,i_n} \cdot k_1^{i_1} \cdot \ldots \cdot k_n^{i_n} \tag{8.10}$$

L'équation du polynôme ainsi obtenue correspond au cas particulier où tous les ligands pouvant se lier à la macromolécule sont du même type. La partie suivante constitue la généralisation de cette approche pour des ligands différents.

8.1.3 Généralisation de la méthodologie de modélisation pour la liaison de différents types de ligands à une protéine

Pour la généralisation de la modélisation à base d'un polynôme de liaison, nous considérons maintenant une protéine P capable de lier différents ligands L_1, … , L_r, chacun sur un site de liaison propre. Cette approche permet de couvrir tous les cas de figure :

- Dans le cas d'une protéine disposant de *n* sites de liaisons pour *n* ligands différents, chacun étant capable de lier un ligand distinct, le modèle s'applique directement.
- Si les ligands sont tous de même type, les différents L_1, … , L_r seront les mêmes et les équations développées ci-après permettent de retrouver les équations de la partie 8.1.2.
- Si *p* ligands sont susceptibles de se lier sur le même site, nous décrirons le site en question comme *p* sites virtuels (un pour chaque ligand) en imposant des facteurs de couplages nuls entre ces sites virtuels. En effet, la liaison d'un ligand sur l'un des sites virtuels empêchera la liaison de tout autre ligand sur les autres sites virtuels associés.

Par extension des formules définies dans la section précédente, la formation du complexe ${PL^1}_{i_1} \dots {L^r}_{i_r}$ peut être décrite par une constante d'équilibre $\beta_{i_1,i_2,\dots,i_r}$, définie par :

$$\beta_{i_1,i_2,\dots,i_r} = \frac{\left[{PL^1}_{i_1} \dots {L^r}_{i_r}\right]}{[P]\,[L^1]^{i_1} \cdots [L^r]^{i_r}} \tag{8.11}$$

Le polynôme de liaison associé à ce système devient :

$$p(x_1, \dots, x_r) = \sum_{i_1,\dots,i_r} \beta_{i_1,i_2,\dots,i_n} \cdot {x_1}^{i_1} \cdot \dots \cdot {x_r}^{i_r} \tag{8.12}$$

où x_k est la concentration du ligand L^k se fixant sur le site k et $\beta_{0,0,\dots,0}$ est défini par convention comme égal à 1. Une fois l'équation (8.12) développée avec les constantes d'association et les facteurs de couplage définis de la même manière que dans la section 8.1.2, nous obtenons le polynôme de liaison suivant :

$$p(x_1, \dots, x_r) = \sum_{i_1=0}^{1} \sum_{i_2=0}^{1} \cdots \sum_{i_r=0}^{1} c_{i_1,\dots,i_r} \cdot k_1^{i_1} \cdot \dots \cdot k_r^{i_r} \cdot {x_1}^{i_1} \cdot \dots \cdot {x_r}^{i_r} \tag{8.13}$$

Avec par convention $c_{0,0,\dots,0}$ = 1. Chaque terme de l'équation (8.13) représente bien la concentration relative (la référence étant la concentration de la macromolécule P non liée à l'équilibre) de chaque configuration ${PL^1}_{i_1} \dots {L^r}_{i_r}$.

Notons **A** = (a_1, …, a_r) un vecteur quelconque constitué de r composantes ayant des valeurs binaires et $\mathscr{E}$ l'ensemble fini des vecteurs **A**. En faisant correspondre la $k^{\text{ième}}$ composante de **A** avec le $k^{\text{ième}}$ indice i_k dans l'équation (8.13), nous pouvons réécrire le polynôme de liaison de manière compacte :

$$p(x_1, \dots, x_r) = \sum_{\mathscr{E}} \left[c_{\mathbf{A}} \cdot \prod_{p=1}^{r} \left(k_p \cdot x_p\right)^{a_p} \right] \tag{8.14}$$

où $c_{\mathbf{A}}$ correspond au coefficient de couplage $c_{a_1}, \dots, c_{a_r}$ défini précédemment.

L'étude complète d'une macromolécule pouvant lier plusieurs types de ligands nécessite la mesure de courbes dose-réponse multidimensionnelles (où nous pouvons faire varier indépendamment chacune des concentrations des ligands). A la place, nous modélisons le plus souvent les courbes dose-réponses pour un ligand donné λ, en fixant les concentrations des autres ligands. Le modèle est alors obtenu à partir du polynôme de liaison $p_\lambda(\lambda)$ spécifique au ligand λ.

Dans un premier temps, nous supposons que tous les ligands L_1, … L_r sont différents. Au niveau macroscopique, nous définissons une constante d'association apparente K_{app} et le polynôme de liaison spécifique au ligand λ se résume alors à une équation linéaire :

$$p_\lambda(\lambda) = \alpha_\lambda \cdot \left(1 + K_{app} \cdot \lambda\right) \tag{8.15}$$

La constante α_λ est introduite à des fins de normalisation. En effet, dans le polynôme de liaison, le terme '1' correspond à la macromolécule non liée, alors que dans le polynôme spécifique à λ, il correspond à l'ensemble des configurations pour lesquelles le ligand λ n'est pas lié. α_λ correspond au rapport entre la concentration de macromolécule non liée et la somme des concentrations des configurations pour lesquelles λ n'est pas lié. Bien évidemment, les constantes α_λ et K_{app} dépendent des concentrations des ligands autres que λ.

L'expression de $p_\lambda(\lambda)$ peut également être obtenue à partir du modèle microscopique de l'équation (8.14). Supposons que λ soit le ligand qui se fixe au $m^{ième}$ site. Nous divisons l'ensemble $\mathscr{E}$ en deux sous-ensembles : $\mathscr{E}_0$ correspondant à l'ensemble des vecteurs **A** pour lesquels $a_m = 0$ et $\mathscr{E}_1$ correspondant à l'ensemble des vecteurs **A** pour lesquels $a_m = 1$. Ainsi, nous pouvons écrire :

$$p_\lambda(\lambda) = \sum_{\mathscr{E}_r}\left[c_{\mathbf{A}} \cdot \prod_{\substack{p=1 \\ p\neq m}}^{r}\left(k_p \cdot x_p\right)^{a_p}\right] + \lambda \cdot \left(k_m \cdot \sum_{\mathscr{E}_r}\left[c_{\mathbf{A}} \cdot \prod_{\substack{p=1 \\ p\neq m}}^{r}\left(k_p \cdot x_p\right)^{a_p}\right]\right) \tag{8.16}$$

Par identification entre les équations (8.14) et (8.16), nous trouvons les expressions de α_λ et K_{app} :

$$\alpha_\lambda = \sum_{\mathscr{E}_r} c_{\mathbf{A}} \cdot \prod_{\substack{p=1 \\ p\neq m}}^{r}\left(k_p \cdot x_p\right)^{a_p} \tag{8.17}$$

$$K_{app} = \frac{k_m \cdot \sum_{\mathscr{E}_r} c_{\mathbf{A}} \cdot \prod_{\substack{p=1 \\ p\neq m}}^{r}\left(k_p \cdot x_p\right)^{a_p}}{\sum_{\mathscr{E}_r} c_{\mathbf{A}} \cdot \prod_{\substack{p=1 \\ p\neq m}}^{r}\left(k_p \cdot x_p\right)^{a_p}} \tag{8.18}$$

Considérons maintenant le cas où l'espèce λ correspond à m sites de liaison. Avec l'approche macroscopique, ce cas de figure se modélise exactement de la même manière qu'un système à un seul type de ligand (voir section 8.1.2), à la différence près que les constantes d'association macroscopique K_i sont remplacées par des constantes d'association apparentes $K_{i,app}$. Elles correspondent aux constantes d'association d'un $i^{ème}$ ligand λ à la macromolécule, dans le cas où les concentrations des espèces autres que λ sont fixées. Avec ces constantes macroscopiques, le

polynôme de liaison spécifique s'écrit alors (la constante α_λ est introduite pour des raisons similaires à celles exprimées ci-dessus) :

$$p_\lambda(\lambda) = \alpha_\lambda \cdot \sum_{i=0}^{m} \left[\lambda^i \cdot \prod_{s=1}^{i} K_{s,app} \right] \tag{8.19}$$

Là encore, ce même polynôme peut être obtenu à l'aide de l'approche microscopique. Soit f une fonction de permutation de $\mathbb{N}$ vers $\mathbb{N}$ permettant de trier les numéros des sites de liaison de sorte à ce que les m premiers correspondent au ligand λ. Les constantes c' et k' correspondent aux constantes c et k après permutation des numéros de site.

$$c'_{i_1,\ldots,i_r} = c_{f^{-1}(i_1),\ldots,f^{-1}(i_r)} \tag{8.20}$$

$$k'_j = k_{f^{-1}(j)} \tag{8.21}$$

L'ensemble $\mathscr{E}$ est maintenant divisé en m+1 sous-espaces : $\mathscr{E}_0$, qui contient l'ensemble des vecteurs **A** pour lesquels les m premiers indices sont tous à 0, $\mathscr{E}_1$, qui contient l'ensemble des vecteurs **A** pour lesquels un et un seul des m premiers indices est à 1, $\mathscr{E}_2$, qui contient l'ensemble des vecteurs **A** pour lesquels exactement 2 des m premiers indices sont à 1 et, par extension, $\mathscr{E}_q$, l'ensemble des vecteurs **A** pour lesquels exactement q des m premiers indices sont à 1. Avec cette notation, le polynôme de liaison spécifique à λ s'écrit :

$$p_\lambda(\lambda) = \sum_{q=0}^{m} \left[\lambda^q \cdot \prod_{s=1}^{m} {k'_s}^{a_s} \cdot \sum_{\mathscr{E}_q} \left[{c'}_{\mathbf{A}} \cdot \prod_{p=m+1}^{r} \left({k'}_p \cdot x_p \right)^{a_p} \right] \right] \tag{8.22}$$

Par identification des termes entre les équations (8.19) et (8.22) nous en déduisons le lien entre les constantes microscopiques et les constantes apparentes microscopiques :

$$\alpha_\lambda = \sum_{\mathscr{E}} {c'}_{\mathbf{A}} \cdot \prod_{p=m+1}^{r} \left({k'}_p \cdot x_p \right)^{a_p} \tag{8.23}$$

Pour s variant de 1 à q, nous obtenons également :

$$\prod_{s=1}^{q} K_{s,app} = \frac{\sum_{\mathscr{E}_q} {c'}_{\mathbf{A}} \cdot \prod_{p=m+1}^{r} \left({k'}_p \cdot x_p \right)^{a_p} \cdot \prod_{s=1}^{m} {k'_s}^{a_s}}{\sum_{\mathscr{E}} {c'}_{\mathbf{A}} \cdot \prod_{p=m+1}^{r} \left({k'}_p \cdot x_p \right)^{a_p}} \tag{8.24}$$

8.1.4 Polynôme de liaison et signal observable

Le polynôme de liaison ne peut être utilisé directement pour estimer les concentrations relatives des différents sous-complexes et donc obtenir les modèles correspondant aux courbes dose-réponse expérimentales. Il faut pour cela calculer le signal observable. A partir du polynôme de liaison, nous définissons ϑ_j, le rapport de concentration spécifique au ligand L^j comme le rapport entre la somme des concentrations normalisées de l'ensemble des espèces pour lesquelles L^j est lié, et la concentration normalisée initiale de macromolécule P. Mathématiquement, le ratio ϑ_j est obtenu de la manière suivante :

$$\vartheta_j = x_j \cdot \frac{\frac{\partial p(x_1, \dots, x_r)}{\partial x_j}}{p(x_1, \dots, x_r)} \tag{8.25}$$

Enfin, lorsque nous confrontons des résultats expérimentaux aux modèles, nous avons rarement une mesure directe de la concentration d'une espèce mais le plus souvent la mesure d'un signal global S (correspondant à la mesure de fluorescence par exemple) auquel contribue plus ou moins chacune des configurations $PL^1{}_{i_1} \dots L^r{}_{i_r}$. Pour obtenir la valeur du signal global à partir du polynôme, nous associons un signal molaire σ_j à chaque complexe $PL^1{}_{i_1} \dots L^r{}_{i_r}$ pour lequel le ligand L^j est lié. Le signal s_j propre à cet ensemble de complexe est défini par :

$$s_j = \sigma_j \cdot \vartheta_j \tag{8.26}$$

et le signal total mesuré S_T comme suit :

$$S_T = \sum s_j = \sum \sigma_j \cdot \vartheta_j \tag{8.27}$$

L'équation (8.27) est donc l'expression du modèle qui, dans une démarche d'extraction des paramètres, sera comparée aux résultats expérimentaux.

Le modèle générique nous permet de nous rapprocher au plus près du comportement réel des mécanismes biologiques, au détriment d'un nombre très important de paramètres, valant $2^r - 1$ dans le cadre général de r sites réels avec r ligands. Ce nombre peut-être réduit en admettant un certain nombre d'hypothèses sur le comportement microscopique. En faisant cela, nous retombons sur des approches de modélisation plus classiques. C'est ce que nous allons voir dans la section suivante.

8.2 Lien avec les approches de modélisation standards

Nous avons appliqué la méthodologie de modélisation proposée sur un exemple de macromolécule P possédant deux sites de liaison, le premier pour un ligand L^1 et le deuxième pour un ligand L^2. Cette situation est illustrée Figure 8.4. Nous allons notamment montrer que ce modèle permet d'unifier les approches de modélisation standard, en commençant par le modèle de l'ajustement induit.

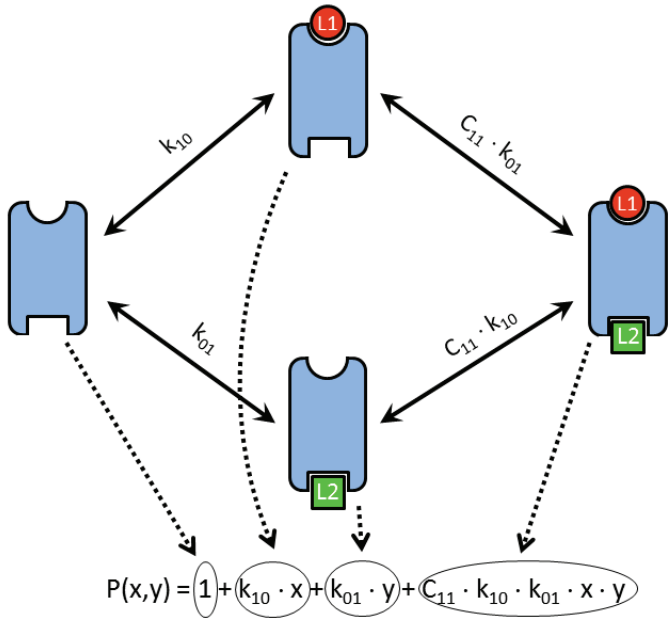


Figure 8.4 : Représentation d'une protéine pouvant lier deux ligands L_1 et L_2 et le lien entre les différents complexes moléculaires et les termes du polynôme de liaison.

8.2.1 Modèle de l'ajustement induit

Le modèle de l'ajustement induit est une application directe de la représentation microscopique du système biologique. En partant de l'équation (8.12), nous pouvons associer à ce système biologique le polynôme de liaison équation (8.28). Chaque terme du polynôme correspond à un des 4 complexes du système illustré sur la Figure 8.4 :

$$p(x,y) = 1 + k_{1,0} \cdot x + k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y \tag{8.28}$$

avec x et y, les concentrations respectives des ligand L^1 et L^2. $k_{1,0}$ et $k_{0,1}$ sont les constantes d'association pour chaque site et $c_{1,1}$ est le facteur de couplage entre les deux sites. Lorsque le site associé au ligand L^1 est occupé, la protéine peut subir un changement de conformation, ce qui modifie le deuxième site, et, par conséquent, l'affinité de liaison pour le ligand L^2. Pour tenir compte de ce phénomène, il faudrait définir deux constantes d'affinité pour le site L^2, l'une

correspondant au cas où L^1 est lié et l'autre au cas où il ne l'est pas. En pratique, nous utilisons le facteur de couplage $c_{1,1}$ pour distinguer les deux valeurs possibles de la constante d'affinité du deuxième site qui vaut soit $k_{0,1}$ lorsque L^1 n'est pas lié soit $c_{1,1} \cdot k_{0,1}$ lorsque L^1 est lié. $c_{1,1}$ est toujours un réel positif et nous pouvons distinguer trois cas de figure :

- si $c_{1,1}$ est supérieur à 1, l'occupation d'un site va induire une augmentation de l'affinité du site jumelé. Nous parlons alors de coopérativité positive ;
- si $c_{1,1}$ est inférieur à 1, l'affinité de liaison pour le ligand L^2 est diminuée lorsque L^1 est déjà lié. Nous parlons alors de coopérativité négative ;
- si $c_{1,1}$ est égal à 1, les deux sites se comportent comme deux sites indépendants.

Le raisonnement est identique si nous inversons les rôles des sites 1 et 2.

A partir du polynôme de liaison et de l'équation (8.25), nous pouvons obtenir les rapports de concentration ϑ_x et ϑ_y, relatifs respectivement aux ligands L^1 et L^2, qui sont donnés par :

$$\vartheta_x = \frac{x \cdot \frac{\partial p}{\partial x}}{p(x,y)} = \frac{k_{1,0} \cdot x + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y}{1 + k_{1,0} \cdot x + k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y} \tag{8.29}$$

$$\vartheta_y = \frac{y \cdot \frac{\partial p}{\partial y}}{p(x,y)} = \frac{k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y}{1 + k_{1,0} \cdot x + k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y} \tag{8.30}$$

Afin de comparer ces deux équations à des résultats expérimentaux, il reste encore à associer un signal molaire à chacun des complexes. Celui-ci dépendra des observables obtenus, ce qui est expliqué dans la partie précédente. Pour le ligand L^1 nous associons le signal molaire σ_x et pour le ligand L^2 nous associons le signal molaire σ_y. Nous obtenons ainsi le signal total S_T :

$$\begin{aligned} S_T &= \sum \sigma_j \cdot \vartheta_j = \sigma_x \cdot \vartheta_x + \sigma_y \cdot \vartheta_y \\ &= \frac{\sigma_x \cdot \left(k_{1,0} \cdot x + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y\right) + \sigma_y \cdot \left(k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y\right)}{1 + k_{1,0} \cdot x + k_{0,1} \cdot y + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y} \end{aligned} \tag{8.31}$$

Cette partie nous a permis de présenter l'approche de modélisation de l'ajustement induit sur le cas simple de la liaison entre une macromolécule et deux ligands différents. Dans cette approche, la totalité des paramètres microscopiques apparait, ce qui rend leur extraction complexe. En fonction des mesures et des connaissances déjà acquises sur le fonctionnement des mécanismes étudiés, il est possible d'émettre un certain nombre d'hypothèses et de fixer certains paramètres microscopiques.

8.2.2 Modèle séquentiel

Le modèle séquentiel consiste à considérer que la macromolécule va lier les ligands sur les différents sites de manière séquentielle, un site n'étant capable de lier un ligand que si les sites précédents sont occupés [137]. La protéine, dans sa configuration sans aucun ligand lié, présente un site de liaison unique pour le ligand L^1. Dès que ce ligand se lie, la protéine subit un changement de conformation, qui permet ensuite au ligand L^2 de se lier sur le second site, et ainsi de suite.

Reprenons l'exemple précédent (Figure 8.4), en considérant par exemple que L^1 se liera avant L^2. Le modèle séquentiel correspond à un modèle générique pour lequel $k_{0,1}$ tend vers zéro et $c_{1,1}$ vers l'infini. Le produit $k_{0,1} \cdot c_{1,1}$, dont la valeur tend, dans un cas idéal, vers une forme indéterminée zéro multiplié par l'infini, correspond alors à l'affinité du ligand L^2 une fois le ligand L^1 lié. Le polynôme de liaison résultant est donc de la forme simplifiée :

$$p(x,y) = 1 + k_{1,0} \cdot x + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y \tag{8.32}$$

Dans ces conditions, le rapport de concentration ϑ_y, spécifique au ligand L^2 ne dépend plus que du produit $x \cdot y$, ce qui redémontre que la présence de L^1 est nécessaire pour lier L^2.

$$\vartheta_y = \frac{y \cdot \frac{\partial p}{\partial y}}{p(x,y)} = \frac{c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y}{1 + k_{1,0} \cdot x + c_{1,1} \cdot k_{1,0} \cdot k_{0,1} \cdot x \cdot y} \tag{8.33}$$

L'approche peut être étendue à plusieurs ligands. Le modèle séquentiel est donc bien un cas particulier du modèle générique présenté dans ce chapitre. Il tend à se rapprocher du modèle macroscopique, si nous identifions $k_{1,0}$ à K_1 et $c_{1,1} \cdot k_{0,1} \cdot k_{1,0}$ à K_2.

8.2.3 Modèle allostérique

Le modèle allostérique consiste à considérer la protéine comme étant en équilibre entre deux conformations (tendue et relâchée). Chaque conformation présente un site pour un ou plusieurs ligands (dans cet exemple, nous ne considèrerons qu'un ligand L^1) [127]. Le passage d'une conformation à l'autre se fait par liaison ou libération d'une molécule d'eau, réelle ou virtuelle. Nous allons montrer dans cette partie que le modèle générique présenté ici permet également de retomber sur le modèle allostérique, sous certaines conditions.

Considérons que la molécule d'eau permettant le changement de conformation n'est autre qu'un deuxième ligand L^2. Nous avons donc un équilibre entre la protéine P, dans une conformation appelée tendue, et le complexe P/H_2O, dans une conformation appelée relâchée. L'étude de la

liaison du ligand sur une protéine en équilibre entre deux conformations (modèle allostérique) et l'étude d'une protéine pouvant lier deux ligands, L^1 et l'eau, selon une approche microscopique sont donc équivalentes. Un tel système est représenté Figure 8.5. Nous y retrouvons l'équivalence entre le modèle microscopique en A. et le modèle allostérique classique en B.

Dans le modèle allostérique, nous considérons l'eau comme étant en excès et donc sa concentration comme constante. Cette hypothèse semble cohérente avec les conditions expérimentales utilisées pour la validation des modèles.

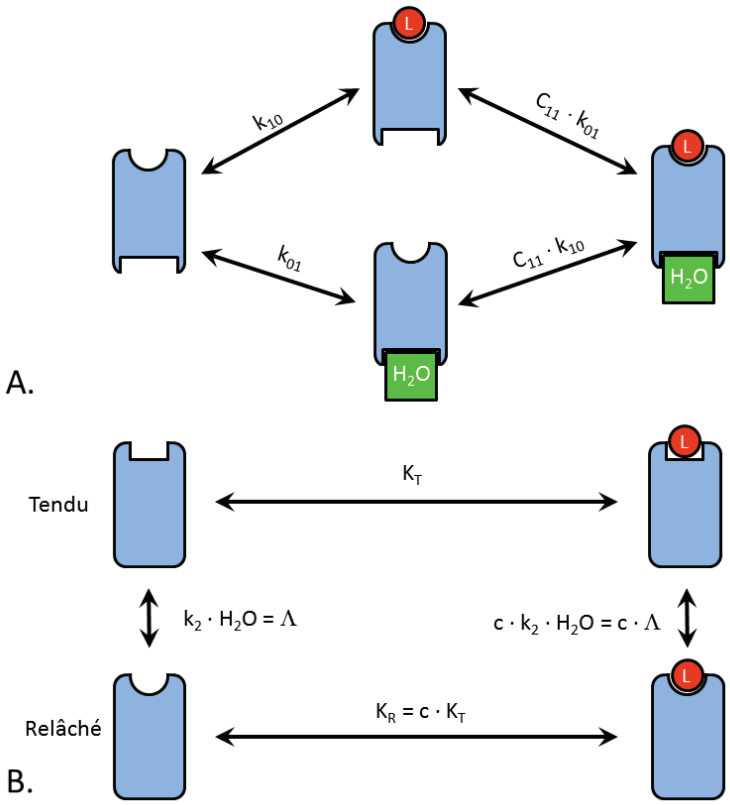


Figure 8.5 : A. Une protéine pouvant lier deux ligands chacun sur un site différent, dont une molécule d'eau. B. Le même système, mais représenté suivant l'approche allostérique, avec la protéine dans un équilibre entre deux états, tendu et relâché.

En se référant à la Figure 8.5.A, le polynôme de liaison de la liaison $P/L^1/H_2O$ peut s'écrire sous la forme suivante :

$$P(x) = (1 + k_{0,1} \cdot y) + k_{1,0} \cdot x \cdot (1 + c_{1,1} \cdot k_{0,1} \cdot y) \tag{8.34}$$

où x est la concentration de L^1 et y la concentration d'eau. Si y est biologiquement en excès, le polynôme peut également s'écrire, à une constante près, comme le polynôme de liaison d'un système à un seul ligand :

$$P'(x) = 1 + k'_1 \cdot x \tag{8.35}$$

avec le coefficient de couplage k'_1 égal à:

$$k'_1 = \frac{k_{1,0} \cdot (1 + c_{1,1} \cdot k_{0,1} \cdot y)}{(1 + k_{0,1} \cdot y)} \tag{8.36}$$

En introduisant $\Lambda = k_{0,1} \cdot y$, $K_T = k_{1,0}$ et $c = c_{1,1}$, nous trouvons un lien entre les paramètres du modèle à ajustement induit et ceux du modèle allostérique (Λ, K_T et c). L'expression du coefficient de l'affinité équivalent du site L^1 k'_1, avec les paramètres allostériques est :

$$k'_1 = K_T \cdot \frac{(1 + \Lambda \cdot \mathrm{c})}{(1 + \Lambda)} \tag{8.37}$$

Il est à noter que dans cet exemple particulier pour lequel la macromolécule ne dispose que d'un site en plus de celui de l'eau, les représentations macroscopiques et microscopiques sont équivalentes : la constante d'affinité équivalente k'_1 est égale à la constante d'Adair-Klotz du modèle macroscopique K'_1. Il est bien sûr possible de modéliser des systèmes plus complexes en ajoutant d'autres sites de liaison pour des ligands ainsi que d'autres sites, réels ou virtuels, pour les molécules d'eau. La complexité du système augmentera, mais il sera toujours possible d'exprimer le modèle allostérique à partir de l'approche générique.

8.2.4 Lien avec l'équation de Hill

Dans cette dernière partie, nous allons faire le lien entre notre modèle générique et l'équation de Hill, utilisée jusqu'ici dans les modèles pour modéliser divers aspects, comme les mécanismes intervenant autour du promoteur d'un brin d'ADN (fixation des activateurs et des répresseurs). Repartons pour cela du polynôme de liaison de l'équation (8.28), en considérant que les ligands L^1 et L^2 sont de même type et que les sites de liaisons associés sont donc équivalents. Dans ce contexte, nous avons les propriétés suivantes :

$$\begin{aligned} k_{1,0} &= k_{0,1} = k \\ y &= x \\ c_{1,1} &= c \end{aligned} \tag{8.38}$$

Ceci conduit à une simplification de l'équation (8.28):

$$p(x) = 1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2 \tag{8.39}$$

Ici, le signal étudié correspond au nombre total de ligands liés à la macromolécule. Nous posons donc le signal molaire $\sigma_x = 1$ et le signal total associé à cette interaction s'écrit alors :

$$S_T = \sigma_x \cdot \vartheta_x + \sigma_x \cdot \vartheta_x = 2 \cdot \sigma_x \cdot \frac{x \cdot \frac{\partial p}{\partial x}}{p(x)} = \frac{2 \cdot \sigma_x \cdot (2 \cdot k \cdot x + c \cdot k^2 \cdot x^2)}{1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2} \tag{8.40}$$

Pour rappel, l'équation de Hill est une équation phénoménologique permettant simplement (avec peu de paramètres) de modéliser une courbe dose-réponse de forme classique et pour laquelle le signal observable s'écrit sous la forme suivante :

$$S_T = \frac{S_{MAX}}{1 + \left(\frac{K}{x}\right)^n} \tag{8.41}$$

avec S_{MAX} le signal maximum observable (valeur lorsque la concentration x tend vers l'infini), K le coefficient de Hill associé à la variable x (égal à la valeur de x pour lequel le signal vaut $½S_{MAX}$) et n le nombre de Hill traduisant la pente de la transition dans la courbe dose-réponse. En principe, n = 1 si les sites de liaisons sont indépendants, n < 1, dans le cas d'une coopération négative entre les sites et n > 1 et tend vers r (où r est le nombre de sites), dans le cas d'une coopération positive.

8.2.4.1 Sites indépendants

Supposons dans un premier temps que les deux sites sont indépendants. Dans le cadre de notre modèle, cela se traduit par un coefficient de couplage c = 1. Dans ce cas, le signal s'écrit de la manière suivante :

$$S_T = \frac{2 \cdot k \cdot x + 2 \cdot k^2 \cdot x^2}{1 + 2 \cdot k \cdot x + k^2 \cdot x^2} = \frac{2 \cdot k \cdot x \cdot (1 + k \cdot x)}{(1 + k \cdot x)^2} = \frac{2 \cdot k \cdot x}{1 + k \cdot x} = \frac{2}{1 + \left(\frac{K}{x}\right)} \tag{8.42}$$

L'équation (8.42) correspond à l'équation de Hill pour laquelle la constante de Hill K vaut $1/k$ et le nombre de Hill n = 1, ce qui est la valeur attendue pour des sites indépendants. Ce lien a été démontré ici dans le cadre d'un système à 2 ligands. La généralisation de ce calcul pour un système à r ligands est donnée en Annexe I.1.

8.2.4.2 Coopérativité positive

Dans le cadre d'une coopérativité positive extrême (correspondant au modèle séquentiel, ou en enzymologie, au modèle compétitif), k peut prendre une valeur très faible, tandis que c tend vers l'infini. Dans la zone d'intérêt de la courbe dose-réponse, le produit $k \cdot x$ est de l'ordre de 1, par

conséquent, $k \cdot x$ et $(k \cdot x)^2$ sont du même ordre de grandeur. c étant très grand, nous pouvons écrire que :

$$c \cdot k^2 \cdot x^2 \gg 2 \cdot k \cdot x \tag{8.43}$$

Cette hypothèse permet de simplifier l'expression du signal total de la manière suivante :

$$S_T \cong \frac{2 \cdot c \cdot k^2 \cdot x^2}{1 + c \cdot k^2 \cdot x^2} = \frac{2}{1 + \frac{1}{c \cdot k^2 \cdot x^2}} = \frac{2}{1 + \left(\frac{K}{x}\right)^2} \tag{8.44}$$

Nous retrouvons donc l'équation de Hill, avec la constante de Hill K égal à :

$$K = \frac{1}{\sqrt{c} \cdot k} \tag{8.45}$$

Le nombre de Hill est égal à 2, ce qui est la valeur attendue dans le cas d'un système coopératif extrême. En pratique, l'approximation (8.44) n'est valable qu'autour du point d'inflexion de la courbe dose-réponse et le coefficient de couplage n'est pas infini. Cela va avoir pour conséquence de dégrader le nombre de Hill qui va donc se situer entre 1 et 2 dans un cas réel. Ce calcul peut être généralisé à une macromolécule liant r ligands en Annexe I.2.

8.3 Générateur du polynôme de liaison

Pour simplifier l'utilisation du modèle générique, nous avons développé un programme sous Matlab afin d'automatiser la génération du polynôme. L'interface graphique du programme est illustrée Figure 8.6. Dans un premier temps, nous abordons le formalisme employé dans le programme, puis son utilisation.

8.3.1 Formalisme

Soit une macromolécule P sur laquelle des ligands L peuvent se lier. Selon la théorie développée dans la section précédente, le modèle polynômial pour une approche microscopique va dépendre de deux jeux de paramètres, les k_j (les constantes d'association pour chaque site différent) et les $c_{i_1,i_2,\ldots,i_r}$ (les facteurs de couplage entre r sites). Dans le cas présenté précédemment des deux ligands, nous pouvons facilement déduire les sites auxquels se réfèrent les coefficients et ainsi les manipuler dans le programme. Cependant, quand le nombre de ligand augmente cela devient plus complexe et il est nécessaire de développer une règle de notation.

Pour simplifier la manipulation de ces coefficients, nous noterons les constantes d'association de chacun des sites $k_{i_1,i_2,\ldots,i_r}$, où, pour le $j^{\text{ième}}$ site, seul l'indice i_j vaut 1 (les autres valant 0). Par

exemple, dans le cas d'une macromolécule à 4 sites, la constante d'association pour le site de liaison numéro 3 va être définie par $k_{0,0,1,0}$. Pour les facteurs de couplage $c_{i_1,i_2,\ldots,i_n}$, la même notation que dans la partie théorique sera utilisée. Par exemple, le coefficient de couplage d'une macromolécule à quatre sites pour lesquels les sites 1, 2 et 4 sont occupés s'écrit $c_{1,1,0,1}$. Plus le nombre de ligands ou de sites est important, plus cette nomenclature est lourde à utiliser pour la génération du polynôme. Pour simplifier les notations, nous transformons les indices des coefficients k et c en un nombre décimal correspondant au code binaire formé par les indices i_1, i_2,..., i_r (i_r étant le bit de poids fort). Ainsi $k_{0,0,1,0}$ devient k_4 et $c_{1,1,0,1}$ devient c_{11}. Cette nouvelle notation n'est utilisée que dans le moteur de calcul du programme. Elle est donc transparente pour l'utilisateur via l'interface graphique (voir Figure 8.6).

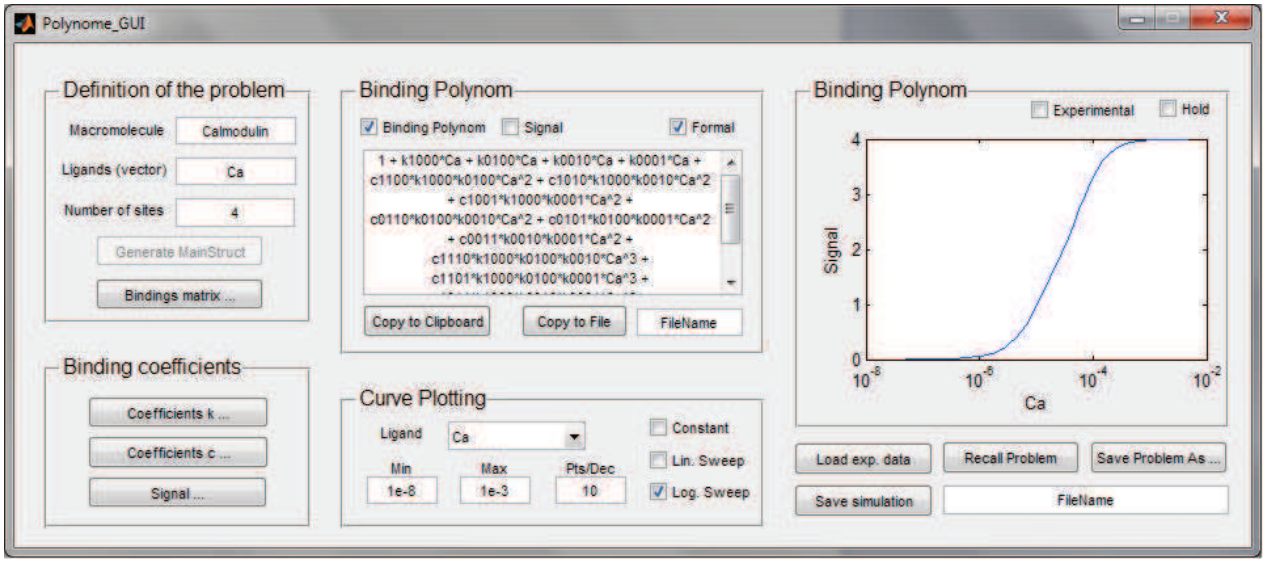


Figure 8.6 Fenêtre principale de l'interface graphique du générateur automatique de polynôme de liaison.

8.3.2 Paramètres d'entrée

Les paramètres d'entrée du calculateur sont fournis via une interface graphique. L'utilisateur spécifie notamment : le nom de la macromolécule, un vecteur contenant les ligands susceptibles de se lier à la macromolécule et le nombre de sites (qui correspond au nombre réel de sites de liaison de la molécule). A partir de ces informations, une matrice de liaison est proposée à l'utilisateur, lui permettant d'affecter un ou plusieurs sites à chaque ligand. Dans le cas où deux ligands partagent le même site réel de liaison, deux sites virtuels sont créés (un pour chaque ligand) et le coefficient de couplage de ces deux sites est automatiquement réglé à 0, comme nous l'avions évoqué dans la partie théorique. La deuxième étape de paramétrage du système biologique est la définition des paramètres k, c et σ. Seuls les paramètres variables sont proposés à l'utilisateur, ceux fixés par convention à 1 ou 0 ne sont pas affichés.

L’ensemble des paramètres est sauvegardé dans une structure Matlab qui est ensuite traitée par l’algorithme de calcul. Ces structures de données peuvent être sauvegardées, exportées sous la forme d’un fichier texte, ou rechargées et importées depuis un fichier texte créé par l’utilisateur.

8.3.3 Algorithme de calcul

L’algorithme de calcul permet notamment d’évaluer les équations formelles et numériques (avec les valeurs numériques des paramètres spécifiés par l’utilisateur) du polynôme de liaison et du signal associé, ainsi que de tracer les courbes dose-réponses théoriques calculées à partir du modèle selon différentes formes et échelles. Afin de générer le polynôme de manière automatique, nous utilisons l’équation (8.14) que nous avons adapté au formalisme décrit dans le premier paragraphe.

L'utilisation de l’interface est intuitive. Un manuel d’utilisation est proposé en Annexe J avec un exemple d'application du logiciel dans le cas de la liaison des ions calcium à la calmoduline, dont les résultats sont présentés dans la section suivante.

8.4 Application de la méthodologie au cas de la calmoduline

La calmoduline est une protéine possédant quatre sites sur lesquels peuvent se lier des ions calcium [138]. Les courbes dose-réponses (nombre moyen de calcium liés à la calmoduline en fonction de la concentration d’ions calcium), obtenues au moyen de différentes techniques de mesure, montrent un comportement classique, de type « sigmoïde » qui peut au premier ordre être décrit par un modèle phénoménologique simple de type équation de Hill. Toutefois, le comportement microscopique de la calmoduline (la formation des liaisons calmoduline-calcium et les changements de sa conformation au cours de cette liaison) demeure toujours inconnu en raison de la difficulté à obtenir suffisamment de résultats expérimentaux permettant d’isoler le comportement individuel de chacun des sites. Plusieurs hypothèses ont été avancées, conduisant à trois modèles classiques. Le premier repose sur l’existence de deux sites ayant une forte affinité de liaison et deux sites ayant une faible affinité de liaison. Il a été introduit par Crouch et Klee [138] puis développé par Wang [139]. Le second modèle, proposé par Haiech [137], développe un modèle séquentiel. Enfin le troisième modèle, proposé par Stefan *et al.* [140], est un modèle allostérique. Dans cette section, nous allons appliquer le modèle générique à ce problème en réutilisant les hypothèses formulées par Crouch et Klee, Haiech ou encore Stefan. La validation du modèle se fera sur une courbe dose-réponse réalisée par Crouch et Klee afin d’utiliser les mêmes données expérimentales pour comparer les différents modèles [138].

Pour réaliser la modélisation du système à l’aide du modèle générique, nous utilisons le logiciel en spécifiant que la macromolécule en présence est la calmoduline qui possède quatre sites

auxquels se lient les ligands Ca^{2+}. Cet exemple de l'utilisation du programme est développé en Annexe J. Nous obtenons le polynôme suivant :

$$\begin{aligned} P(x) = 1 + \ & (k_{1000} + k_{0100} + k_{0010} + k_{0001}) \cdot x + (\mathrm{c}_{1100} \cdot k_{1000} \cdot k_{0100} + c_{1010} \\ & \cdot k_{1000} \cdot k_{0010} + c_{1001} \cdot k_{1000} \cdot k_{0001} + c_{0110} \cdot k_{0100} \cdot k_{0010} \\ & + c_{0101} \cdot k_{0100} \cdot k_{0001} + c_{0011} \cdot k_{0010} \cdot k_{0001}) \cdot x^2 \\ & + \ (c_{1110} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} + c_{1101} \cdot k_{1000} \cdot k_{0100} \cdot k_{0001} \\ & + c_{1011} \cdot k_{1000} \cdot k_{0010} \cdot k_{0001} + c_{0111} \cdot k_{0100} \cdot k_{0010} \cdot k_{0001}) \cdot x^3 \\ & + \ c_{1111} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} \cdot k_{0001} \cdot x^4 \end{aligned} \tag{8.46}$$

où x est la concentration d'ions calcium et les c et les k sont les coefficients de couplage tels que définis en 8.3.1. Ce polynôme est constitué de 15 paramètres qui ne peuvent pas tous être extraits d'une simple courbe expérimentale. Il est donc nécessaire de simplifier le problème en émettant un certain nombre d'hypothèses qui seront calquées sur les modèles existants. Ces hypothèses se traduiront mathématiquement par des contraintes sur les coefficients du polynôme de liaison. Nous rappelons également que dans la démarche de recherche de modèle expliqué en introduction, le fait de trouver un jeu de paramètres qui permette d'ajuster la courbe expérimentale ne suffit pas à valider une hypothèse. En revanche, le contraire est suffisant pour l'invalider.

8.4.1 Deux sites forts et deux sites faibles

Ce modèle consiste à considérer que la calmoduline dispose de deux sites ayant une forte affinité de liaison des ions calcium et deux sites avec une faible affinité. Dans cette approche, il y a coopérativité positive entre les deux paires de site et la liaison d'ions calcium sur les deux sites forts entrainant un changement de conformation qui augmente considérablement l'affinité des deux sites faibles (Crouch et Klee [138]).

Ces hypothèses se traduisent mathématiquement par des choix sur les paramètres suivants du modèle (on considère dans la suite que les sites forts sont les sites 1 et 2) :

- Les affinités des sites forts $k_{1,0,0,0}$ et $k_{0,1,0,0}$ sont très grands devant $k_{0,0,1,0}$ et $k_{0,0,0,1}$.
- Les facteurs de couplage de type $c_{1,1,i3,i4}$ sont fixés au moins un ordre de grandeur au-dessus des autres paramètres.

Ces hypothèses conduisent à une version simplifiée de l'équation (8.46) :

$$P(x) \cong 1 + (k_{1000} + k_{0100}) \cdot x + (\mathrm{c}_{1100} \cdot k_{1000} \cdot k_{0100}) \cdot x^2 + (c_{1110} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} + c_{1101} \cdot k_{1000} \cdot k_{0100} \cdot k_{0001}) \cdot x^3 + c_{1111} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} \cdot k_{0001} \cdot x^4 \quad (8.47)$$

Le signal qui nous intéresse est le nombre d'ions calcium liés. Nous fixons donc le signal molaire relatif à chaque site de liaison à 1. Après avoir posé ces hypothèses sur les constantes, nous réalisons un ajustement des constantes du polynôme avec le modèle de l'ajustement induit et nous obtenons la courbe bleue Figure 8.7.A. Les carrés bleus correspondent aux résultats expérimentaux de Crouch et Klee et le trait au modèle présenté ci-dessus. La qualité de l'ajustement est donnée Table 8.1 avec l'écart quadratique moyen (RMSE) et le coefficient de détermination (R^2) comme indicateur de qualité [141]. Ces chiffres montrent la validité du modèle proposé, ou tout du moins, indiquent que le résultat expérimental obtenu ne permet pas d'infirmer le modèle de Crouch et Klee. Pour aller plus loin, d'autres mesures expérimentales seraient nécessaires pour poursuivre la validation du modèle.

8.4.2 Modèle séquentiel

L'hypothèse du modèle séquentiel est que les ions calcium se lient dans l'ordre des sites. Une fois qu'un ion calcium s'est lié au premier site, un autre ion peut se lier au second site et ainsi de suite jusqu'à ce que les quatre sites soient occupés. Cette hypothèse se traduit, au niveau des coefficients du polynôme de liaison, par les hypothèses suivantes :

- L'affinité du premier site $k_{1,0,0,0}$ est un ordre de grandeur au-dessus de l'affinité du second $k_{0,1,0,0}$, qui est elle-même un ordre de grandeur au-dessus de l'affinité du troisième…
- Le coefficient de couplage entre les sites permet de compenser les faibles affinités des sites 2, 3 et 4. Ainsi, $c_{1,1,0,0}$ est supérieur à 1, $c_{1,1,1,0}$ est très grand devant $c_{1,1,0,0}$ et $c_{1,1,1,1}$ est très grand devant $c_{1,1,1,0}$. Les autres coefficients de couplage sont égaux à 1.

Ces hypothèses permettent de simplifier l'équation (8.46) :

$$P(x) \cong 1 + k_{1000} \cdot x + C_{1100} \cdot k_{1000} \cdot k_{0100} \cdot x^2 + C_{1110} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} \cdot x^3 + C_{1111} \cdot k_{1000} \cdot k_{0100} \cdot k_{0010} \cdot k_{0001} \cdot x^4 \quad (8.48)$$

Après un ajustement avec le modèle séquentiel, les différents coefficients obtenus sont résumés dans la Table 8.1 et la courbe obtenue est présentée Figure 8.7.B. Les résultats de simulation du polynôme y sont comparés avec le modèle séquentiel (trait plein) et les résultats expérimentaux (carrés). Comme pour le modèle précédent, le modèle séquentiel ne peut être infirmé par les

résultats expérimentaux présentés ici et d'autres mesures sont nécessaires pour valider l'hypothèse séquentielle.

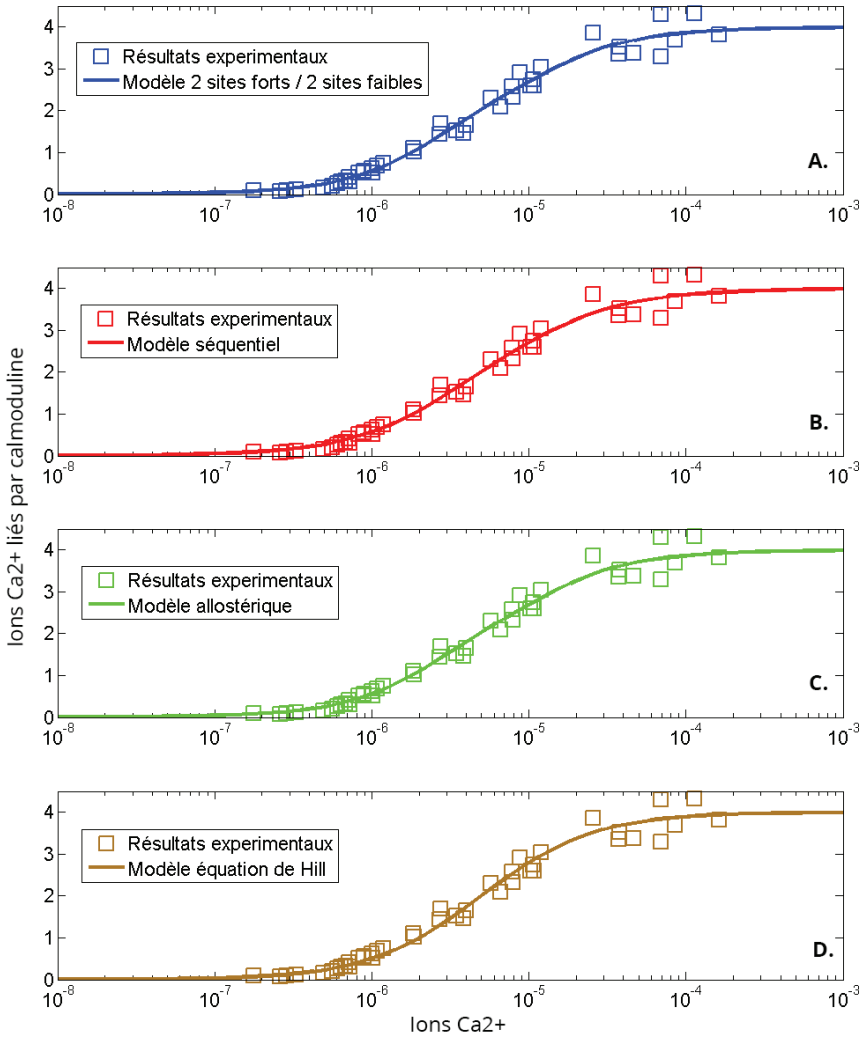


Figure 8.7 : Evolution du nombre d'ions calcium liés par la calmoduline, dépendant de la concentration des ions libres, modélisée à l'aide du polynôme de liaison avec les paramètres Table 8.1 pour un modèle sites forts / sites faibles en A, une approche séquentielle en B, une approche allostérique en C et avec l'équation de Hill en D. La concentration de calmoduline est fixée à 2e-7 Mol comme dans [138].

8.4.3 Modèle allostérique

Il a été démontré précédemment que le modèle allostérique pouvait être considéré comme un modèle d'ajustement induit avec un site supplémentaire pour une molécule d'eau. Pour dériver les équations de ce modèle, un nouveau polynôme est généré avec l'ajout d'un cinquième site pour une molécule d'eau. Les constantes du polynôme sont choisies avec les mêmes hypothèses que pour le modèle à deux sites forts et faibles. Cependant les constantes d'association de

chaque site sont fixées à des valeurs inférieures, mais sont compensées par les facteurs de couplage forts lorsqu'une molécule d'eau est liée. Les éléments du polynôme correspondant aux complexes dans lesquels la molécule d'eau n'est pas liée se réfèrent à la conformation tendue du modèle allostérique et ceux où la molécule est liée à la conformation relâchée. Nous retrouvons les valeurs des autres constantes obtenues après ajustement avec les données expérimentales dans la Table 8.1. Les résultats de simulation du polynôme sont comparés aux résultats de Stefan (Figure 8.7.C). Là encore, le modèle et les résultats expérimentaux coïncident, ce qui permet de dire qu'au regard de cette seule courbe expérimentale, le modèle allostérique pourrait être valide.

Coefficients	2 sites forts / 2 sites faibles	Séquentiel	Allostérique		
k_{1000}	0.03804	1.817	0.0012	Autres Coefficients	
k_{0100}	0.03804	0.1817	0.0012	k_{00001}	4708
k_{0010}	0.001049	0.01817	1.47e-05	C_{10001}, C_{01001}	100
k_{0001}	0.001049	0.001817	1.47e-05	C_{00101}, C_{00011}	1000
C_{1100}	6.255	9.881	5	C_{11001}	2.014e+04
C_{1010}	1	1	1	C_{10101}	1
C_{1001}	1	1	1	C_{10011}	1
C_{0110}	1	1	1	C_{01101}	1
C_{0101}	1	1	1	C_{01011}	1
C_{0011}	1	1	1	C_{00111}	1
C_{1110}	50	237.8	150	C_{11101}	1e+07
C_{1101}	50	1	150	C_{11011}	1e+07
C_{1011}	1	1	1	C_{10111}	1
C_{0111}	1	1	1	C_{01111}	1
C_{1111}	1500	3.489e+04	679.7	C_{11111}	2.562e+10
Coefficient de détermination R^2	0.979	0.9789	0.9789		
RMSE	0.2081	0.2088	0.2205		

Table 8.1: Constantes d'association (k_i) et facteurs de couplage (C_j) du polynôme de liaison pour un modèle à deux sites forts et deux sites faibles, une approche séquentielle et une approche allostérique. La validité de l'ajustement est donnée par le coefficient de détermination et le RMSE.

8.4.4 Lien avec l'équation de Hill

Dans l'exemple de la calmoduline, nous avons quatre ligands identiques. D'après l'hypothèse formulée dans les différentes approches, les sites ne sont pas indépendants. En reprenant l'équation (8.46) du polynôme de liaison et à l'aide de la méthodologie développée en Annexe I.2 nous pouvons approximer le signal total S(x) par :

$$S(x) = \frac{s(x)}{p(x)} = \frac{n}{1 + \left(\frac{K}{x}\right)^n} = \frac{4}{1 + \left(\frac{K}{x}\right)^n} \tag{8.49}$$

avec K et n (qui peut varier de 1 à 4 dans le cas d'une coopérativité positive) les coefficients de Hill. Nous réalisons un ajustement de ces deux paramètres et nous obtenons la Table 8.2 avec les valeurs obtenues, ainsi que la justesse de l'ajustement. Les résultats de simulation sont comparés aux résultats expérimentaux Figure 8.7.D.

Coefficients	Modèle de Hill
K	24.96
n	1.208
R-square	0.9769
RMSE	0.2100

Table 8.2 : Coefficients de l'équation de Hill pour un ajustement sur les données expérimentales mesurées par Crouch et Klee, ainsi que la justesse de cet ajustement.

Nous constatons que la courbe du modèle s'ajuste très bien aux données expérimentales et que la valeur extraite du coefficient de Hill correspond au résultat attendu, à savoir une valeur supérieure à 1, signe d'une coopérativité positive entre les sites.

Ces différents exemples montrent qu'une même courbe expérimentale peut être modélisée de différentes manières avec des ajustements de qualité comparable. Le modèle par ajustement induit est celui qui utilise directement le formalisme que nous avons développé. Il s'agit donc du modèle le plus complet. Il est aussi le plus gourmand en nombre de paramètres et conduit à des modèles sous-déterminés (15 paramètres pour le cas de la calmoduline alors que nous ne disposons pour l'extraction des paramètres que d'une seule courbe dose-réponse). Les hypothèses posées dans le modèle séquentiel ou le modèle allostérique permettent d'ajouter des relations entre les paramètres et de réduire ce nombre de paramètres libres proche de 4 (Table 8.3), qui est la valeur théorique minimale permettant de modéliser la courbe dose-réponse pour un système à 4 sites (voir le polynôme de liaison du modèle macroscopique). Le modèle

optimal semble néanmoins être l'équation de Hill qui n'utilise que deux paramètres et permet d'obtenir un bon coefficient de détermination. Néanmoins, cette conclusion n'est vraie que dans ce cas de figure, compte-tenu du signal observé et dans la mesure où nous ne disposons que d'une seule courbe dose-réponse pour valider les modèles. Elle ne doit donc pas être généralisée et peut ne pas être suffisante dans d'autres cas [142]. En revanche, nous voyons très bien que notre approche de modélisation, complétée par quelques hypothèses sur les mécanismes de liaison, permet de regrouper les différents cas de figure que l'on retrouve classiquement.

Modèle	Nombre de paramètres dans l'approche générique	Nombre de paramètres fixés par hypothèse	Nombre de paramètres restant à extraire
Ajustement induit	15	0	15
Sites fort / faibles	15	10	5
Séquentiel	15	11	4
Allostérique	31	24	7
Equation de Hill			2

Table 8.3 : Nombre des paramètres en fonctions des approches.

8.5 Conclusion

Ce chapitre nous a permis de présenter une méthodologie de modélisation générale pour la liaison de n'importe quel nombre de ligands à une macromolécule possédant un certain nombre de sites. Cette méthodologie permet l'unification de divers modèles microscopiques existants (allostérique, séquentiel, etc.) et permet également de faire des liens avec les modèles macroscopique (Adair-Klotz, Hill, etc.).

Afin d'obtenir facilement ce polynôme associé à un système biologique spécifique, un générateur automatique a été développé sous Matlab. Il fournit directement à l'utilisateur le polynôme après une brève saisie du système. En indiquant les constantes et les signaux associés pour chaque complexe ligand-macromolécule, il permet également à l'utilisateur de simuler la courbe dose-réponse et de la comparer avec les résultats expérimentaux en chargeant les données expérimentales.

Enfin, nous avons montré avec le cas de la liaison des ions calcium à la calmoduline que les mêmes données expérimentales peuvent être modélisées avec le polynôme de liaison en partant d'un modèle d'ajustement induit, pour obtenir un modèle deux sites forts / deux sites faibles, un modèle ordonné et séquentiel ou un modèle allostérique, en fonction des hypothèses sur les

constantes. La mesure expérimentale utilisée pour ajuster les modèles, et les résultats de simulation sont concordantes. A défaut de valider les modèles, ceci permet au moins de ne pas les infirmer. La comparaison à une base de données de résultats expérimentaux plus large serait nécessaire pour affiner les modèles. Dans le cadre du travail de thèse, il est important de noter que l'approche de modélisation présentée ici peut s'imposer comme un standard et est plus précise que les modèles comportementaux utilisés jusqu'ici.

Chapitre 9

Analyse des différents types de bruits biologiques et comparaison avec les bruits électroniques

Jusqu'ici, le caractère aléatoire des signaux biologiques n'a pas été pris en compte dans les modèles présentés. C'est un élément très important car, au niveau cellulaire, les concentrations des espèces peuvent être faibles et la loi des grands nombres ne s'applique plus. La moindre fluctuation peut ainsi entrainer des changements importants dans le comportement d'un biosystème. Par analogie avec l'électronique, nous qualifions donc de bruit l'ensemble des phénomènes physiques aléatoires qui perturbent la réponse nominale des systèmes. Ce bruit est généralement considéré comme néfaste, et, en fonction de ses caractéristiques, peut être vecteur d'une simple pollution du signal ou conduire à un dysfonctionnement complet du système.

En microélectronique, le bruit est un phénomène omniprésent étudié depuis de nombreuses années [143], [144], [145]. Il est quantifié et classifié en fonction de ses caractéristiques selon plusieurs aspects. Compte tenu de la complexité et de la sensibilité au bruit des systèmes microélectroniques actuels, cette problématique est étudiée en amont, lors de la conception des circuits, grâce à des modèles et des outils prédictifs puissants [146], [147], [148], [149]. Les principales sources de bruit sont dues à la nature corpusculaire et au mouvement aléatoire des électrons. La biologie n'échappe pas à la règle. Les phénomènes mis en jeu dans les processus biologiques sont également source de bruit car les espèces biologiques possèdent des propriétés similaires aux électrons (nature corpusculaire et déplacement aléatoire). Le rapport signal sur bruit est souvent plus faible qu'en électronique, à cause de la faible quantité de matière formant le signal. Or la modélisation réalisée jusqu'à présent ne tient pas compte de ces fluctuations car les modèles représentent le comportement moyen d'une population de cellules et non la cellule elle-même. Le bruit est introduit à l'aide de deux modélisations : une approche stochastique, qui n'est pas traitée dans cette thèse, et une approche bas niveau avec l'ajout de bruit aux modèles déterministes, ce que nous proposons dans ce chapitre.

L'objet de ce chapitre est de décrire les bruits présents dans les processus biologiques par analogie avec les bruits connus en électronique. Après un état de l'art sur l'utilisation du bruit par les systèmes biologiques, dans une première partie, nous allons faire un rappel des différents

types de bruit que nous retrouvons dans les systèmes électroniques et de manière générale dans tous systèmes physiques. Dans une seconde partie, nous tenterons d'identifier les sources de bruit dans ces processus biologiques et essayerons d'établir des analogies avec les bruits électroniques. Enfin, dans une troisième partie, nous présenterons la modélisation de ces bruits et l'intégration dans les modèles existants, puis nous finirons par un exemple présentant l'effet du bruit sur un oscillateur biologique.

9.1 Etudes sur le bruit en biologie

A travers l'étude de deux revues consacrées au bruit présent dans les processus biologiques, nous allons nous intéresser aux moyens de le réduire au maximum et à son impact sur des systèmes biologiques.

9.1.1 Limitation de la suppression du bruit

La synthèse et la dégradation des différentes espèces biochimiques (protéines, ligands, enzymes, etc.) se font avec des fluctuations, ce qui engendre du bruit. C'est à partir de ce postulat qu'I. Lestas *et al.* ont réalisé une étude théorique visant la suppression du bruit dans les processus biologiques [150].

Pour mettre en évidence l'influence du bruit, deux espèces X1 et X2 sont étudiées. X1 influe sur la production de X2 qui à son tour intervient dans le contrôle de la production de X1. X1 peut être vue comme de l'ARN messager et X2 comme la protéine synthétisée qui sera son propre activateur.

La majeure partie du système biologique est remplacée par une sorte de « démon » du même type que le démon de Maxwell, utilisé en thermodynamique pour expliquer la régulation de température entre deux réservoirs. Il s'agit d'une sorte de créature intelligente fictive, illustrée Figure 9.1, ayant connaissance des différentes concentrations passées, présentes et futures de X2 et tentant de minimiser la variance de X1. Il correspond ainsi au système biologique idéal le plus optimisé pour réduire les fluctuations.

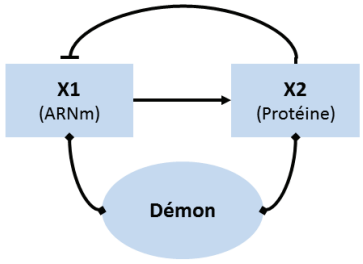


Figure 9.1 : Système biologique composé de deux espèces X1 et X2, régulées par un démon.

L'étude montre que dans le cas de figure évoqué ci-dessus, aucun système de contrôle ne permet la diminution du bruit si X2 est produit à une fréquence inférieure à celle du processus à contrôler. De plus si X1 est produit en rafale (ce qui est le cas pour la production d'ARN messager) le bruit à supprimer est beaucoup plus élevé.

L'ajout de plusieurs systèmes en cascade démontre que le transfert de l'information va être limité par le système présentant le nombre d'espèces le plus faible. L'existence de certains systèmes, comme les plasmides, ne produisant qu'un nombre très faible d'espèces tout en restant très précis, peut ainsi paraître contradictoire.

La seule solution qui ressort de l'étude pour réduire au maximum les fluctuations est l'obligation pour les systèmes de générer des molécules régulatrices à un taux extrêmement élevé. Le bruit est donc bien trop coûteux à supprimer dans un système biologique. L'évolution va au contraire faire en sorte de l'utiliser, comme nous allons le voir dans la sous-partie suivante.

9.1.2 Rôle du bruit dans les circuits génétiques

A. Eldar et M.B. Elowitz se sont intéressés au rôle du bruit dans les circuits génétiques ainsi qu'à certains mécanismes rendus possibles uniquement grâce à sa présence [151]. La régulation génétique est prise comme exemple pour analyser le bruit par le biais de trois postulats.

Le premier part du principe que la traduction d'un ARN messager en protéine se fait en rafale au lieu de se faire à taux constant, ce qui va provoquer des fluctuations aléatoires. Le deuxième est que ces rafales sont moyennées quand la durée de vie d'une protéine est plus longue que le temps entre deux rafales de synthèse de celle-ci. Et enfin le troisième correspond à la propagation du bruit quand l'expression d'un gène est dépendant d'une espèce sujette à une expression en rafale ou au moyennage temporel. De nombreux autres processus doivent aussi avoir un impact important dans la génération de ce bruit, mais restent encore très peu étudiés.

De ces informations nous pouvons distinguer deux types de bruit, le bruit intrinsèque (bruit provenant des rafales) et le bruit extrinsèque (bruit de propagation). Ces différents bruits peuvent jouer un rôle important dans le fonctionnement de certains mécanismes. Il a ainsi été démontré que le bruit intrinsèque peut être utilisé pour générer une régulation coordonnée de plusieurs gènes [151].

Au niveau d'une population de cellules, le bruit autorise aussi une grande gamme de stratégies de différentiation basées sur les probabilités. Dans le cas de systèmes oscillants entre plusieurs états à rétroaction positive, le bruit permet la coexistence de plusieurs états distincts même sans

bi-stabilité du système. Il est aussi requis pour l'initialisation de changements d'état comme dans le cas de la différentiation par procrastination.

En plus de son rôle dans la différentiation, le bruit est indispensable dans l'évolution en augmentant le nombre de phénotypes qui résultent d'un même génotype. Cet effet peut être observé aussi bien dans l'évolution des phénotypes quantitatifs que dans les transitions qualitatives comme celles qui ont lieu dans l'évolution du développement.

Plusieurs autres études ont confirmé la nécessité de la présence du bruit biologique dans le bon fonctionnement de certains mécanismes biologiques [152], [153], [154]. Puisque la suppression du bruit est trop coûteuse et que le bruit est indispensable dans certain cas, il est intéressant d'intégrer des modèles de bruits biologiques dans les modèles existants afin que le comportement des systèmes soit plus proche de la réalité. Pour ce faire nous allons tout d'abord passer en revue les différents types de bruits électroniques et tenter d'identifier leur équivalent biologique afin de proposer des modèles de bruit pertinents.

9.2 Le bruit en électronique

Potentiellement, tout phénomène physique intervenant dans un système peut être source de bruit. Nous distinguons en général les sources de bruit intrinsèques qui sont liés à la physique des composants utilisés pour le système (ex : déplacement aléatoire des électrons) des bruits environnementaux (ex : courants induits par la pollution électromagnétique autour du système). Du fait de la superposition de différentes sources de bruit, il est difficile de les identifier individuellement. En fonction de leurs origines, elles possèdent différentes propriétés et nous pouvons les regrouper en différentes classes.

Les principaux types de bruit que nous retrouvons en électronique peuvent être classés en trois catégories : les bruits d'amplitude, englobant les bruits les plus connus ; les bruits de phase, correspondant à des bruits affectant la fréquence du signal ; et, pour finir, les bruits de quantification intervenant dans des dispositifs où le nombre d'électrons est faible.

9.2.1 Bruit d'amplitude

Les bruits d'amplitude sont les bruits électroniques les plus connus et les plus étudiés. Ils correspondent à des fluctuations aléatoires affectant l'amplitude du signal et sont majoritairement induits par la nature même des composants électroniques.

9.2.1.1 Le bruit blanc

Le premier bruit de ce type est le bruit blanc. Il s'agit d'un signal aléatoire dont la densité spectrale de puissance (DSP) est toujours la même, quelle que soit la fréquence étudiée. Son nom provient de l'analogie avec la lumière blanche qui présente toutes les longueurs d'onde d'émission de la lumière. En pratique, un tel bruit ne peut exister car il possèderait alors une puissance moyenne infinie. Le bruit blanc se limite donc en général à la bande passante du système et se caractérise par une DSP constante dans cette bande de fréquence. Cette propriété se traduit, d'un point de vue temporel, par une décorrélation totale du bruit : la valeur du bruit à un instant *t+dt* est totalement indépendante de la valeur de celui-ci à l'instant précédent *t.* Un bruit blanc est généralement une représentation macroscopique de la somme de plusieurs contributions de bruits décorrélés.

Le bruit thermique est un type de bruit blanc particulier, de nature thermodynamique, possédant une intensité proportionnelle à la température. Il provient de l'agitation thermique c'est-à-dire du mouvement plus ou moins intense des électrons parcourant une résistance électrique (principe du mouvement Brownien). Nous retrouvons donc principalement ce bruit aux bornes des résistances mais il est présent dans tous les dispositifs.

9.2.1.2 Le bruit de grenaille

Le bruit de grenaille ou bruit de Schottky, du nom de son découvreur, est un bruit présent dans chaque jonction entre deux matériaux semi-conducteurs de dopage différent. Il est indépendant de la température mais provient de la nature corpusculaire de la matière. En effet, le courant électrique n'est pas un flux constant mais représente le déplacement des porteurs de charges élémentaires, le plus souvent les électrons. Le nombre de porteurs de charge traversant la jonction ne va pas être constant dans le temps mais va présenter des fluctuations aléatoires, comparables au passage des véhicules sur un tronçon d'autoroute. Sur une courte période, le flux moyen de véhicules peut-être constant mais, à un instant *t*, le nombre de véhicules qui se situent effectivement sur ce tronçon peut varier autour de la valeur moyenne. Le même phénomène se produit avec les électrons en électronique. Ce bruit de grenaille est donc un bruit d'amplitude dont la densité spectrale possède la propriété de varier proportionnellement à la racine carrée du signal.

Nous le modélisons habituellement par une loi de Poisson. Statistiquement, la probabilité de passage d'un nouvel électron à un instant *t* augmente avec le temps *t-t1*, où *t1* est le temps de passage du dernier électron. Cependant, malgré cette loi statistique un peu particulière, il s'agit d'un processus sans mémoire : le temps d'attente entre le passage de deux électrons ne dépend

pas du nombre d'électrons déjà passés par la jonction. De ce fait, le bruit n'est pas non plus corrélé et peut être considéré comme un bruit blanc aux fréquences usuelles en électronique.

9.2.1.3 Le bruit de scintillation

Le bruit de scintillation ou *flicker noise* ou encore bruit en 1/f est un bruit spécifique à la nature même des composants électroniques. Il a pour origine les impuretés présentes dans les matériaux utilisés pour la création des composants électroniques, ainsi que les recombinaisons aléatoires des porteurs de charge dans les paires électrons/trous.

Sa puissance de bruit dépend de l'inverse de la fréquence : ainsi, à faible fréquence de fonctionnement d'un système, sa densité spectrale de puissance est très élevée. Il reste prédominant dans le système jusqu'à une certaine fréquence, appelée fréquence de coude. Au-delà de cette fréquence, il devient négligeable devant les autres bruits du système. De par sa nature, il est souvent associé à des notions de dérives temporelles.

De nombreux phénomènes naturels présentent aussi des fluctuations évoluant selon cette loi en 1/f, comme la hauteur d'eau d'un fleuve, qui, à long terme (à basse fréquence), peut fortement varier en fonction des saisons, mais qui reste plutôt constante à court terme (à haute fréquence).

9.2.2 Bruit de phase

Contrairement aux bruits d'amplitude, les bruits de phase concernent l'aspect temporel des signaux et correspondent plus précisément à une perturbation aléatoire de la phase d'un signal périodique, comme une horloge. Ces fluctuations sont appelées gigue dans le domaine de l'électronique (jitter en anglais). Ces fluctuations altèrent le fonctionnement global d'un système, comme cela pourrait être le cas par exemple pour un orchestre dont le chef donne une mesure de manière plus ou moins aléatoire. Dans un circuit électronique, la gigue d'un oscillateur peut très rapidement introduire de forts bruits d'amplitudes sur les autres signaux.

Pour illustrer ce type de bruit prenons l'exemple de seaux d'eau remplis avec un robinet à flux constant. Si le temps de remplissage de chaque seau varie légèrement, le niveau de remplissage de chacun des seaux sera également différent. Un bruit de phase sur le temps de remplissage des seaux se traduira par un bruit d'amplitude sur le niveau d'eau de chaque seau.

9.2.3 Bruit quantique

Dans la nature, le bruit quantique apparaît lorsque le signal macroscopique est constitué d'un faible nombre de processus microscopiques élémentaires. Nous pouvons comparer ce genre de systèmes à un réservoir rempli par des seaux. Si le réservoir est très grand, le nombre de seaux

pour le remplir sera important et si nous observons le volume global d'eau dans le réservoir, nous aurons l'impression qu'il évolue linéairement. Par contre si le réservoir est petit, il ne faudra que quelques seaux pour le remplir et nous aurons l'impression que le volume d'eau dans le réservoir évolue par paliers.

En électronique, un exemple classique du bruit quantique est le bruit de Barkhausen [155]. Il tire son nom du scientifique qui l'a découvert en 1919. Ce bruit est entre autres présent dans des systèmes faisant intervenir des matériaux ferromagnétiques. Il vient de la structure même de ce type de matériaux, constitué de plusieurs sous-domaines magnétiques appelés domaines de Weiss. Ces domaines sont les plus petites subdivisions d'un matériau ferromagnétique dans lesquelles l'aimantation est homogène. Ainsi, l'aimantation d'un matériau ferromagnétique va se faire par sauts correspondant aux différents domaines de Weiss qui vont progressivement tous prendre la même direction d'aimantation. Le bruit apparaît à ce niveau, avec des fluctuations sur la vitesse de l'aimantation de chaque domaine. De plus, les domaines de Weiss n'ont pas la même taille, ce qui entraîne aussi une disparité dans les sauts d'aimantation qui sont plus importants quand les domaines sont grands et inversement. Le bruit de Barkhausen est prédominant dans les couches ferromagnétiques de taille moyenne car le nombre de domaines de Weiss y est faible.

Autre exemple d'intervention de bruit quantique dans un système électronique : le capteur CCD. A éclairement constant, il provient de la nature corpusculaire des photons. Le nombre de photons tombant sur chaque pixel de la matrice CCD n'est pas le même pendant un même intervalle de temps (ce nombre suit une distribution de Poisson). Cela entraîne un signal bruité qui n'est pas uniforme comme l'a montré Y. Reibel [156].

9.3 Différents types de bruits biologiques

Dans les processus biologiques, le bruit est présent lors de différents mécanismes et sous différentes formes. Nous allons prendre l'exemple de la synthèse de protéines, qui peut être perturbée par différents bruits comparables avec certains bruits électroniques, comme les bruits d'amplitude, de phase et de quantification.

9.3.1 Bruits d'amplitude

9.3.1.1 Bruit blanc

Contrairement au bruit blanc présent dans une résistance, il est difficile de trouver un phénomène aléatoire présentant ce genre de propriétés à l'intérieur d'une cellule. Cependant, certains travaux tendent à montrer que les échanges intercellulaires à travers la membrane

d'une cellule peuvent être représentés par plusieurs sources de bruit différentes, dont une de bruit blanc [157]. Ce cas particulier n'est donc pas à prendre en compte dans les mécanismes que nous avons modélisés jusqu'à présent car ils ne correspondent qu'à des phénomènes qui se déroulent à l'intérieur de la cellule.

9.3.1.2 Bruit de grenaille

Même à l'échelle d'une population de cellules, la nature microscopique de la protéine ne peut être négligée. Ainsi, les caractères temporellement aléatoires des mécanismes de synthèse et de dégradation entrainent une perturbation de la concentration moyenne de protéines pour une population de cellules. Du fait de sa nature, cette perturbation s'apparente au bruit de grenaille que nous connaissons en électronique. Dans le cas précis du processus de synthèse de protéines, le phénomène fait intervenir deux étapes : la transcription d'ARNm à partir d'ADN dans un premier temps, puis la traduction de cet ARNm en protéine dans un second temps. Quatre sources de bruit de grenaille peuvent donc être identifiées, associées à : la synthèse d'ARNm, la dégradation d'ARNm, la synthèse d'une protéine et la dégradation de cette protéine.

Nous retrouvons aussi des sources de bruit de grenaille dans d'autres mécanismes, comme la complexation. En règle générale, chaque mécanisme biologique impliquant une espèce contribue à son propre bruit de grenaille.

9.3.1.3 Bruit en 1/f

Il est possible de retrouver des sources de bruit en 1/f dans les processus biologiques dont les conditions présentent des fluctuations lentes (comme l'évolution du pH ou de la température). Il s'agit alors de bruit extrinsèque (lié à l'environnement et non au mécanisme en lui-même). Cependant, il semble exister des sources de bruit en 1/f au niveau de la membrane des cellules lors du passage des ions [158], et plus particulièrement lors de leur passage à travers la membrane des cellules nerveuses [159]. Ces sources de bruit sont présentes dans des mécanismes dont les modèles n'ont pas été réalisés. Elles ne sont donc pas prises en compte dans cette étude.

9.3.2 Bruit de phase et bruit quantique

Le bruit de grenaille est une manière de modéliser à l'échelle macroscopique la nature corpusculaire des espèces chimiques. Néanmoins, au niveau de la cellule elle-même, cette description n'est plus valable puisque le nombre d'espèces évolue par palier à chaque fois qu'une espèce est créée ou disparait. Cela entraine une quantification des niveaux de concentration.

Si nous reprenons l'exemple de la synthèse de protéines, l'apport d'activateur va se traduire par la synthèse de quelques protéines seulement. Nous allons donc observer, sur la réponse transitoire de la concentration locale de protéine, une évolution par sauts correspondant à la contribution unitaire de chaque protéine.

De plus, du fait du caractère aléatoire du déplacement des protéines au sein de la cellule, celles-ci ne vont pas pouvoir être toutes synthétisées en même temps. Cela va entrainer une fluctuation temporelle de ces sauts de concentration qui est assimilée à un bruit de phase.

9.4 Modélisation

Le bruit est donc un aspect important dans les processus biologiques et, comme en électronique, nous souhaiterions pouvoir anticiper et prédire ses effets sur le fonctionnement des systèmes. Il semble donc indispensable de l'intégrer dans les modèles des mécanismes biologiques déjà développés. Nous pouvons ainsi distinguer trois approches qui permettent d'introduire la notion de bruit :

- l'approche macroscopique développée dans ce chapitre ;
- l'approche microscopique traitée à l'aide d'un simulateur dédié dans le chapitre 10 ;
- l'approche stochastique [160] en cours d'étude, mais qui n'est pas abordée dans cette thèse.

L'approche macroscopique est souvent privilégiée car elle est plus rapide. Elle consiste à modéliser les mécanismes biologiques de manière idéale à l'aide d'équations déterministes, et d'ajouter dans le modèle des fluctuations aléatoires. Elle permet de compléter les modèles bas niveau présentés dans le chapitre 6, en introduisant des sources de bruit correspondant aux types de bruit identifiés, à savoir le bruit de grenaille et le bruit quantique.

9.4.1 Modélisation du bruit de grenaille

Pour modéliser ce bruit nous nous plaçons à une échelle macroscopique et nous appliquons le bruit sur une population de cellules. Pour produire un bruit de grenaille, nous devons tout d'abord générer un bruit blanc. Celui-ci est produit à partir d'un nombre aléatoire x_i, généré à l'aide d'une distribution uniforme dans un intervalle borné à un certain pourcentage de la valeur maximale de la concentration normalisée. Afin d'obtenir un bruit blanc plus proche de la réalité qu'un bruit uniforme, nous transformons ensuite ce bruit en bruit blanc gaussien à l'aide de la transformation de Box-Muller [161] dont la formule est :

$$y_1 = \sqrt{-2 \cdot \ln(x_1)} \cdot \cos(2\pi \cdot x_2)$$
$$y_2 = \sqrt{-2 \cdot \ln(x_1)} \cdot \sin(2\pi \cdot x_2) \quad (9.1)$$

où x_1 et x_2 sont deux nombres aléatoires à distribution uniforme et y_1 et y_2 deux nombres aléatoires à distribution gaussienne. Ce signal est ensuite échantillonné pour produire un signal de bruit blanc à distribution gaussienne.

Le bruit de grenaille dépendant de la valeur du signal, nous multiplions ce bruit blanc par le signal de la concentration de la protéine générée. Ce modèle de bruit de grenaille est ensuite rajouté au signal de la protéine non bruitée. Les résultats de simulation sont présentés Figure 9.2.

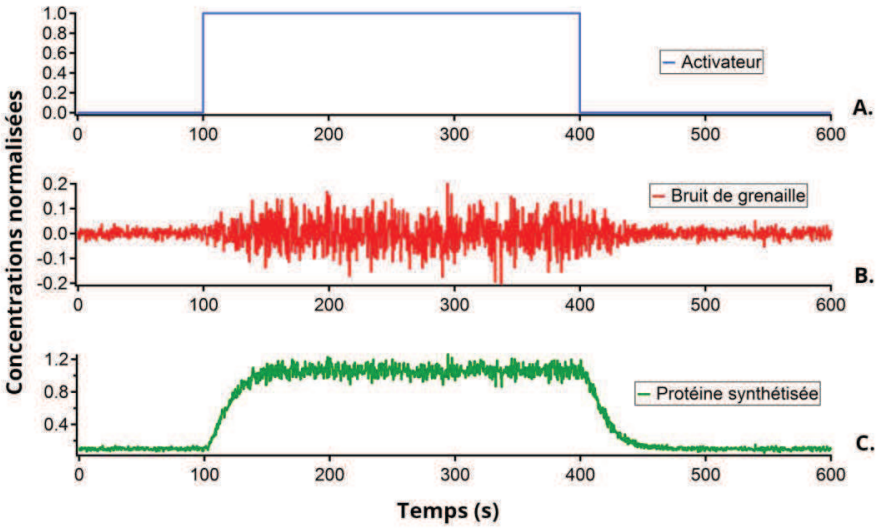


Figure 9.2 : Résultat de simulation du modèle de bruit de grenaille appliqué à une synthèse de protéines. En A. le stimulus de l'activateur, En B. le bruit de grenaille calculé et en C. la concentration de la protéine bruitée dans une cellule.

La courbe en bleu Figure 9.2.A correspond à la concentration d'activateur imposée. La courbe rouge Figure 9.2.B est le modèle de bruit de grenaille, calculé à partir de la synthèse non bruitée. Enfin la dernière courbe en vert Figure 9.2.C intègre ce modèle de bruit de grenaille à la synthèse de la protéine. Nous constatons que ce modèle possède bien une distribution gaussienne et que son amplitude s'adapte à la valeur de la concentration de la protéine.

9.4.2 Bruit quantique et bruit de phase

Le deuxième type de bruit modélisé est le bruit quantique. L'introduction de ce bruit est nécessaire dès lors que nous nous intéressons à un volume où le nombre d'espèces est faible et

donc quantifiable. Les modèles développés jusqu'à présent reposent sur des ODEs donnant le résultat moyen du comportement d'un nombre important d'espèces. Pour obtenir des résultats correspondant à un faible nombre d'espèces, deux approches coexistent : l'approche stochastique qui est plus réaliste mais demande un temps de calcul important, et l'approche consistant à repartir des ODEs et à rajouter au signal des modèles de bruits, qui correspond à une approche plus comportementale. Nous utilisons cette deuxième approche en essayant de régler les paramètres des modèles de bruit pour donner des résultats équivalents à l'approche stochastique.

La première étape a été de quantifier le signal généré par les modèles macroscopiques. Afin de réaliser cette étape, nous avons développé une chaine de quantification représentée Figure 9.3. Cette chaine correspond à un modèle mathématique qui n'a aucune réalité physique ou biologique mais qui est nécessaire pour représenter les mécanismes à une échelle microscopique.

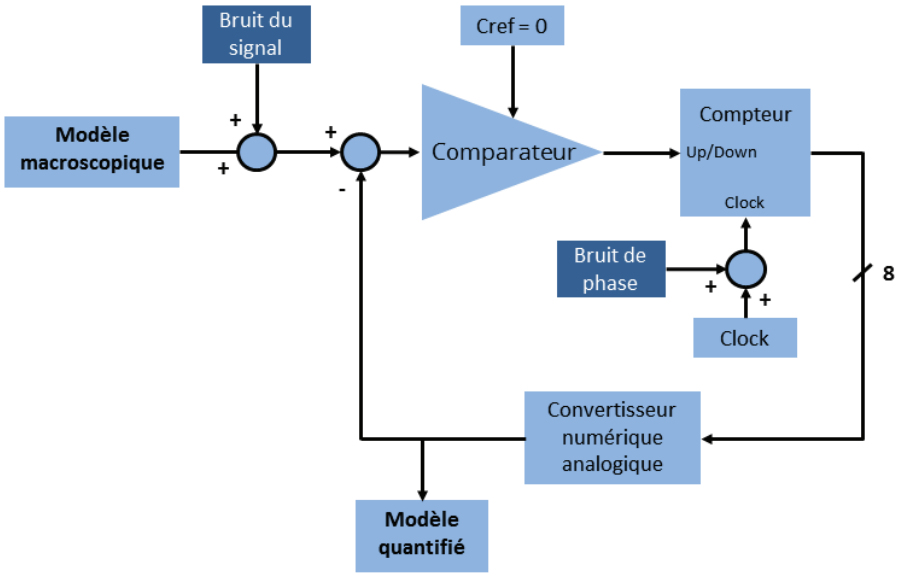


Figure 9.3 : Chaîne de quantification, avec les sources de bruit du signal et de phase.

Le premier bloc, « Modèle macroscopique », correspond au modèle flux de signal du mécanisme de synthèse des protéines présenté au chapitre 6. Les paramètres standards présentés dans ce chapitre sont utilisés pour ce modèle. Ce bloc calcule la concentration moyenne de protéines en fonction du temps sur une population de cellules. La différence entre le signal généré par ce bloc et le signal quantifié est comparée à un seuil « Cref » fixé à 0. Le comparateur va ainsi régler le compteur en mode comptage ou décomptage en fonction du signe du signal en entrée. Le compteur génère un signal codé sur 8 bits correspondant à la valeur quantifiée du modèle macroscopique. Ce signal est finalement retransformé en quantité par le biais d'un convertisseur numérique analogique pour obtenir le modèle quantifié.

Pour ajouter l'aspect temporellement aléatoire des mécanismes de traduction et de transcription, nous ajoutons deux sources de bruit. La première est ajoutée directement au signal provenant du modèle macroscopique, et correspond au bruit du signal. Ce bruit de signal est composé d'une source de bruit de grenaille présentée dans la section précédente, et d'une source de bruit en 1/f, correspondant à une source de bruit blanc filtrée par un filtre passe-bas. La seconde source de bruit est ajoutée sur l'horloge du compteur afin de lui ajouter un bruit de phase pour ne pas avoir une dépendance fréquentielle de l'horloge.

En simulant ce système, nous obtenons le graphique Figure 9.4, présentant la concentration macroscopique de protéines synthétisées en vert, et les protéines quantifiées avec les sources de bruit en rouge.

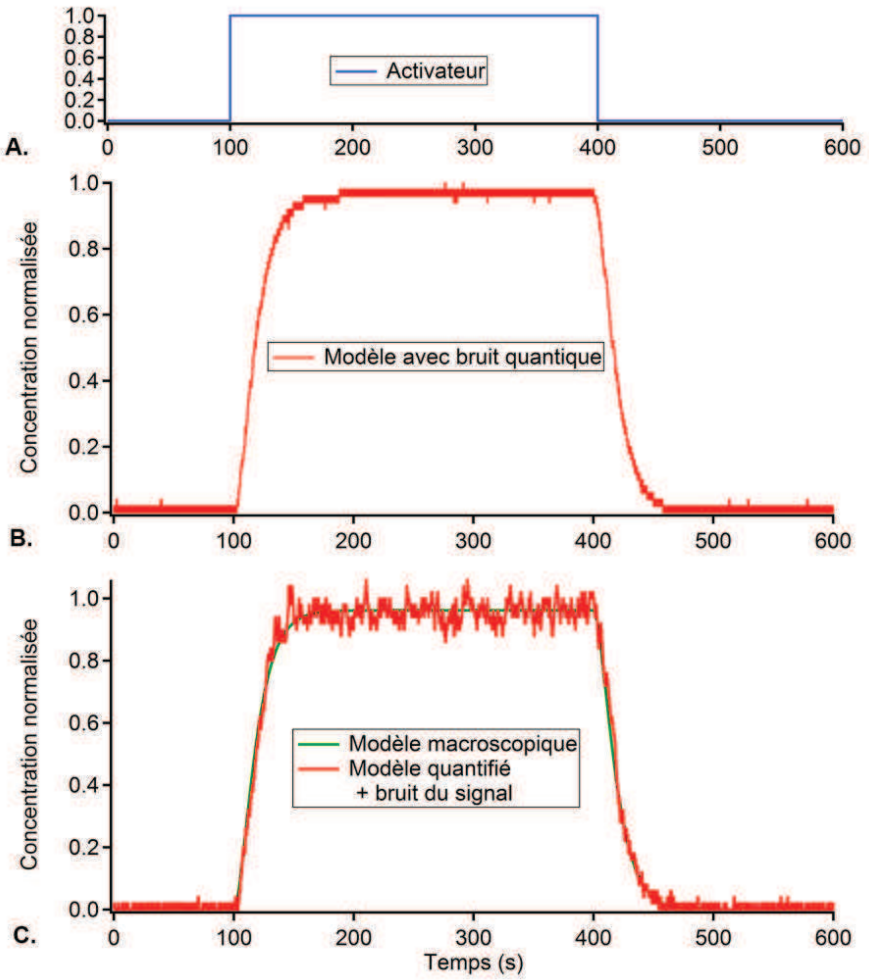


Figure 9.4 : Résultats de simulation des modèles de bruits quantiques et de phase de la synthèse de protéines. En A., la concentration normalisée de l'activateur, en B. le modèle avec le bruit quantique uniquement, et en C., le modèle macroscopique et le modèle quantifié auquel a été rajouté le bruit du signal.

En analysant la distribution du signal du modèle quantifié, une fois le palier de synthèse maximale atteint, nous obtenons la répartition des valeurs Figure 9.5. Nous constatons que nous avons bien une distribution gaussienne des échantillons centrée en 0.96, la valeur du palier générée par le modèle macroscopique.

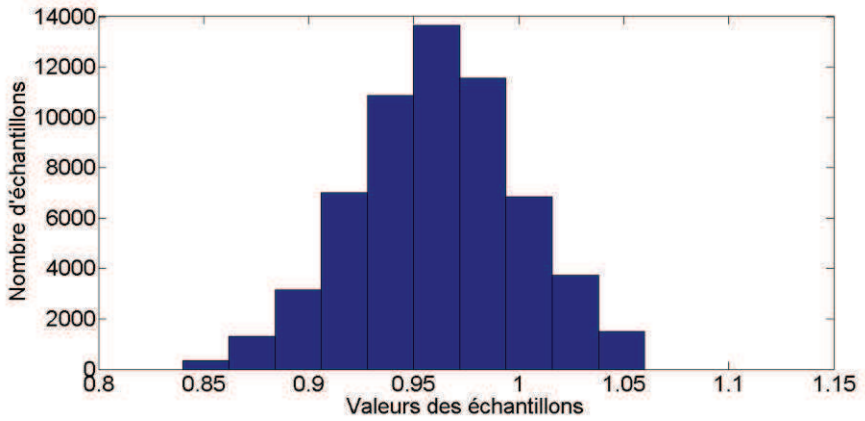


Figure 9.5 : Histogramme du signal du modèle quantifié.

Nous avons également réalisé une étude en fréquence de ce signal. Nous obtenons la répartition de la densité spectrale en fréquence Figure 9.6. Nous constatons bien que la fréquence de base du compteur (fixée à 5 Hz) n'apparait pas dans le spectre fréquentiel, grâce au bruit de phase ajouté à l'horloge du compteur. La densité spectrale obtenue illustre bien la présence du bruit en 1/f provenant de la source de bruit du signal.

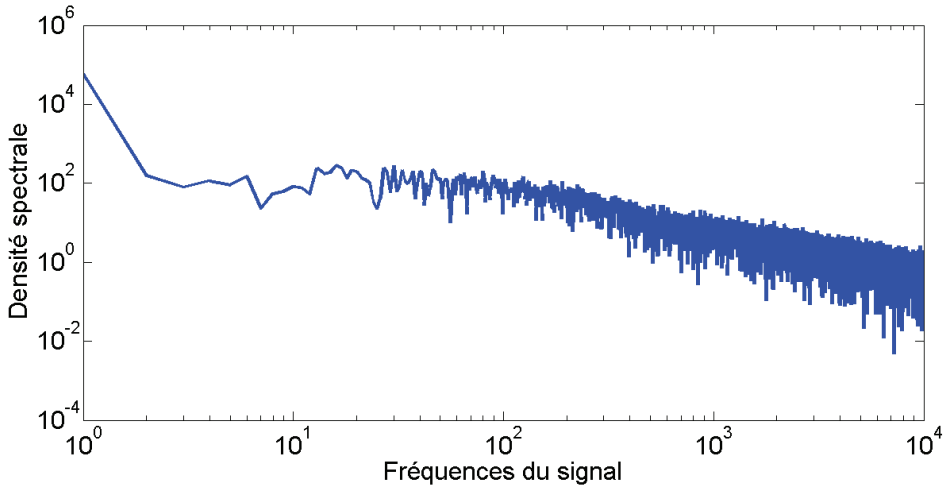


Figure 9.6 : Spectre fréquentiel du signal quantifié.

Le modèle de bruit quantique présenté nous permet d'obtenir des résultats similaires à ceux obtenus pour une approche stochastique. D'après les premiers résultats des modèles stochastiques, il semble plus intéressant, en termes de temps de calcul, de disposer de modèles macroscopiques illustrant le comportement du biosystème, et de rajouter le bruit aux endroits

nécessaires, que de disposer de modèles stochastiques qu'il faut ensuite moyenner pour retrouver le comportement du système.

9.5 Exemples

Afin d'illustrer l'importance de la prise en compte du bruit biologique dans les modèles, nous allons voir deux exemples simples où la présence de bruit modifie fortement le comportement du système. Le premier est le mécanisme de synthèse de protéines et le second est un oscillateur.

9.5.1 Synthèse de protéines

Dans cet exemple, le bruit est ajouté à l'ensemble des espèces contrôlant la synthèse d'une protéine X, à savoir l'activateur et le répresseur du gène codant pour cette protéine, mais aussi à l'ARN messager et à la protéine X. L'ajout des différentes sources de bruit est représenté Figure 9.7.

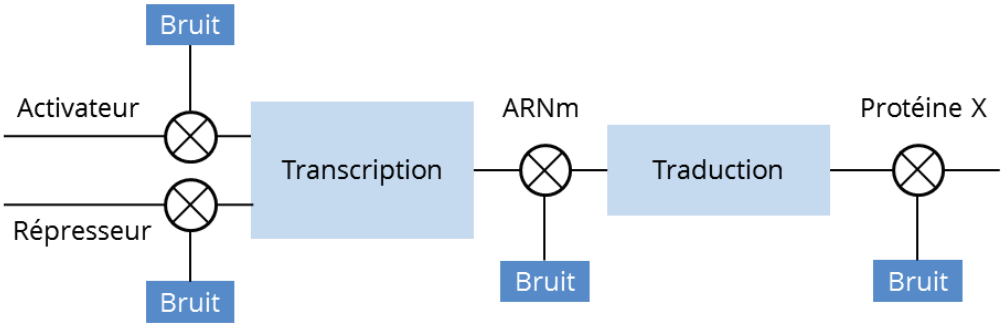


Figure 9.7 : Ajout de sources de bruit dans le mécanisme de synthèse des protéines.

Nous réalisons ensuite les simulations de ce mécanisme avec l'ajout successif des différentes sources de bruit de la Figure 9.7. Les résultats obtenus pour les différents cas de figure sont présentés Figure 9.8.

Nous constatons que l'impact du bruit ne se fait que très peu sentir quand il est appliqué sur l'activateur, sur l'ARNm, ou directement sur la protéine produite. En revanche, pour le répresseur les résultats sont bien différents entre la situation bruitée et non bruitée. Dans le cas idéal, nous supposons que la concentration du répresseur est nulle et que nous avons donc une synthèse d'un niveau constant de protéine X. En introduisant un léger bruit sur le répresseur, nous observons une fluctuation importante sur la concentration de protéine synthétisée.

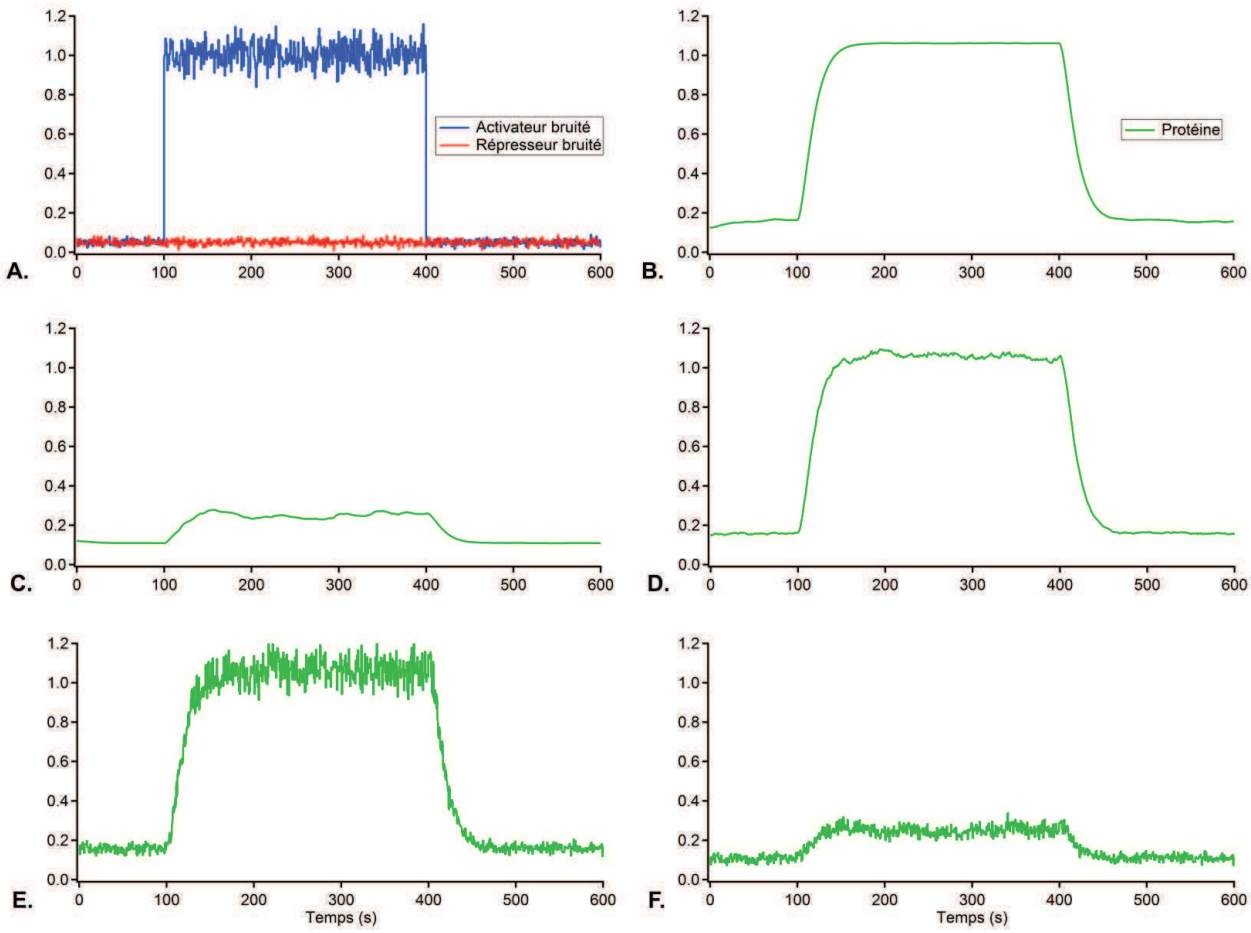


Figure 9.8 : Simulation de la synthèse d'une protéine, avec les différentes sources de bruit de la Figure 9.5, ajoutées au fur et à mesure. En A., l'activateur et le répresseur sont représentés avec les sources de bruit ajoutées. En B., la simulation est réalisée avec l'activateur bruité, en C., avec le répresseur bruité. En D., une source de bruit est ajoutée à l'ARNm et en E. la source de bruit est directement ajoutée à la protéine synthétisée. Enfin en F., toutes les sources de bruit sont ajoutées simultanément.

Cela se produit car nous avons fixé une sensibilité élevée du mécanisme au répresseur (la constante d'association du répresseur avec l'ADN est fixée à $2{\cdot}10^{-3}$). En effet, du fait du bruit et de la forte sensibilité du mécanisme au répresseur, la concentration de protéines est divisée par trois. Si cette protéine synthétisée est elle-même répresseur d'un autre gène, le bruit sera amplifié et le signal sera noyé dans le bruit.

Nous retrouvons un tel phénomène d'amplification du bruit en électronique. En règle générale, nous essayons de mettre en début de chaîne d'amplification des amplificateurs à faible bruit, afin d'éviter d'obtenir un bruit important, que nous amplifions à chaque étage et qui finalement noie le signal.

9.5.2 Oscillateur

Les oscillateurs sont très fréquents en biologie [162], avec de nombreux systèmes naturels alternant entre deux états. Nous allons à nouveau utiliser le mécanisme de synthèse des protéines pour créer un système pouvant osciller à l'aide d'une contre-réaction négative.

Pour obtenir un oscillateur, l'une des solutions est d'utiliser un système constitué d'un gène synthétisant son propre répresseur. A cause du retard introduit par les différents mécanismes biologiques, le système peut se retrouver dans deux états : un état d'équilibre où la concentration de répresseur est juste suffisante pour réguler l'expression du gène, et un état oscillant où nous passons successivement d'une phase d'activité à une autre. Le système peut alterner entre ces deux états en fonction de sa marge de phase. Si cette marge de phase est faible, le système stable aura plus tendance à se retrouver dans un état oscillant que si la marge de phase est élevée, ce qui garantit une plus grande stabilité. Ainsi, la structure biologique Figure 9.9 peut être réalisée pour remplir ces fonctions.

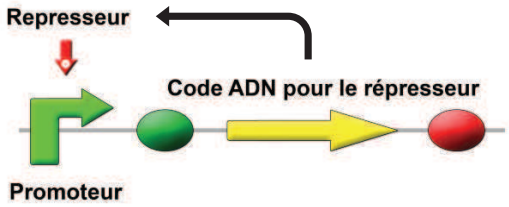


Figure 9.9 : Oscillateur biologique utilisant un seul gène.

Nous nous plaçons initialement dans des conditions où le système joue le rôle de régulateur et n'oscille pas, mais avec une marge de phase très faible. Dans les mêmes conditions, l'ajout de bruit sur la quantité de répresseur synthétisé conduit à un système oscillant, illustré Figure 9.10. L'amplitude du bruit est fixée à 1% de la concentration maximale normalisée.

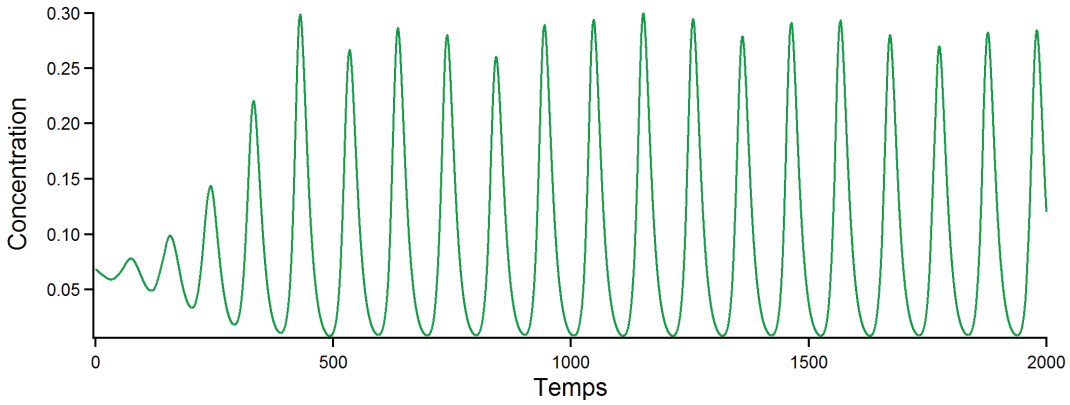


Figure 9.10 : Oscillations constatées grâce à l'ajout de bruit.

Il s'agit d'un phénomène bien connu, ici utile mais le plus souvent gênant. Lorsqu'un système rebouclé est placé en limite de stabilité, le bruit peut suffire à le faire osciller, voire le faire diverger.

9.6 Conclusion

Comme dans la majorité des systèmes électroniques, l'intégration du phénomène de bruit biologique dans la conception de systèmes est vitale afin d'obtenir un système qui soit le plus proche possible du comportement souhaité. Nous avons montré, sur deux exemples précis, que le bruit en biologie peut modifier considérablement le comportement des systèmes.

Ces changements de comportement peuvent être anticipés dès la conception si les modèles de bruits sont précis, comme c'est le cas aujourd'hui pour les circuits électroniques. Cette première étape du travail permet, de manière simple, de tenir compte des sources de bruits les plus connues en biologie. L'intégration du bruit en biologie est d'autant plus importante qu'il n'est pas toujours néfaste, comme en électronique. En effet, certaines études récentes ont montré que le bruit biologique est indispensable à la réalisation de certaines fonctions biologiques, ainsi qu'à l'évolution génétique [151].

Chapitre 10

Simulateur de cellule sur le principe du jeu de la vie

Les études effectuées aux chapitres 8 et 9 montrent que pour obtenir des modèles efficaces, il faut de grandes quantités de données expérimentales sur lesquelles plusieurs mesures sont conjointement accessibles. Or, les mesures de concentration d'espèces chimiques intracellulaires sont complexes et coûteuses. De plus, elles imposent souvent la modification du système testé (en ajoutant des molécules fluorescentes par exemple). Enfin, il est impossible d'effectuer en même temps la mesure des concentrations de nombreuses espèces, ce qui est pourtant indispensable à l'établissement et à la validation des modèles.

Ce chapitre porte sur l'élaboration d'un simulateur de fonctionnement de la cellule qui a pour vocation de simuler le comportement exact des espèces dans la cellule et donc de pallier les difficultés à obtenir une large base de résultats expérimentaux. Ce simulateur repose sur le principe du jeu de la vie : la cellule est subdivisée en territoires sous forme de grille, et les différentes molécules peuvent se déplacer de nœuds en nœuds sur cette grille. A chaque itération, quand deux molécules sont présentes sur le même nœud, des règles probabilistes d'association et de dissociation sont appliquées. Le principe de cette modélisation est élémentaire mais il permet de mettre en évidence des phénomènes biologiques intéressants. Ce simulateur a été développé sous Matlab et les premiers résultats obtenus ont permis des analyses sur l'équation de Hill et plus particulièrement de faire le lien entre les paramètres stochastiques microscopiques (probabilité d'association et de dissociation) et les paramètres macroscopiques du modèle de Hill.

10.1 Fonctionnement du simulateur

Dans la littérature, nous pouvons trouver une étude portant sur la modélisation du fonctionnement d'une cellule à base d'automates cellulaires [163] ainsi qu'une autre étude très complète portant sur les différents outils de simulation de fonctionnement d'une cellule adaptés à la biologie synthétique [164]. Dans cette dernière, les simulateurs sont classés en fonction de la méthode employée pour représenter la cellule. La première approche consiste à attribuer des coordonnées spatiales à chaque espèce. Elle permet d'obtenir des résultats très précis mais présente des temps de simulation extrêmement importants. Dans cette catégorie nous retrouvons les logiciels ChemCell [165], MCell [166] et Smoldyn [167], parmi les plus connus. La

deuxième approche consiste à subdiviser la cellule en une grille de calcul, où sont ensuite réparties les différentes espèces. Cette approche repose sur des calculs probabilistes et est utilisée par les logiciels GridCell [168], Spatiocyte [169] , ou encore ECell [170].

Néanmoins, la plupart de ces logiciels sont basés sur des modèles hauts niveaux de la cellule ou possèdent un fonctionnement trop complexe et ne conviennent pas au but recherché. Nous avons donc développé notre propre logiciel avec un moteur de simulation dédié. Le concept est très simple et ressemble à une sorte de jeu de la vie ou d'automate cellulaire. En premier lieu, la cellule est divisée en un maillage 2D ou 3D, sur lequel les différentes espèces peuvent se déplacer de façon aléatoire. Ces espèces vont se déplacer de nœud en nœud et quand deux espèces se retrouvent sur le même nœud, elles auront une probabilité d'interagir, définie par les paramètres du modèle.

Avec cette première version du simulateur, notre objectif est de démontrer la faisabilité et l'intérêt de notre approche. Pour simplifier l'implémentation et le temps de calcul, nous utilisons un maillage 2D de forme carrée et de taille NxN. Le logiciel est développé sous Matlab, un outil optimisé pour le calcul matriciel, ce qui est très utile pour cette application.

10.1.1 Moteur de simulation

Le programme se présente sous la forme d'une interface graphique (GUI) permettant de saisir la structure de données du problème. Le moteur de simulation va ensuite créer une structure interne de données. A chaque itération du simulateur, il va tout d'abord calculer le déplacement des espèces, les interactions pouvant intervenir entre elles, puis leur dégradation. Enfin, il va mettre à jour la structure de données en intégrant les changements qui viennent d'être calculés. Les différentes étapes de ce fonctionnement sont résumées Figure 10.1.

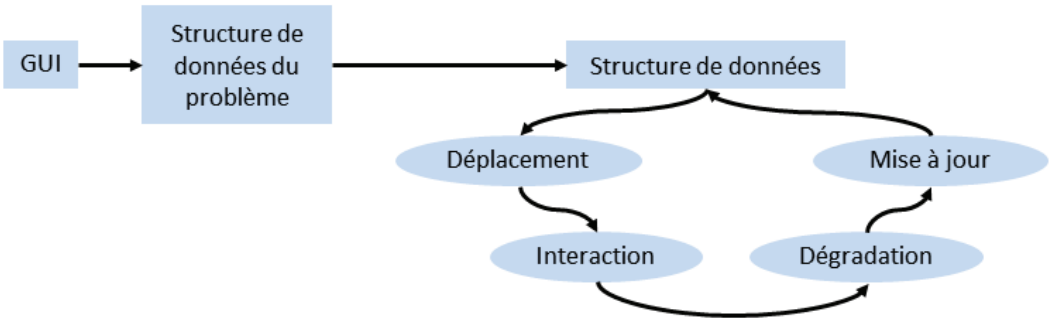


Figure 10.1 : Moteur de simulation du fonctionnement de la cellule.

10.1.2 Structure des données

La première tâche a été de définir la structure des données, ce qui peut être fait par deux approches naturelles. La première approche consiste à définir, pour chacune des espèces en jeu, un vecteur donnant la position de chaque molécule sur la grille ou le maillage. La modélisation du déplacement des espèces à l'aide de cette approche est évidente, mais la modélisation de l'interaction entre plusieurs espèces devient vite compliquée à gérer. Elle exige une recherche itérative dans les différents vecteurs de déplacement des espèces pour trouver les nœuds partagés par deux molécules et ainsi connaître les endroits où une interaction peut se produire.

La deuxième approche consiste à utiliser une matrice pour chaque espèce, de la taille de la maille de la cellule. Chaque élément de la matrice correspond à un nœud de la maille et est égal au nombre de molécules de cette espèce sur ce nœud. La modélisation des déplacements est a contrario plus complexe avec cette méthode. La modélisation des interactions est cependant optimisée, surtout sous Matlab, qui gère très bien le calcul matriciel et l'optimisation des matrices lacunaires.

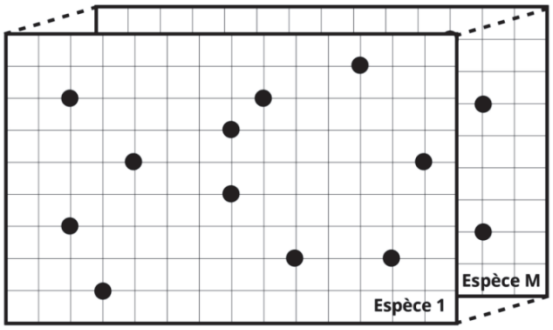


Figure 10.2 : Approche de la structure de données retenue : une matrice de la taille de la maille de la cellule par espèce.

Nous définissons N, le nombre total de mailles, M le nombre d'espèces, et P le nombre moyen de molécules par espèce. Une évaluation comparative de deux méthodes aboutit à la conclusion que la première est plus intéressante lorsque la maille est fine (N important) alors que la seconde est plus intéressante lorsque le nombre d'espèces est important (M élevé). L'espace mémoire nécessaire au stockage des données évolue pour les deux approches en fonction de N, M et P selon les formules de la Table 10.1. Pour la première approche, il faut d'abord calculer les interactions puis réaliser un tri dans les vecteurs de données de taille M·P. Le temps de calcul pour cette approche varie ainsi en fonction de $(M \cdot P)^2$. Pour la seconde approche, le temps de calcul varie en fonction de N·M. La comparaison entre les deux approches est résumée Table 10.1.

Structure de données	Calcul du déplacement	Calcul de l'interaction	Taille des données	Temps de calcul	Critères de choix
Vecteur de position	Facile	Complexe	$M \cdot P \cdot \log_2(N)$	$(M \cdot P)^2$	Maillage de la cellule fin
Matrice de la cellule	Modéré	Facile	$N \cdot M \cdot \log_2(P)$	$N \cdot M$	Nombre d'espèces important

Table 10.1 : Récapitulatif des deux approches pour la structure des données.

Dans cette version du logiciel nous avons décidé d'utiliser la seconde approche illustrée Figure 10.2, ce qui implique que pour chaque espèce M, une matrice de taille X·Y nommée $\mathbf{A_k}$ est créée.

10.1.3 Modélisation du déplacement des espèces

Le déplacement d'une molécule à l'intérieur de la cellule dépend de sa forme et de sa taille. Nous définissons pour chaque type d'espèce un paramètre μ_k de mobilité, variant de 0 à 1. Celui-ci tend vers 0 pour une molécule qui est immobile et vers 1 pour une molécule qui peut se déplacer très rapidement et ainsi changer de nœud à chaque pas de temps. Pour chaque espèce k, le déplacement est ensuite calculé par le biais de la procédure suivante :

- Deux matrices de taille NxN, $\mathbf{RX_k}$ et $\mathbf{RY_k}$, sont générées de manière aléatoire. Leurs termes varient de -1 à 1 selon une distribution uniforme.
- $\mathbf{RX_k}$ et $\mathbf{RY_k}$ sont multipliées terme à terme à la matrice $\mathbf{A_k}$ qui contient des valeurs non nulles uniquement aux nœuds où des espèces sont présentes. Ainsi les deux matrices résultantes contiennent des coefficients aléatoires pour chaque endroit où une molécule est présente.
- Les coefficients de ces matrices sont ensuite comparés à 1 - μ_k. S'ils sont supérieurs, la molécule se déplacera sinon elle restera sur le même nœud.
- Nous créons alors une matrice de déplacement DX_k avec les règles suivantes :

$$\begin{gathered} Si\ A_k(i,j).RX_k(i,j) < -\mu_k et\ i > 1\ alors\ DX_k(i,j) = -1\ et\ DX_k(i-1,j) = +1 \\ Si\ A_k(i,j).RX_k(i,j) > \mu_k et\ i < N\ alors\ DX_k(i,j) = -1\ et\ DX_k(i+1,j) = +1 \\ Si\ |A_k(i,j).RX_k(i,j)| < \mu_k\ alors\ DX_k(i,j) = 0 \end{gathered} \tag{10.1}$$

- Le déplacement des molécules selon l'axe X est mis à jour en effectuant la somme entre $\mathbf{A_k}$ et $\mathbf{DX_k}$.
- Les deux dernières étapes sont réitérées pour le déplacement selon l'axe y en utilisant cette fois-ci la matrice de déplacement $\mathbf{DY_k}$.

Les différentes possibilités de déplacement d'une molécule sont illustrées Figure 10.3. Le déplacement en diagonale survient quand les deux matrices $\mathbf{DX_k}$ et $\mathbf{DY_k}$ entraînent un déplacement de la molécule.

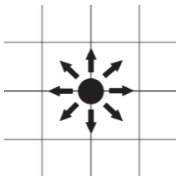


Figure 10.3 : Possibilités de déplacement d'une molécule sur la grille.

10.1.4 Modélisation des interactions entre espèces

La modélisation des interactions est plus complexe à réaliser que celle des déplacements, mais l'utilisation de matrices de la taille de la cellule pour chaque espèce à la place d'un vecteur de position facilite grandement la tâche. Chaque interaction potentielle est traitée séparément en reprenant la méthodologie pour le calcul du déplacement.

Dans un premier temps, il faut déterminer à quels nœuds l'interaction peut se produire. Pour cela, les matrices sont multipliées terme à terme. Un tirage aléatoire de deux matrices, une pour la liaison (**B**) et une deuxième pour la dissociation (**D**) est ensuite réalisé. Le choix de l'interaction est finalement réalisé par comparaison entre ces matrices.

Prenons l'exemple de l'équilibre de la liaison suivante, caractérisée par les paramètres k_{on} (pour la liaison) et k_{off} (pour la dissociation) :

$$P_T + P_U \underset{k_{off}}{\overset{k_{on}}{\longleftrightarrow}} P_V \tag{10.2}$$

Dans notre simulateur, ces coefficients sont remplacés par deux probabilités $PB_{T,U,V}$ et $PD_{T,U,V}$ qui représentent respectivement la probabilité que les espèces P_T et P_U se lient pour former l'espèce P_V quand elles se retrouvent sur le même nœud, et la probabilité que l'espèce P_V se dissocie pour redonner P_T et P_U. Quand deux espèces se retrouvent sur un même nœud, les différents scénarii pouvant intervenir sont illustrés Figure 10.4.

Pour le mécanisme de dissociation, nous recherchons tout d'abord les valeurs non nulles dans la matrice A_V. Cela nous donne les positions sur la maille où les interactions peuvent avoir lieu. Si l'élément de la matrice $D \cdot A_V$ est inférieur à la valeur $PD_{T,U,V}$ pour un nœud, alors la dissociation se produit (la valeur correspondante de la matrice A_V est décrémentée et celles des matrices A_T et A_U incrémentées).

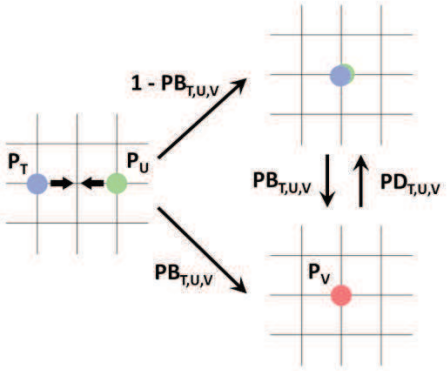


Figure 10.4 : Possibilités d'interactions quand deux molécules se retrouvent sur le même nœud.

Le processus est le même pour la liaison de deux molécules, sauf pour le calcul de la position de l'interaction potentielle qui se fait en réalisant le produit termes à termes des matrice A_T et A_U. Les valeurs de la matrice $B \cdot A_T \cdot A_U$ sont ensuite comparées à la probabilité $PB_{T,U,V}$ et si la liaison se produit, la valeur correspondante de la matrice A_V est incrémentée et celles de A_T et A_U sont décrémentées. Dans la pratique, le calcul des liaisons s'effectuant est réalisé avant celui des dissociations.

10.2 Exemples

Dans cette partie, nous allons montrer le potentiel de notre approche sur la liaison de ligands à une macromolécule, ce qui nous permettra de retrouver certains résultats trouvés précédemment.

10.2.1 Liaison de deux ligands sur une macromolécule

Nous considérons dans un premier temps le mécanisme de liaison entre une macromolécule A et des ligands B présenté par l'équation biologique suivante :

$$A + B \underset{k_{off}}{\overset{k_{on}}{\longleftrightarrow}} AB + B \underset{k_{off2}}{\overset{k_{on2}}{\longleftrightarrow}} AB_2 \tag{10.3}$$

Les espèces A et B peuvent se lier pour former le complexe moléculaire AB, qui peut à son tour se lier avec un ligand B pour former AB_2. Les mobilités μ_A, μ_{AB}, et μ_{AB2}, sont fixées à 0,1 tandis que la mobilité μ_B est fixée à 0,9. Cela illustre bien le fait que le ligand B a une mobilité importante, alors que la macromolécule A et les complexes associés sont beaucoup moins mobiles de par leur taille. Les probabilités de liaison et de dissociation sont respectivement fixées à 0,7 et à 0,003 pour les deux réactions. La cellule est divisée en une grille de 50x50. Ces paramètres sont résumés Table 10.2. L'évolution temporelle des espèces est donnée Figure 10.5.

Paramètres	Valeur
Taille de la Grille	50 x 50
Concentration initiale de A	20
Concentration initiale de B	500
Mobilité de A	0.1
Mobilité de B	0.9
Probabilité d'une liaison A-B par seconde	0.7
Probabilité d'une liaison AB-B par seconde	0.7
Probabilité d'une dissociation A-B par seconde	0.003
Probabilité d'une dissociation AB-B par seconde	0.003

Table 10.2 : Récapitulatif des paramètres du problème simulé.

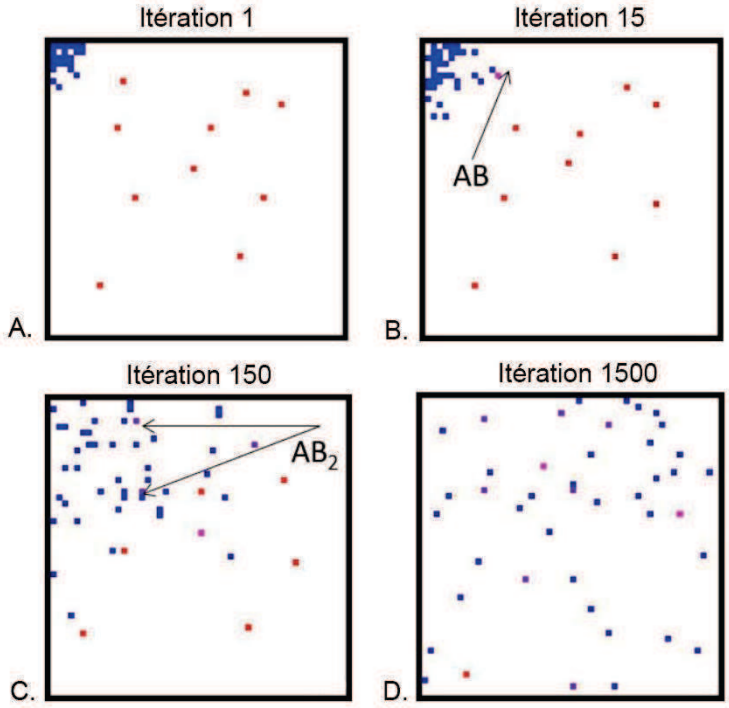


Figure 10.5 : Simulation d'une liaison entre une molécule A et deux ligands B en fonction du temps. Les pixels rouges, bleus, roses et violets correspondent respectivement aux positions de A, B, AB, et AB_2.

Au temps de simulation initial (Figure 10.5.A), 10 molécules de l'espèce A (pixels rouges) sont distribuées aléatoirement sur la grille représentant la cellule, et 50 molécules de l'espèce B (pixels bleus) sont injectées dans le coin supérieur gauche de la grille. Après quelques itérations (Figure 10.5.B), une molécule B se retrouve sur le même nœud qu'une des molécules A, ce qui conduit à la formation du complexe AB (pixels roses). Les molécules continuent de se propager dans la cellule et atteignent d'autres molécules A. Dans le même temps (Figure 10.5.C), une deuxième molécule B se lie avec le complexe AB déjà formé pour donner le complexe AB_2 (pixels

violets). Après environ 1000 itérations (Figure 10.5.D), l'équilibre est atteint, les molécules B étant réparties aléatoirement sur la grille et les quatre espèces A, B, AB et AB_2 coexistant ensemble.

En raison du nombre très faible de molécules, l'évolution temporelle de la concentration de chaque espèce est très bruitée. Cela correspond à la réalité biologique au sein d'une cellule. Nous pouvons voir le résultat de la simulation d'une seule cellule illustrée Figure 10.6.

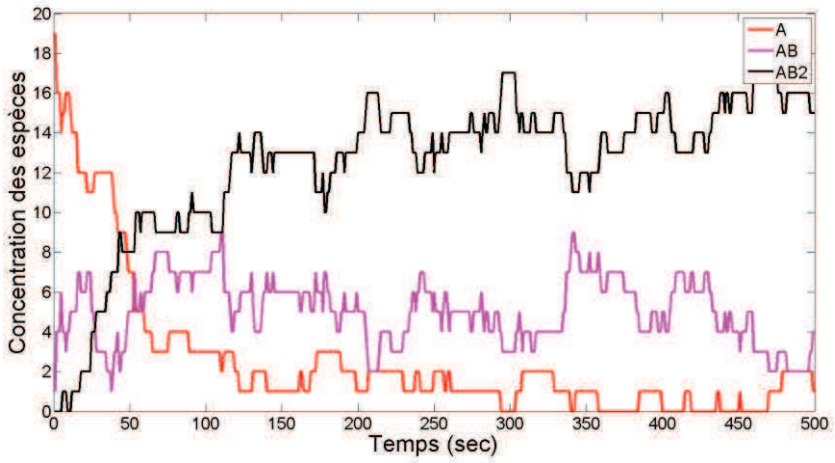


Figure 10.6 : Résultats de simulation d'une cellule pour le problème équation (10.3).

Cependant, pour retrouver les résultats obtenus à l'aide du modèle déterministe, il faut considérer une population de cellules et non une seule cellule. Cela revient à réaliser un moyennage sur plusieurs simulations. Les résultats après moyennage sur 20 simulations sont présentés Figure 10.7.

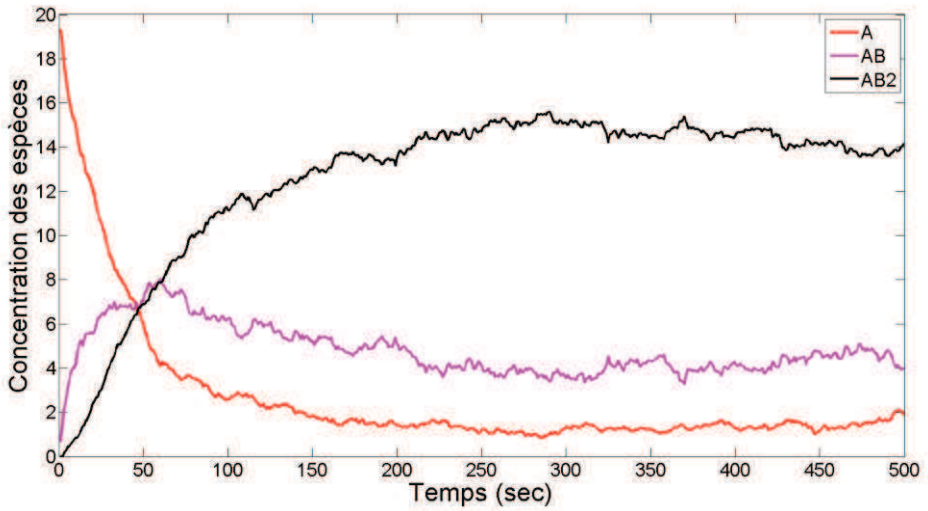


Figure 10.7 : Simulation temporelle du système présenté équation (10.3) montrant l'évolution de la concentration des espèces A, AB et AB2. Ces résultats sont obtenus après moyennage sur 20 simulations.

Nous allons maintenant nous servir des résultats obtenus grâce au simulateur afin de faire le lien entre les différentes modélisations présentées dans les parties précédentes. Dans un premier temps nous allons réaliser l'ajustement des paramètres macroscopiques, puis l'ajustement des paramètres microscopiques du polynôme de liaison, et enfin l'ajustement des paramètres de l'équation de Hill.

10.2.2 Lien avec les paramètres macroscopiques

Pour faire le lien avec les paramètres macroscopiques utilisés dans le modèle déterministe, nous reprenons le système représenté équation (10.3). Nous nous plaçons dans le cas de figure où la concentration de B est très grande devant celles de A, de AB et de AB_2, de sorte à pouvoir supposer que la concentration de B est constante au cours du temps ([A] = 20 et [B] = 500). Nous obtenons ainsi le modèle macroscopique de la réaction, défini par le système d'équations suivant :

$$\begin{aligned} \frac{d[A]}{dt} &= -k_{app1}[A] + k_{off1}[AB] \\ \frac{d[AB]}{dt} &= k_{app1}[A] - \left(k_{off1} + k_{app2}\right)[AB] + k_{off2}[AB_2] \\ \frac{d[AB_2]}{dt} &= k_{app2}[AB] - k_{off2}[AB_2] \end{aligned} \tag{10.4}$$

avec $k_{app1} = k_{on1}[B]$ et $k_{app2} = k_{on2}[B]$. Nous réalisons l'ajustement de ce modèle macroscopique sur les résultats obtenus avec le simulateur. Le résultat de l'ajustement des paramètres est présenté Table 10.3 et illustré Figure 10.8.

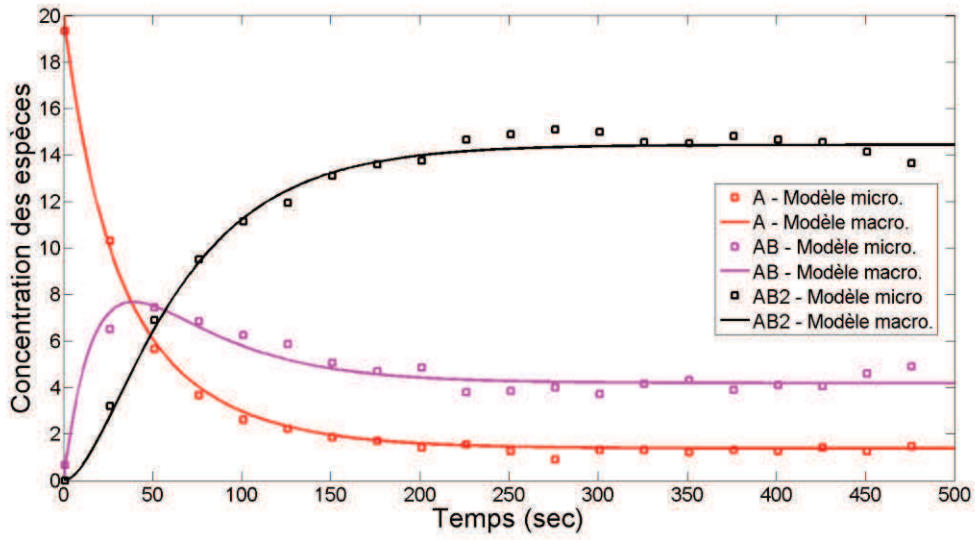


Figure 10.8 : Ajustement du modèle macroscopique à partir des résultats du simulateur.

Paramètres	Valeur
Kapp1	0.3
Kon1	6e-4
Koff1	0.1
Kapp2	0.245
Kon2	4.9e-4
Koff2	0.07

Table 10.3 : Ajustement des paramètres macroscopiques à partir des résultats du simulateur.

10.2.3 Lien avec les paramètres microscopiques du polynôme de liaison

En reprenant l'approche de l'ajustement induit du chapitre 8, nous allons faire le lien entre le polynôme de liaison et les résultats du simulateur. Nous partons des hypothèses suivantes : la molécule A possède deux sites similaires pour les ligands L et il existe une coopérativité positive entre ces deux sites. Les paramètres de probabilité de liaison choisis Table 10.2 pour le simulateur vont dans ce sens : les coefficients de liaison k sont les mêmes pour le premier et le second ligand alors qu'il y a deux fois plus de sites disponibles pour le premier ligand que pour le second. Dans ces conditions, le polynôme de liaison vaut :

$$p(x) = 1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2 \tag{10.5}$$

où x est la concentration de ligand L. A l'aide des simplifications présentées chapitre 8, nous obtenons les différents signaux respectifs de A, AB et AB_2 :

$$S_A(x) = \frac{1}{1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2}$$

$$S_{AB}(x) = \frac{2 \cdot k \cdot x}{1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2} \tag{10.6}$$

$$S_{AB_2}(x) = \frac{c \cdot k^2 \cdot x^2}{1 + 2 \cdot k \cdot x + c \cdot k^2 \cdot x^2}$$

Nous réalisons l'ajustement entre ces signaux et les résultats du simulateur. Les résultats de l'ajustement sont illustrés Figure 10.9 et les coefficients obtenus sont présentés Table 10.4.

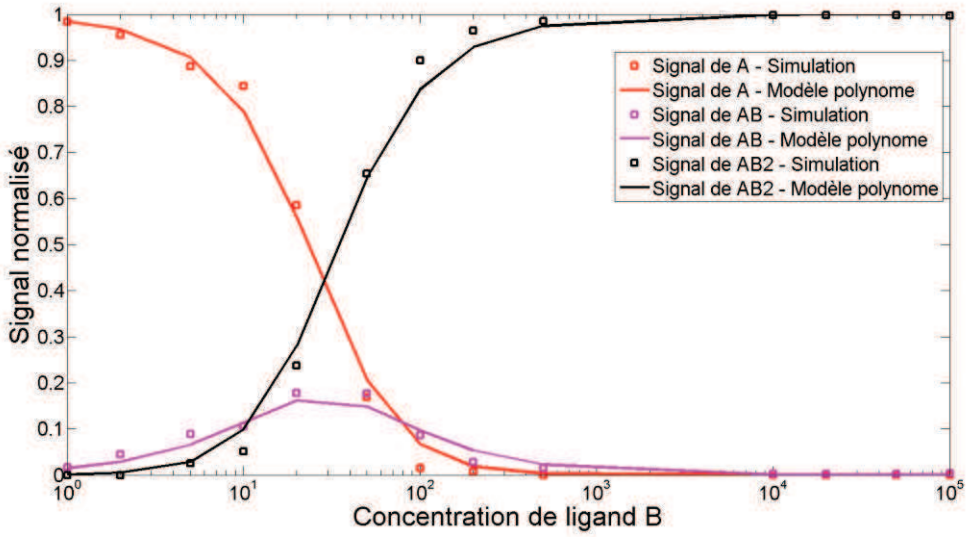


Figure 10.9 : Résultats de l'ajustement entre le polynôme de liaison et les résultats du simulateur.

Paramètres	Valeur
k	0.0072
c	24.1

Table 10.4 : Ajustement des paramètres du polynôme de liaison à partir des résultats du simulateur.

Nous constatons un facteur de couplage c de 24. Nous retrouvons donc dans ces paramètres le couplage coopératif entre les deux sites.

10.2.4 Lien avec l'équation de Hill

Nous avons vu dans les chapitres précédents qu'un système de la forme donnée à l'équation (10.3) peut être approximé par l'équation de Hill de la forme :

$$S_A = \frac{S_{MAX}}{1 + \left(\frac{K}{x}\right)^n} \tag{10.7}$$

Nous avons réalisé l'ajustement de cette équation à partir des résultats fournis par le simulateur. Les résultats sont illustrés Figure 10.10 et l'ajustement des paramètres est donné Table 10.5.

Paramètres	Valeur
S_{max}	2.002
K	28.72
n	1.965

Table 10.5 : Ajustement des paramètres de l'équation de Hill à partir des résultats du simulateur.

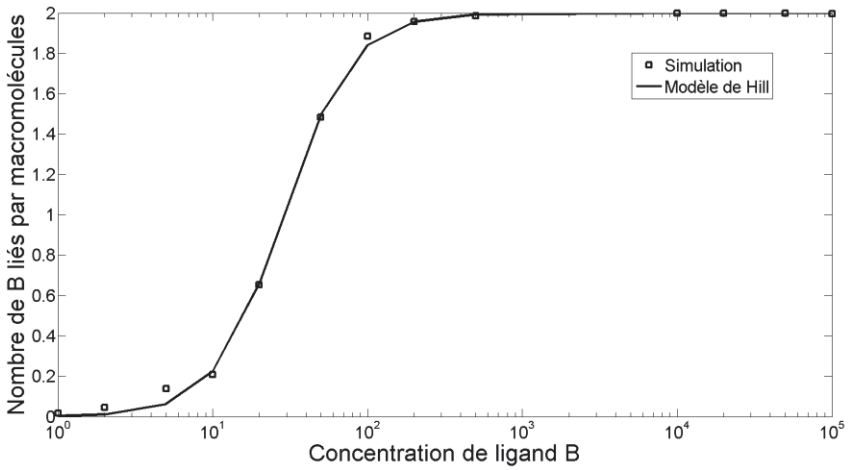


Figure 10.10 : Ajustement de l'équation de Hill sur les résultats du simulateur.

Nous retrouvons, après ajustement, le coefficient de Hill n, proche de 2. Ce paramètre est conforme à nos attentes car nous avons supposé une forte coopérativité.

Pour aller plus loin, considérons maintenant que A peut lier quatre ligands. Nous simulons ce nouveau système avec les paramètres suivants : les valeurs de la mobilité des espèces A et B sont fixées respectivement à 0,95 et 0,05, et 20 molécules A et 500 molécules B sont distribuées de manière aléatoire sur la grille. Les résultats de simulation sont donnés Figure 10.11 pour les trois cas décrits dans la Table 10.6. Ces cas correspondent aux scénarii d'une liaison non-coopérative, d'une liaison avec une coopérativité légèrement positive et d'une liaison avec une coopérativité légèrement négative.

Nous posons $PB_{0,1}$ et $PU_{0,1}$ les probabilités de liaison et de dissociation pour le premier ligand. Dans les cas de liaisons non-coopératives, le premier ligand dispose de quatre sites sur A sur lesquels il peut se lier alors que le $k^{\text{ième}}$ ligand ne dispose plus que de 4-k sites. Par conséquent, nous obtenons la relation suivante :

$$PB_{k-1,k} = \frac{PB_{0,1}}{k} \tag{10.8}$$

où $PB_{k-1,k}$ est la probabilité de liaison de B sur AB_{k-1}. De la même manière, la probabilité que AB_k se dissocie est égale à k fois la probabilité que AB se dissocie et nous avons la deuxième règle :

$$PU_{k-1,k} = k * PB_{0,1} \tag{10.9}$$

où $PU_{k-1,k}$ est la probabilité de dissociation de B de AB_k.

Avec la coopérativité entre les sites, les probabilités $PB_{k-1,k}$ et $PU_{k-1,k}$ augmentent ou diminuent plus rapidement que dans le cas où il n'y a pas de coopérativité. Les courbes représentant S_A en fonction de B obtenues après simulation et ajustement des paramètres avec l'équation de Hill, ainsi que les coefficients de Hill sont donnés Table 10.6.

Cas	Paramètres				Valeur extraite pour n
	$PB_{0,1}$ $PU_{1,0}$	$PB_{1,2}$ $PU_{2,1}$	$PB_{2,3}$ $PU_{3,2}$	$PB_{3,4}$ $PU_{4,3}$	
Cas 1: 4 ligands sans coopérativité	0.3 0.003	0.15 0.006	0.1 0.009	0.075 0.012	1.045
Cas 2: 4 ligands coopérativité pos.	0.3 0.003	0.3 0.003	0.3 0.003	0.3 0.003	1.89
Cas 3: 4 ligands coopérativité neg.	0.3 0.003	0.1 0.009	0.044 0.020	0.022 0.041	0.92

Table 10.6 : Paramètres du simulateur et coefficients de l'équation de Hill correspondante.

Cette étude nous a permis de confirmer les propriétés du coefficient de Hill ayant déjà été observées, à savoir :

- n est d'environ 1 pour un cas non coopératif (cas 1),
- n augmente dans le cas d'une coopérativité positive (cas 2) et diminue dans le cas d'une coopérativité négative (cas 3).

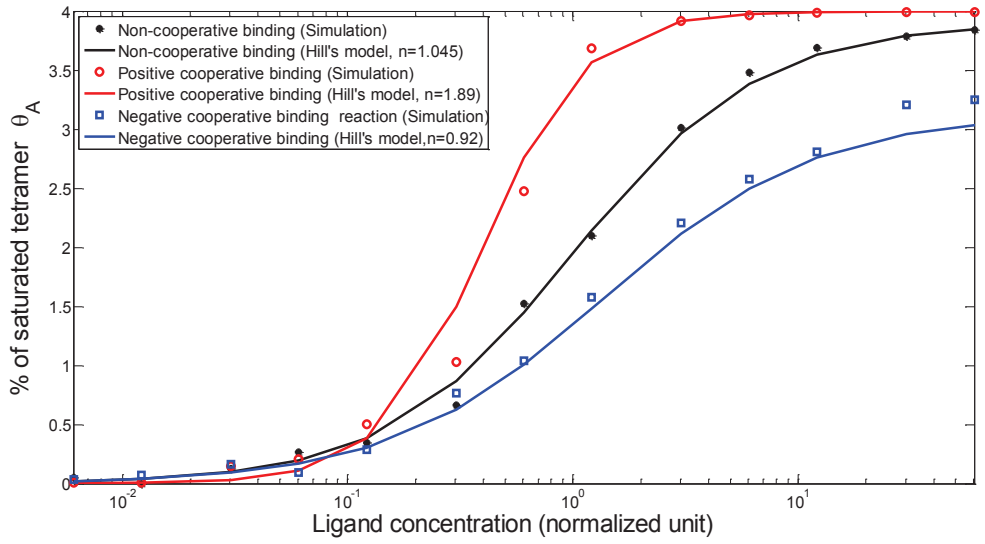


Figure 10.11 : Pourcentage du tétramère S_A à l'équilibre en fonction de la concentration initiale de ligand. Les marqueurs correspondent aux résultats de simulation et les lignes représentent le modèle de Hill correspondant. Dans les cas d'une liaison sans coopérativité, K_A a été choisi comme la concentration de référence pour les abscisses.

10.3 Conclusion

Le travail présenté dans ce chapitre correspond à une première version d'un nouveau simulateur que nous avons développé, et qui devrait fournir des jeux de données complets et exhaustifs, nécessaires à l'élaboration de modèles orientés conception pour la biologie synthétique. Il devrait faciliter l'établissement des liens entre les paramètres mathématiques d'une modélisation macroscopique (comme l'équation de Hill) et les paramètres biochimiques (l'inertie d'une molécule, la probabilité de liaison entre deux espèces, la dégradation des espèces...). Le logiciel a fait ses preuves sur deux exemples, correspondant au deux mécanismes biologiques sur lesquels le travail de modélisation est axé. Néanmoins, certaines améliorations doivent être effectuées pour la prochaine version du logiciel, comme l'utilisation d'un maillage en 3D et la comparaison avec un maillage 1D, l'utilisation de mailles plus complexes (en permettant, par exemple, que la finesse du maillage soit paramétrable en fonction des zones), la simplification de l'ensemble des paramètres du simulateur ou encore l'introduction de l'anisotropie pour le déplacement des espèces. Jusqu'à présent, le problème majeur rencontré dans le développement et l'utilisation de ce simulateur est le temps de calcul (environ 1 minute pour simuler le système présenté dans la Section 4 avec 2000 itérations). Des solutions pour l'exécution rapide de ces algorithmes (calcul sur GPU ou sur FPGA) doivent aussi être explorées.

Au niveau de l'exploitation des résultats, le simulateur nous a permis de faire le lien avec les modèles compacts utilisés dans le flot de conception proposé, et nous permet ainsi de disposer d'une base d'un équivalent de simulateur par éléments finis pour la biologie. Quand il sera plus abouti, ce simulateur devrait nous permettre de pouvoir valider les modèles sans passer par des résultats expérimentaux et ainsi disposer d'expériences virtuelles complètement contrôlables et dont les paramètres peuvent être extraits facilement.

Conclusions et perspectives

Depuis les débuts de la biologie synthétique dans les années 2000, la complexité des systèmes n'a fait qu'augmenter, pour passer de simples oscillateurs ou de fonctions logiques élémentaires comme des portes ET, à des systèmes entiers constitués d'une petite dizaine de gènes, pouvant servir à des fonctions biologiques (détection de certaines combinaisons d'espèces par exemple). Cette évolution est bien sûr liée au nombre de plus en plus important d'équipes qui travaillent sur ce domaine. La preuve en est que le concours iGEM a vu son nombre de participants grandement augmenter au cours de ces dernières années. Paradoxalement, le nombre de BioBriques développées chaque année par les équipes est en baisse, mais elles sont de mieux en mieux caractérisées, ce qui permet une réutilisation plus importante et une meilleure prédiction de leur comportement. Les domaines d'application sont aussi en pleine expansion avec le développement futur de nouveaux carburants ou de nouveaux médicaments, la synthèse de molécules thérapeutiques grâce à des biosystèmes, la recherche de pathologies voire le remplacement de fonctions biologiques défaillantes ou inexistantes dans le vivant.

L'interface entre les sciences pour l'ingénieur et la biologie synthétique permet d'allier l'expertise de ce domaine en conception aux connaissances nécessaires à l'élaboration des biosystèmes. Le travail de cette thèse constitue ainsi une contribution importante dans la structuration et l'automatisation des étapes de conception pour les biosystèmes synthétiques. Il a permis de tracer les contours d'un flot de conception complet, à partir d'une approche novatrice (en comparaison avec les approches existantes présentées dans l'état de l'art du chapitre 2) issue de la microélectronique, et de mettre en évidence ses intérêts. Les différentes étapes de ce flot de conception présenté dans le chapitre 4 sont reprises dans la figure ci-après, avec en bleu, les parties développées pendant cette thèse, et en rouge les éléments qui restent à élaborer.

Les deux principales contributions de ce travail concernent les outils en amont, permettant de passer de spécifications haut-niveau à des spécifications situées au niveau des BioBriques à implémenter, ainsi que les éléments liés à la modélisation. Le premier apport (Figure CCL.1), décrit au chapitre 5, exploite l'abstraction numérique (dont les détails sont présentés au chapitre 3 en même temps que les mécanismes élémentaires), afin de réutiliser directement deux logiciels libres de synthèse logique, utilisés dans la conception de circuits électroniques numériques (Odin II et ABC). L'efficacité de cet outil est illustrée sur la conception de deux systèmes biologiques fictifs mais dont la complexité peut s'étendre bien au-delà de ce qui est réalisable avec les technologies actuelles.

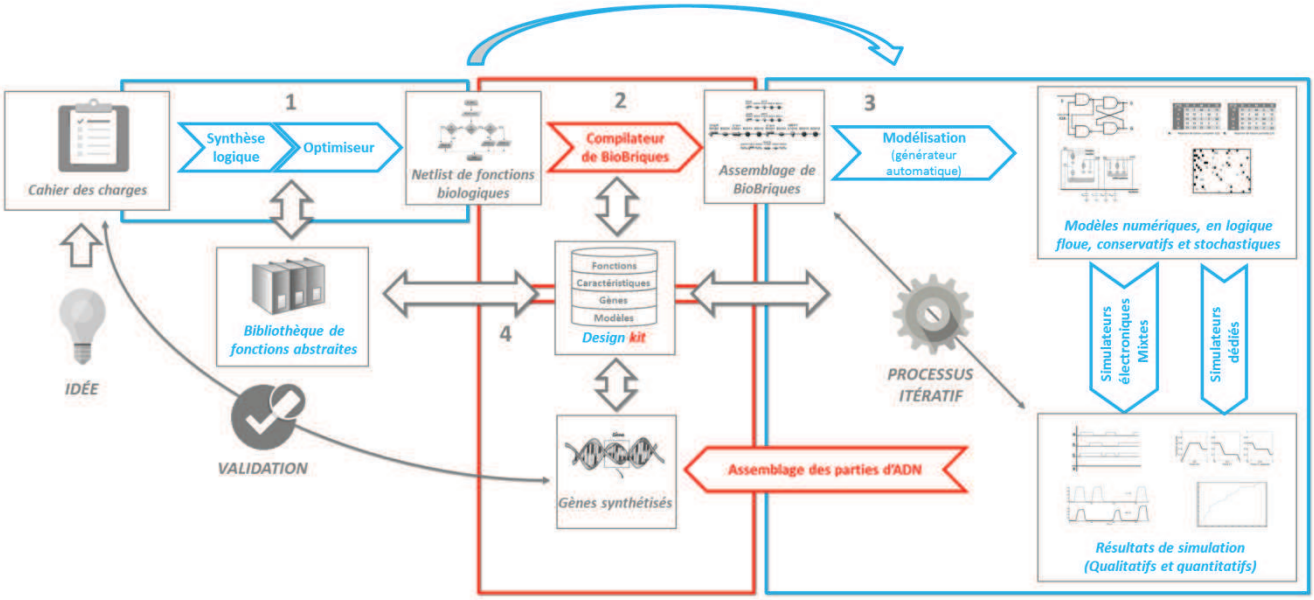


Figure CCL : Résumé du flot de conception proposé, présenté en détail dans le chapitre 4.

L'autre contribution majeure de ce manuscrit (Figure CCL.3) concerne les travaux de modélisation. Dans un premier temps, nous avons modélisé les mécanismes biologiques à l'aide des deux niveaux d'abstraction traditionnellement utilisés par les outils de la microélectronique : l'abstraction numérique, principalement utilisée pour les phases de synthèse en amont, et l'abstraction analogique conservative, utilisée pour la validation des systèmes. Un formalisme spécifique a été développé (présenté au chapitre 6) afin de pouvoir exprimer les modèles biologiques avec les langages de description matériel, comme le VHDL-AMS et le SystemC-AMS. Ces langages ont été choisis car ils sont intégrés au flot de conception de la microélectronique et pris en charge par les outils associés. Le formalisme étant relativement lourd, en particulier pour des utilisateurs qui ne sont pas habitués à ce langage, nous avons également mis au point un générateur automatique de modèles et proposé une netlist standard permettant de faire le lien entre les différents outils.

Les deux niveaux d'abstraction s'étant avérés trop éloignés, nous avons réfléchi à la description des systèmes à un niveau intermédiaire, et l'approche de modélisation en logique floue nous est apparue comme la plus pertinente (elle est développée au chapitre 7). Elle constitue un bon compromis, dans la mesure où le système y est décrit avec un nombre limité de règles (ce qui est avantageux pour les étapes en amont) mais où elle fournit quand même des résultats quantitatifs fiables (qui sont utiles pour la validation). A ce niveau, une représentation des mécanismes biologiques en logique floue a été mise au point et un simulateur associé a été développé sous Matlab.

Enfin, la dernière partie de ce manuscrit a été entièrement consacrée à l'amélioration des modèles. En effet, les modèles utilisés jusque-là étaient principalement des modèles macroscopiques, indiquant le comportement moyen d'un organisme dans une population. L'amélioration des modèles passe donc par la modélisation microscopique. Cela a été fait dans un premier temps au niveau du mécanisme élémentaire en biologie : la complexation entre deux molécules (étudiée au chapitre 8). A ce niveau, un formalisme restant compatible avec une modélisation conservative a été établi. Nous avons notamment démontré que ce formalisme était suffisamment générique pour englober toutes les approches classiques de modélisation des mécanismes biologiques au niveau macroscopique. Ce niveau de modélisation peut s'avérer très efficace et permet notamment de rendre compte de comportements assez particuliers et impossibles à intégrer dans les modèles standards. Il reste néanmoins deux autres effets à prendre en compte dans les modèles, ayant pour conséquences principales l'apparition de bruits importants et la discrétisation de certains signaux. Ces deux problèmes ont été abordés dans les chapitres 9 et 10 avec deux approches différentes. Au chapitre 9, nous avons opté pour une approche consistant à repartir des modèles idéaux et à les dégrader par des sources de bruits, jusqu'à ce que les signaux biologiques ressemblent le plus possible aux signaux réels (obtenus par mesure ou par des simulateurs microscopiques). Au final, nous avons identifié quelques sources de bruits aux propriétés et aux caractéristiques différentes, mais très semblables à des sources de bruits tels qu'il en existe en électronique. Au chapitre 10, nous sommes partis du comportement élémentaire des molécules pour réaliser un simulateur dédié. Celui-ci permet de suivre le déplacement et les interactions entre les protéines au sein de la cellule, ce qui est essentiel à la mise au point de bon modèles comportementaux. Son principal défaut reste le temps de calcul.

Pour poursuivre dans cette voie, l'une des principales perspectives pour ce projet serait la validation des modèles par des expériences (Figure CCL.4). Le problème est qu'il n'existe actuellement que très peu de méthodes de mesure permettant d'accéder simultanément à la mesure de la concentration de plusieurs espèces différentes au sein du même milieu. Le développement de méthodologies d'extraction des paramètres d'un biosystème devra être envisagé. Elles pourront se baser sur des méthodes telles que la mesure de fluorescence par micro-fluidique. Ces paramètres auront pour vocation, à terme, d'être intégrés aux design kits pour les raisons évoquées ci-dessus.

Le lien entre notre flot de conception et les langages de description biologique, comme le SBML et le SBOL, est lui aussi en cours d'étude. Cela éviterait aux bio-ingénieurs de devoir utiliser le VHDL-AMS ou le SystemC-AMS pour décrire leurs systèmes. De plus, les logiciels de biologie synthétique les plus employés utilisent pour la plupart ce formalisme. Le développement de ces interfaces ouvrirait donc la voie à l'utilisation d'outils existant, mis au point par la communauté

biologique, pour exécuter certaines tâches du flot de conception, comme le compilateur de BioBriques (Figure CCL.2).

Les modèles développés couvrent les différents niveaux d'abstraction possibles, allant de l'abstraction numérique à la modélisation bas niveau en passant par la logique floue. Néanmoins, ils concernent tous une approche macroscopique. Pour la modélisation microscopique, nous disposons de peu d'éléments. Les approches présentées dans les deux derniers chapitres ont des résultats convaincants mais un certain nombre de limites ont néanmoins été mises en avant. Pour répondre à ce besoin, des modèles stochastiques reposant sur les équations différentielles sont en cours de développement et permettront de confirmer les modèles de bruit développés. Le simulateur quantique donne déjà des résultats intéressants mais possède un temps de calcul trop important. La nouvelle approche à l'étude est donc de s'orienter vers des ODEs stochastiques.

Enfin, toujours au niveau de la modélisation, les modèles développés jusqu'ici concernent principalement la modélisation des mécanismes biologiques intracellulaires. Cependant, un certain nombre de travaux récents montrent également le potentiel des biosystèmes synthétiques constitués de plusieurs micro-organismes reprogrammés différemment et qui échangeraient entre eux des informations sous la forme de protéines. Cette approche permettrait la mise au point de systèmes plus complexes avec des circuits génétiques plus simples (donc plus réalistes du point de vue de la réalisation). Il faut donc envisager à court terme l'extension de nos formalismes à ce type de biosystèmes en donnant une possibilité de vue hiérarchique, en plus de la vue modulaire mise au point dans cette thèse.

La biologie synthétique est une science en plein essor où de nombreuses découvertes restent à faire. Beaucoup de domaines peuvent bénéficier de ses retombées, et notamment la santé, l'environnement et l'agroalimentaire. Il me semble donc indispensable de combiner les connaissances et le savoir-faire des différentes sciences pour faire avancer la recherche dans ce domaine. Les compétences des microélectroniciens en conception de systèmes sont particulièrement avancées car elles bénéficient de longues années d'expérience. Elles seraient donc un atout important pour l'élaboration de biosystèmes synthétiques complexes. L'évolution constante des techniques employées pour l'élaboration de systèmes biologiques reste évidemment une nécessité pour arriver à réaliser des systèmes de plus en plus complexes. Enfin un travail de présentation et de vulgarisation de la biologie synthétique est impératif afin de mettre en lumière cette science innovante encore méconnue du grand public.

J'espère ainsi que le travail fourni durant cette thèse servira de base à de futurs travaux et contribuera, à son échelle, au développement de la biologie synthétique qui s'annonce, sans nul doute, comme un domaine de recherche de pointe des plus prometteurs.

Bibliographie

[1] "Synthetic Biology Engineering Research Center (Synberc)," *http://www.synberc.org/*.

[2] "SynBiology - An Analysis of Synthetic Biology Research in Europe and North America," *FP6-2003-NEST-B4 Project 015357*, no. September, 2006.

[3] "Biologie de synthèse," *http://www.biologiedesynthese.fr/*, 2011.

[4] "Synthetic Biology: Emerging Global Markets," *BBC Research*, 2011.

[5] "Synthetic Biology Market - Global Industry Analysis, Size, Growth, Share And Forecast 2012 - 2018," *Transparency Market Research*, 2012.

[6] S. Leduc, "Théorie physico-chimique de la vie et générations spontanées," 1910.

[7] S. Leduc, *La biologie synthétique*. 1912.

[8] J. Loeb, *The Mechanistic Conception of Life. Biological Essays*. The University of Chicago Press, 1912.

[9] W. Szybalski, "In Vivo and in Vitro Initiation of Transcription," *Advances in Experimental Medicine and Biology*, vol. 44, pp. 23–24, 1974.

[10] W. Szybalski and A. Skalka, "Nobel prizes and restriction enzymes," *Gene*, vol. 4, no. 3, pp. 181–182, 1978.

[11] S. N. Cohen, A. C. Y. Chang, H. W. Boyert, and R. B. Hellingt, "Biologically Functional Bacterial Plasmids In Vitro," *Proceedings of the National Academy of Sciences USA*, vol. 70, no. 11, pp. 3240–3244, 1973.

[12] R. K. Saiki, D. H. Gelfand, S. Stoffel, S. J. Scharf, R. Higuchi, G. T. Horn, K. B. Mullis, and H. A. Erlich, "Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase.," *Science*, vol. 239, no. 4839, pp. 487–91, Jan. 1988.

[13] A. M. Maxam and W. Gilbert, "A new method for sequencing DNA," *Proceedings of the National Academy of Sciences USA*, vol. 74, pp. 560–564, Jan. 1977.

[14] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the National Academy of Sciences USA*, vol. 74, no. 12, pp. 5463–5467, 1977.

[15] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators.," *Nature*, vol. 403, no. 6767, pp. 335–8, Jan. 2000.

[16] T. Knight, “Idempotent Vector Design for Standard Assembly of Biobricks Standard Biobrick Sequence Interface,” *MIT Artificial Intelligence Laboratory*, pp. 1–11.

[17] D. Endy, “Foundations for engineering biology.,” *Nature*, vol. 438, no. 7067, pp. 449–53, Nov. 2005.

[18] J. Peccoud and M. Isalan, “The PLOS ONE Synthetic Biology Collection : Six Years and Counting,” vol. 7, no. 8, 2012.

[19] J. Haiech, R. Ranjeva, and M. Kilhoffer, “Biologie des systèmes et ingénierie biologique modifient la découverte et le développement des médicaments,” *Médecine/Science*, vol. 27, pp. 207–212, 2011.

[20] C. M. Fraser, J. D. Gocayne, O. White, M. D. Adams, R. A. Clayton, R. D. Fleischmann, C. J. Bult, A. R. Kerlavage, G. Sutton, J. M. Kelley, R. D. Fritchman, J. F. Weidman, K. V Small, M. Sandusky, J. Fuhrmann, D. Nguyen, T. R. Utterback, D. M. Saudek, C. A. Phillips, J. M. Merrick, J. F. Tomb, B. A. Dougherty, K. F. Bott, P. C. Hu, T. S. Lucier, S. N. Peterson, H. O. Smith, C. A. Hutchison, and J. C. Venter, “The minimal gene complement of Mycoplasma genitalium,” *Science*, vol. 270, no. 5235, pp. 397–403, Oct. 1995.

[21] D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie, C. Merryman, S. Vashee, R. Krishnakumar, N. Assad-Garcia, C. Andrews-Pfannkoch, E. A. Denisova, L. Young, Z.-Q. Qi, T. H. Segall-Shapiro, C. H. Calvey, P. P. Parmar, C. A. Hutchison, H. O. Smith, and J. C. Venter, “Creation of a bacterial cell controlled by a chemically synthesized genome,” *Science*, vol. 329, no. 5987, pp. 52–6, Jul. 2010.

[22] MIT, “International Genetically Engineered Machine competition (iGEM),” *http://igem.org/*.

[23] G. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[24] R. Carlson, “The changing economics of DNA synthesis.,” *Nature biotechnology*, vol. 27, no. 12, pp. 1091–4, Dec. 2009.

[25] A. Khalil and J. Collins, “Synthetic biology: applications come of age,” *Nature Reviews Genetics*, vol. 11, no. 5, pp. 367–379, 2010.

[26] J. Aleksic, F. Bizzari, Y. Cai, and B. Davidson, “Development of a novel biosensor for the detection of arsenic in drinking water,” *IET Synthetic Biology*, pp. 87–90, 2007.

[27] “Alkanivore,” *Equipe de l’université de Delft - iGEM 2010 - http://2010.igem.org/Team:TU_Delft*.

[28] M. S. Ferry, J. Hasty, and N. A. Cookson, “Synthetic biology approaches to biofuel production,” *Biofuels*, vol. 3, no. 1, pp. 9–12, Jan. 2012.

[29] T. P. Howard, S. Middelhaufe, K. Moore, C. Edner, D. M. Kolak, G. N. Taylor, D. A. Parker, R. Lee, N. Smirnoff, S. J. Aves, and J. Love, "Synthesis of customized petroleum-replica fuel molecules by targeted modification of free fatty acid pools in Escherichia coli.," *Proceedings of the National Academy of Sciences USA*, vol. 110, no. 19, pp. 7636–41, May 2013.

[30] M. L. Simpson, G. S. Sayler, B. M. Applegate, S. Ripp, D. E. Nivens, M. J. Paulus, and G. E. J. Jr, "Bioluminescent-bioreporter integrated circuits form novel whole-cell biosensors," *Trends in Biotechnology*, vol. 16, no. 8, pp. 332–338, 1998.

[31] "agrEcoli," *Equipe de l'université de Bristol - iGEM 2010 - http://2010.igem.org/Team:BCCS-Bristol*.

[32] C. Kemmer, D. A. Fluri, U. Witschi, A. Passeraub, A. Gutzwiller, and M. Fussenegger, "A designer network coordinating bovine artificial insemination by ovulation-triggered release of implanted sperms," *Journal of controlled release*, vol. 150, no. 1, pp. 23–9, Feb. 2011.

[33] G. Kanter, J. Yang, A. Voloshin, S. Levy, J. R. Swartz, and R. Levy, "Cell-free production of scFv fusion proteins: an efficient approach for personalized lymphoma vaccines.," *Blood*, vol. 109, no. 8, pp. 3393–9, Apr. 2007.

[34] E. K. Jaffe, "An Artificial Gene for Human Porphobilinogen Synthase Allows Comparison of an Allelic Variation Implicated in Susceptibility to Lead Poisoning," *Journal of Biological Chemistry*, vol. 275, no. 4, pp. 2619–2626, Jan. 2000.

[35] W. Weber, R. Schoenmakers, B. Keller, M. Gitzinger, T. Grau, M. Daoud-El Baba, P. Sander, and M. Fussenegger, "A synthetic mammalian gene circuit reveals antituberculosis compounds.," *Proceedings of the National Academy of Sciences USA*, vol. 105, no. 29, pp. 9994–8, Jul. 2008.

[36] N. Saeidi, C. K. Wong, T.-M. Lo, H. X. Nguyen, H. Ling, S. S. J. Leong, C. L. Poh, and M. W. Chang, "Engineering microbes to sense and eradicate Pseudomonas aeruginosa, a human pathogen.," *Molecular systems biology*, vol. 7, no. 521, p. 521, Jan. 2011.

[37] A. A. Gilad, M. T. McMahon, P. Walczak, P. T. Winnard, V. Raman, H. W. M. van Laarhoven, C. M. Skoglund, J. W. M. Bulte, and P. C. M. van Zijl, "Artificial reporter gene providing MRI contrast based on proton exchange.," *Nature biotechnology*, vol. 25, no. 2, pp. 217–9, Feb. 2007.

[38] V. J. J. Martin, D. J. Pitera, S. T. Withers, J. D. Newman, and J. D. Keasling, "Engineering a mevalonate pathway in Escherichia coli for production of terpenoids," *Nature biotechnology*, vol. 21, no. 7, pp. 796–802, Jul. 2003.

[39] CNAM, "Observatoire de la biologie de synthèse," *http://biologie-synthese.cnam.fr/*.

[40] "Ethics of Synthetic Biology - Opinion No 25," *The European Group on Ethics in Science and New Technologies to the European Commission*, 2009.

[41] "New Directions: The Ethics of Synthetic Biology and Emerging Technologies," *Presidential Commission for the Study of Bioethical Issues*, 2010.

[42] T. Landrain, M. Meyer, A. Perez, and R. Sussan, "Do-it-yourself biology: challenges and promises for an open science and technology movement," *Systems and Synthetic Biology*, vol. 7, no. 3, pp. 115–126, Aug. 2013.

[43] M. H. Skolnick, D. E. Goldgar, Y. Miki, J. Swenson, A. Kamb, K. D. Harshman, D. M. Shattuck-Eidens, S. V Tavtigian, R. W. Wiseman, and P. A. Futreal, "US5747282 - 17q-linked breast and ovarian cancer susceptibility gene," 1998.

[44] F. Couch, A. Kamb, J. M. Rommens, J. Simard, S. V. Tavtigian, and B. L. Weber, "US5837492 - Chromosome 13-linked breast cancer susceptibility gene," 1998.

[45] "Association for Molecular Pathology et al. v. Myriad Genetics Inc et al.," *Supreme Court of the United States*, 2013.

[46] Y. Herve, "Virtual prototyping with VHDL-AMS," *IEEE International Conference on Technology*, vol. 2, pp. 761–765, 2003.

[47] "Comsol Multiphysics," *http://www.comsol.fr/*.

[48] "Silvaco," *http://www.silvaco.fr/*.

[49] D. Chandran, F. T. Bergmann, H. M. Sauro, and D. Densmore, "Computer-Aided Design for Synthetic Biology," *Design and Analysis of Biomolecular Circuits*, pp. 203–224, 2011.

[50] "BioJADE," *http://web.mit.edu/jagoler/www/biojade*.

[51] IGEM, "Registry of Standard Biological Parts," *http://parts.igem.org/*.

[52] "GenoCAD," *http://www.genocad.org/*.

[53] "SynBioSS," *http://synbioss.sourceforge.net/*.

[54] "TinkerCell," *http://www.tinkercell.com/*.

[55] J. Beal, T. Lu, and R. Weiss, "Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks," *PLoS ONE*, vol. 6, no. 8, 2011.

[56] J. Beal, R. Weiss, D. Densmore, A. Adler, E. Appleton, J. Babb, S. Bhatia, N. Davidsohn, T. Haddock, J. Loyall, R. Schantz, V. Vasilev, and F. Yaman, "An End-to-End Workflow for Engineering of Biological Networks from High-Level Specifications," *ACS Synthetic Biology*, vol. 1, no. 8, pp. 317–331, 2012.

[57] J. Beal, R. Weiss, D. Densmore, A. Adler, J. Babb, and S. Bhatia, "TASBE: A tool-chain to accelerate synthetic biological engineering," *Proceedings of the 3rd International Workshop on Bio-Design Automation*, pp. 19–21, 2011.

[58] "Clotho," *http://www.clothocad.org/*.

[59] F. Yaman, S. Bhatia, A. Adler, D. Densmore, and J. Beal, "Automated selection of synthetic biology parts for genetic regulatory networks.," *ACS synthetic biology*, vol. 1, no. 8, pp. 332–44, Aug. 2012.

[60] "JDesigner," *http://sbw.kgi.edu/software/jdesigner.htm*.

[61] "CellDesigner," *http://www.celldesigner.org/*.

[62] "COPASI," *http://www.copasi.org/*.

[63] "GeneDesign," *http://www.genedesign.org*.

[64] "GeneDesigner," *http://www.dna20.com/genedesigner*.

[65] Biofab, "Sequence Refiner," *http://biofab.org/*.

[66] B. Canton, A. Labno, and D. Endy, "Refinement and standardization of synthetic biological parts and devices.," *Nature biotechnology*, vol. 26, no. 7, pp. 787–93, Jul. 2008.

[67] "Systems Biology Markup Language," *http://sbml.org/*.

[68] "Synthetic Biology Open Language (SBOL)," *http://www.sbolstandard.org*.

[69] C. J. Myers, N. Barker, H. Kuwahara, K. Jones, C. Madsen, and N.-P. D. Nguyen, "Genetic design automation," *ICCAD 2009. IEEE/ACM International Conference on*, vol. 2, pp. 713–716, 2009.

[70] D. Densmore and S. Hassoun, "Design Automation for Synthetic Biological Systems," *IEEE Design & Test of Computers*, no. June, pp. 7–20, 2012.

[71] M. W. Lux, B. W. Bramlett, D. a Ball, and J. Peccoud, "Genetic design automation: engineering fantasy or scientific renewal?," *Trends in biotechnology*, vol. 30, no. 2, pp. 120–6, 2012.

[72] S. Jones and J. Thornton, "Principles of protein-protein interactions," *Proceedings of the National Academy of Sciences USA*, vol. 93, no. January, pp. 13–20, 1996.

[73] S. Jones, P. van Heyningen, H. M. Berman, and J. M. Thornton, "Protein-DNA interactions: A structural analysis.," *Journal of molecular biology*, vol. 287, no. 5, pp. 877–96, Apr. 1999.

[74] G. U. Nienhaus, *Protein-Ligand Interactions*. Humana Press, 2005.

[75] J. Berg, J. Tymoczko, and L. Stryer, “RNA Synthesis and Splicing & Protein Synthesis,” *Biochemistry*, 2002.

[76] E. Lander, “7.00x Introduction to Biology - The Secret of Life,” *MITx*, 2013.

[77] N. Battey, N. James, A. Greenland, and C. Brownlee, “Exocytosis and endocytosis,” *Plant cell*, vol. 11, no. 4, pp. 643–60, Apr. 1999.

[78] W. D. Stein, *Channels, carriers, and pumps: an introduction to membrane transport*. Academic Press, 1990.

[79] E. Carafoli, “Intracellular calcium homeostasis,” *Annual review of biochemistry*, 1987.

[80] B. Katz and R. Miledi, “The timing of calcium action during neuromuscular transmission,” *The Journal of physiology*, 1967.

[81] G. W. Gokel and S. Negin, “Synthetic Ion Channels: From Pores to Biological Applications,” *Accounts of Chemical Research*, vol. 46, no. 12, pp. 2824–2833, Jun. 2013.

[82] C. M. Ajo-Franklin, D. A. Drubin, J. A. Eskin, E. P. S. Gee, D. Landgraf, I. Phillips, and P. A. Silver, “Rational design of memory in eukaryotic cells,” *Genes & development*, vol. 21, no. 18, pp. 2271–6, Sep. 2007.

[83] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA.,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.

[84] J. Bonnet, P. Subsoontorn, and D. Endy, “Rewritable digital data storage in live cells via engineered control of recombination directionality.,” *Proceedings of the National Academy of Sciences USA*, vol. 109, no. 23, pp. 8884–9, Jun. 2012.

[85] A. Doboli and R. Vemuri, “Behavioral modeling for high-level synthesis of analog and mixed-signal systems from VHDL-AMS,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 22, no. 11, pp. 1504–1520, 2003.

[86] O. Maler, “Analog Circuit Verification: a State of an Art,” *Electronic Notes in Theoretical Computer Science*, vol. 153, no. 3, pp. 3–7, Jun. 2006.

[87] C. J. Myers, R. R. Harrison, D. Walter, N. Seegmiller, and S. Little, “The Case for Analog Circuit Verification,” *Electronic notes in theoretical computer science*, vol. 153, pp. 53–63, 2006.

[88] C. D. Systems, “Encounter RTL Compiler,” *http://www.cadence.com/eu/pages/rtl_compiler.aspx*.

[89] Synopsys, “Design Compiler Graphical,” *http://www.synopsys.com/tools/implementation/rtlsynthesis/dcgraphical/Pages/default.aspx*.

[90] M. Graphics, “Mentor Graphics EDA,” *http://www.mentor.com/*.

[91] LIP6, “Alliance,” *http://www-soc.lip6.fr/recherche/cian/alliance/*.

[92] MIT, “ODIN II,” *https://code.google.com/p/odin-ii/*.

[93] Berkeley, “ABC,” *http://www.eecs.berkeley.edu/~alanmi/abc/*.

[94] Y. Gendrault, M. Madec, C. Lallement, F. Pecheux, and J. Haiech, “Synthetic biology methodology and model refinement based on microelectronic modeling tools and languages.,” *Biotechnology journal*, vol. 6, no. 7, pp. 796–806, Jul. 2011.

[95] A. Hill, “The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves,” *Journal of Physiology*, 1910.

[96] M. L. Simpson, C. D. Cox, G. D. Peterson, and G. S. Sayler, “Engineering in the biological substrate: information processing in genetic circuits,” *Proceedings of the IEEE*, vol. 92, no. 5, pp. 848–863, 2004.

[97] G. Peterson, P. Ashenden, and D. Teegarden, *The System Designer’s Guide to VHDL-AMS*, 2nd ed. Morgan Kaufmann, 2002.

[98] Dolphin, “SMASH Mixed-Signal Simulator,” *http://www.dolphin.fr/*.

[99] “Langage SystemC-AMS,” *http://www.systemc-ams.org*.

[100] A. Vachoux, C. Grimm, and K. Einwich, “SystemC-AMS requirements, design objectives and rationale,” *2003 Design, Automation and Test in Europe Conference and Exhibition*, pp. 388–393, 2003.

[101] Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss, and Y. Benenson, “Multi-input RNAi-based logic circuit for identification of specific cancer cells.,” *Science*, vol. 333, no. 6047, pp. 1307–11, Sep. 2011.

[102] S. Volinia, G. A. Calin, C.-G. Liu, S. Ambs, A. Cimmino, F. Petrocca, R. Visone, M. Iorio, C. Roldo, M. Ferracin, R. L. Prueitt, N. Yanaihara, G. Lanza, A. Scarpa, A. Vecchione, M. Negrini, C. C. Harris, and C. M. Croce, “A microRNA expression signature of human solid tumors defines cancer gene targets.,” *Proceedings of the National Academy of Sciences USA*, vol. 103, no. 7, pp. 2257–61, Mar. 2006.

[103] S. Ausländer, D. Ausländer, M. Müller, M. Wieland, and M. Fussenegger, “Programmable single-cell mammalian biocomputers.,” *Nature*, vol. 487, no. 7405, pp. 123–7, Jul. 2012.

[104] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, “A fast, robust and tunable synthetic gene oscillator.,” *Nature*, vol. 456, no. 7221, pp. 516–9, Nov. 2008.

[105] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. 2002.

[106] M. Madec, C. Lallement, Y. Gendrault, and J. Haiech, "Design methodology for synthetic biosystems," *Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference*, pp. 621–626, 2010.

[107] L. Zadeh, "Fuzzy Sets," *Information and control*, 1965.

[108] P. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiological Genomics*, vol. 3, pp. 9–15, 2000.

[109] M. McKenna and B. M. Wilamowski, "Implementing a fuzzy system on a field programmable gate array," *International Joint Conference on Neural Networks. Proceedings*, vol. 1, pp. 189–194, 2001.

[110] H. R. Pourshaghaghi and J. P. de Gyvez, "Dynamic voltage scaling based on supply current tracking using fuzzy Logic controller," *16th IEEE International Conference on Electronics, Circuits and Systems*, pp. 779–782, Dec. 2009.

[111] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*. 1995.

[112] T. Ross, *Fuzzy logic with engineering applications*. 2009.

[113] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 109–119, 1999.

[114] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.

[115] MathWorks, "Fuzzy Logic Toolbox," *http://www.mathworks.fr/products/fuzzy-logic/*.

[116] P. Cingolani, "jFuzzyLogic," *http://jfuzzylogic.sourceforge.net/*.

[117] R. Liebscher, "pyfuzzy - Python fuzzy package," *http://pyfuzzy.sourceforge.net/*.

[118] B. McCart, "C++ Fuzzy Logic Programming Library," *http://sourceforge.net/projects/cpp-fuzzy-logic/*.

[119] D. Galan, C. J. Jimenez, A. Barriga, and S. Sanchez-Solano, "VHDL package for description of fuzzy logic controllers," *Proceedings of the Design Automation Conference*, pp. 528–533, 1995.

[120] Mathworks, "Matlab," *http://www.mathworks.fr/products/matlab/*.

[121] A. Torres and J. J. Nieto, "Fuzzy logic in medicine and bioinformatics," *Journal of biomedicine & biotechnology*, no. 2, Jan. 2006.

[122] Y. Jin and L. Wang, *Fuzzy systems in bioinformatics and computational biology*. 2009.

[123] H. Ressom, R. Reynolds, and R. S. Varghese, "Increasing the efficiency of fuzzy logic-based gene expression data analysis.," *Physiological genomics*, vol. 13, no. 2, pp. 107–17, Apr. 2003.

[124] M. Terzer, M. Jovanovic, A. Choutko, O. Nikolayeva, A. Korn, D. Brockhoff, F. Zu, M. Friedmann, E. Zitzler, J. Stelling, and S. Panke, "Design of a biological half adder," *IET Synthetic Biology*, pp. 53–58, 2007.

[125] Z. Konkoli, "Safe uses of Hill ' s model : an exact comparison with the Adair-Klotz model," *Theoretical Biology and Medical Modelling*, vol. 8, no. 1, p. 10, 2011.

[126] E. Fischer, "Einfluss der Configuration auf die Wirkung den Enzyme," *Berichte der deutschen chemischen Gesellschaft*, vol. 27, no. 3, pp. 2985–2993, 1984.

[127] J. Monod, J. Wyman, and J. P. Changeux, "On the Nature of Allosteric Transitions: a Plausible Model," *Journal of molecular biology*, vol. 12, no. December, pp. 88–118, May 1965.

[128] D. E. Koshland, G. Némethy, and D. Filmer, "Comparison of experimental binding data and theoretical models in proteins containing subunits," *Biochemistry*, vol. 5, no. 1, pp. 365–385, Jan. 1966.

[129] J. Haiech, B. Vallet, R. Aquaron, and J. G. Demaille, "Ligand binding to macromolecules: Determination of binding parameters by combined use of ligand buffers and flow dialysis; application to calcium-binding proteins," *Analytical Biochemistry*, vol. 105, no. 1, pp. 18–23, 1980.

[130] A. J. R. Heck and R. H. H. Van Den Heuvel, "Investigation of intact protein complexes by mass spectrometry," *Mass spectrometry reviews*, vol. 23, no. 5, pp. 368–389, 2004.

[131] I. D. Alves, C. K. Park, and V. J. Hruby, "Plasmon Resonance Methods in GPCR Signaling and Other Membrane Events," *Curr Protein Pept Sci.*, vol. 6, no. 4, pp. 293–312, 2005.

[132] Z. Salamon, G. Tollin, I. Alves, and V. Hruby, "Chapter 6 Plasmon Resonance Methods in Membrane Protein Biology: Applications to GPCR Signaling," *Methods in Enzymology*, vol. 461, pp. 123–146, 2009.

[133] I. Protasevich, B. Ranjbar, V. Lobachov, A. Makarov, R. Gilli, C. Briand, D. Lafitte, and J. Haiech, "Conformation and thermal denaturation of apocalmodulin: role of electrostatic mutations," *Biochemistry*, vol. 36, no. 8, pp. 2017–2024, Feb. 1997.

[134] R. Gilli, D. Lafitte, C. Lopez, M. Kilhoffer, A. Makarov, C. Briand, and J. Haiech, "Thermodynamic analysis of calcium and magnesium binding to calmodulin," *Biochemistry*, vol. 37, no. 16, pp. 5450–5456, Apr. 1998.

[135] F. Du, Y. Liang, B.-R. Zhou, Y. Xia, M.-C. Kilhoffer, and J. Haiech, “Unfolding of creatine kinase induced by acid studied by isothermal titration calorimetry and fluorescence spectroscopy,” *Thermochimica Acta*, vol. 416, no. 1–2, pp. 17–21, Jun. 2004.

[136] R. Dagher, C. Brière, M. Fève, M. Zeniou, C. Pigault, C. Mazars, H. Chneiweiss, R. Ranjeva, M.-C. Kilhoffer, and J. Haiech, “Calcium fingerprints induced by calmodulin interactors in eukaryotic cells,” *Biochimica et biophysica acta*, vol. 1793, no. 6, pp. 1068–1077, Jun. 2009.

[137] M.-C. Kilhoffer, J. Haiech, and J. G. Demaille, “Ion binding to calmodulin,” *Molecular and Cellular Biochemistry*, vol. 51, pp. 33–54, 1983.

[138] T. H. Crouch and C. B. Klee, “Positive cooperative binding of calcium to bovine brain calmodulin,” *Biochemistry*, vol. 19, no. 16, pp. 3692–36928, Aug. 1980.

[139] C.-L. A. Wang, “A note on Ca2+ binding to calmodulin,” *Biochemical and Biophysical Research Communications*, vol. 130, no. 1, pp. 426–430, 1985.

[140] M. I. Stefan, S. J. Edelstein, and N. Le Novère, “An allosteric model of calmodulin explains differential activation of PP2B and CaMKII,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 31, pp. 10768–10773, Aug. 2008.

[141] A. C. Cameron and F. Windmeijer, “An R-squared measure of goodness of fit for some common nonlinear regression models,” *Journal of Econometrics*, 1995.

[142] J. Haiech, M. C. Kilhoffer, T. J. Lukas, T. A. Craig, D. M. Roberts, and D. M. Watterson, “Restoration of the calcium binding activity of mutant calmodulins toward normal by the presence of a calmodulin binding structure.,” *The Journal of biological chemistry*, vol. 266, no. 6, pp. 3427–31, Feb. 1991.

[143] L. Cohen, “The History of Noise [on the 100th anniversary of its birth],” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 20–45, 2005.

[144] G. Vasilescu, *Electronic Noise and Interfering Signals*. Springer, 1999.

[145] P. J. Fish, *Electronic noise and low noise design*. Macmillan Press, 1993.

[146] W. P. Jolly, *Low Noise Electronics*. The English Universities Press, 1967.

[147] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, and R. C. Goodlin, “Adaptive Noise Cancelling : Principles and Applications,” *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[148] H. W. Ott, *Noise reduction techniques in electronic systems*. John Wiley & Sons, 1976.

[149] R. Howard, *Principles of Random Signal Analysis and Low Noise Design:The Power Spectral Density and its Applications*. Wiley-IEEE Press, 2002, pp. 256–299.

[150] I. Lestas, G. Vinnicombe, and J. Paulsson, “Fundamental limits on the suppression of molecular fluctuations,” *Nature*, vol. 467, no. 7312, pp. 174–178, Sep. 2010.

[151] A. Eldar and M. B. Elowitz, “Functional roles for noise in genetic circuits,” *Nature*, vol. 467, no. 7312, pp. 167–173, Sep. 2010.

[152] E. Sejdić and L. A. Lipsitz, “Necessity of noise in physiology and medicine,” *Computer Methods and Programs in Biomedicine*, vol. 111, no. 2, pp. 459–470, 2013.

[153] J. M. Raser and E. K. O’Shea, “Noise in Gene Expression: Origins, Consequences, and Control,” *Science*, vol. 309, no. 5743, pp. 2010–2013, 2005.

[154] M. L. Simpson, C. D. Cox, M. S. Allen, J. M. McCollum, R. D. Dar, D. K. Karig, and J. F. Cooke, “Noise in biological circuits,” *Wiley Interdisciplinary Reviews: Nanomedicine and Nanobiotechnology*, vol. 1, no. 2, pp. 214–225, 2009.

[155] G. Montalenti, “Le Bruit de Barkhausen dans les Matériaux Ferromagnétiques,” *Revue de Physique Appliquée*, vol. 5, pp. 87–93, 1970.

[156] Y. Reibel, “Développement et caractérisation d’une caméra vidéo numérique rapide (500 l/s) à haute résolution (10 bits). Application à la reconstruction 3D de surfaces microscopiques,” *Université Louis Pasteur*, 2001.

[157] L. J. DeFelice, *Introduction to Membrane Noise*. Springer US, 1981, pp. 231–332.

[158] M. Bier and J. Gallaher, “Ion Traffic Through a Cell Mebrane and how its 1/f Noise Connects to Gambler’s Ruin, Catalan Numbers and Zipf's Law,” *Fluctuation and Noise Letters*, vol. 10, no. 4, pp. 419–430, Dec. 2011.

[159] J. R. Clay, “Unified theory of View the MathML source and conductance noise in nerve membrane,” *Journal of Theoretical Biology*, vol. 66, no. 4, pp. 763–773, 1977.

[160] D. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *The journal of physical chemistry*, vol. 93555, no. 1, pp. 2340–2361, 1977.

[161] G. E. P. Box and M. E. Muller, “A Note on the Generation of Random Normal Deviates,” *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 1958.

[162] H. R. Ueda, S. Hayashi, W. Chen, M. Sano, M. Machida, Y. Shigeyoshi, M. Iino, and S. Hashimoto, “System-level identification of transcriptional circuits underlying mammalian circadian clocks,” *Nature Genetics*, vol. 37, pp. 187–192, 2005.

[163] A. Deutsch and S. Dormann, *Cellular Automaton Modeling of Biological Pattern Formation*. Birkhäuser, 2005, pp. 59–100.

[164] H. Koeppl, D. Densmore, G. Setti, and M. di Bernardo, *Design and Analysis of Biomolecular Circuits*. Springer, 2011, pp. 43–62.

[165] S. Plimpton and A. Slepoy, “ChemCell : A Particle-Based Model of Protein Chemistry and Diffusion in Microbial Cells,” *Sandia National Laboratory Technical Report*, 2003.

[166] J. Stiles and T. Bartol, “Monte Carlo methods for simulating realistic synaptic microphysiology using MCell,” *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 87–127, 2001.

[167] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, “Detailed Simulations of Cell Biology with Smoldyn 2.1,” *PLoS Computational Biology*, vol. 6, no. 3, p. e1000705, Mar. 2010.

[168] L. Boulianne, S. Al Assaad, M. Dumontier, and W. J. Gross, “GridCell: a stochastic particle-based biological system simulator,” *BMC Systems Biology*, vol. 2, no. 1, p. 66, 2008.

[169] S. N. V. Arjunan and M. Tomita, “A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of E. coli MinE to E-ring formation.,” *Systems and synthetic biology*, vol. 4, no. 1, pp. 35–53, Mar. 2010.

[170] M. Tomita, K. Hashimoto, K. Takahashi, T. S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J. C. Venter, and C. a Hutchison, “E-CELL: software environment for whole-cell simulation.,” *Bioinformatics (Oxford, England)*, vol. 15, no. 1, pp. 72–84, Jan. 1999.

Publications

Revues internationales :

- HAIECH J., GENDRAULT Y., KILHOFFER M.C., RANJEVA R., MADEC M., LALLEMENT C., A general framework revisiting the ligand binding to a macromolecule, 2013, Biochimica et Biophysica Acta (session spéciale : calcium), 2013, soumission en cours.
- GENDRAULT Y., MADEC M., LALLEMENT C., HAIECH J., Modeling biology with HDL languages: a first step toward a Genetic Design Automation tool inspired from microelectronics, IEEE Transactions on Biomedical Engineering, 2013, Accepté, sous presse.
- GENDRAULT Y., MADEC M., LALLEMENT C., PÊCHEUX F., HAIECH J., Synthetic biology methodology and model refinement based on microelectronic modeling tools and languages, Biotechnol. J. 6, 2011, pp. 796-806.
- MADEC M., GENDRAULT Y., LALLEMENT C., HAIECH J., Design methodology and modeling for synthetic biosystems, Int. J. Microelectron. Comput. Sci. 1, 2010, pp. 147-155.

Conférences internationales :

- GENDRAULT Y., MADEC M., WLOTZKO V., LALLEMENT C., HAIECH J., Fuzzy logic, an efficient intermediate abstraction level for synthetic biology, BioCAS 2013.
- MADEC M., PECHEUX F., GENDRAULT Y., BAUER L., LALLEMENT C., EDA inspired Open-source Framework for Synthetic Biology, BioCAS 2013.
- GENDRAULT Y., MADEC M., WLOTZKO V., ANDRAUD M., LALLEMENT C., HAIECH J., Using digital electronic design flow to create a genetic design automation tool, 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2012), San Diego (USA), 28 Août – 1 Septembre 2012, Actes pp. 5530-5533.
- MADEC M., GENDRAULT Y., LALLEMENT C., HAIECH J., A game-of-life like simulator for design-oriented modeling of biobricks in synthetic biology, 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2012), San Diego (USA), 28 Août – 1 Septembre 2012, Actes pp. 5462-5465.

- GENDRAULT Y., MADEC M., LALLEMENT C., PÊCHEUX F., HAIECH J., Computer-aided design in synthetic biology: A system designer approach, 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2011), Barcelona (Spain), 26-29 Octobre 2011, Papier invité.

- GENDRAULT Y., MADEC M., LALLEMENT C., HAIECH J., Multi-abstraction modeling in synthetic biology, 3rd IEEE International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010), Rome (Italy), 7-10 Novembre 2010, Actes pp. 1-5.

- GENDRAULT Y., MADEC M., LALLEMENT C., HAIECH J., A design kit for synthetic biology, International Conference on Synthetic Biology, Evry (France), 15-16 Décembre 2010.

- MADEC M., LALLEMENT C., GENDRAULT Y., HAIECH J., Design methodology for synthetic biosystems, 17th IEEE International Conference Mixed Design of Integrated Circuits and Systems (MIXDES'2010), Wroclaw (Poland), 24-26 Juin 2010, Actes pp. 621-626.

Conférences nationales :

- MADEC M., GENDRAULT Y., LALLEMENT C., HAIECH J., Introduction à la biologie synthétique et au développement d'outils de conception dédiés, 12èmes Journées Pédagogiques du CNFM (JPCNFM 2012), Saint-Malo (France), 28-30 Novembre 2012, Actes pp. 123-128.

- GENDRAULT Y., Vers un outil de CAO pour la biologie synthétique, inspiré de la microélectronique, XVèmes Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM'2012), Marseille (France), 2012.

- GENDRAULT Y., Adaptation des outils et de la méthodologie de la microélectronique pour la conception de systèmes biologiques, XIVèmes Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM'2011), Cachan (France), 23-25 Mai 2011.

- MADEC M., LALLEMENT C., GENDRAULT Y., HAIECH J., La microélectronique et la biologie synthétique, Conférence Savoir en commun "Le Corps", Strasbourg (France), 18 Novembre 2010, lien : http://audiovideocast.unistra.fr/avc/courseaccess?id=4974

Annexes

Annexe A

Code graphique

Afin de représenter graphiquement les biosystèmes, nous avons développé deux codes graphiques clairs et facilement compréhensibles. La première version intègre beaucoup de détails, comme les promoteurs des gènes, alors que la deuxième est une version allégée utilisant un symbolisme. Les deux versions sont présentées dans cette annexe.

A.1 Les espèces présentes

Les différentes espèces présentes sont représentées, dans la première version du standard graphique, selon un code couleur correspondant à leur fonction dans le biosystème. Ainsi, les espèces correspondant à des entrées du système (des protéines, des ions ou encore d'autres espèces chimiques) sont illustrées Figure A.1.A, les espèces intermédiaires (des protéines ou des complexes protéiques) sont illustrées Figure A.1.B, et les espèces correspondant aux sorties (des protéines reporteurs) sont illustrées Figure A.1.C.

Figure A.1 : Espèces utilisées dans un biosystème, selon le code graphique détaillé.

Dans la version simplifiée du code graphique, nous ne différencions pas les fonctions des différentes espèces et elles sont représentées simplement par un rectangle de couleur, ce qui est illustré Figure A.2.

Figure A.2 : Espèces utilisées dans un biosystème, selon le code graphique simplifié.

A.2 Le gène

Les différentes zones d'un gène sont représentées à l'aide du code graphique détaillé. La partie de contrôle, constituée du promoteur, va utiliser les éléments graphiques de la Figure A.3. En A.,

nous retrouvons le promoteur, en B., la ou les espèces activateurs, et en C., la ou les espèces répresseurs.

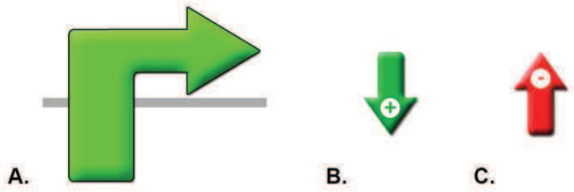


Figure A.3 : Le promoteur, selon le code graphique détaillé.

Le code ADN correspondant à la protéine qui sera synthétisée est représenté par un des éléments Figure A.1.B ou A.1.C. Dans le code détaillé, il est encadré par les codes ADN correspondant à l'initiateur et au terminateur du gène, dont les éléments graphiques sont illustrés Figure A.4.A et A.4.B respectivement.

Figure A.4 : Initiateur et terminateur d'un gène.

Les différents éléments graphiques utilisés dans ce code détaillé n'apparaissent pas dans la version simplifiée, afin de rendre les biosystèmes plus compacts. Pour mettre en évidence la présence d'un activateur nous utilisons une flèche simple et pour un répresseur une flèche avec une terminaison en forme de trait perpendiculaire, dans ce code allégé.

A.3 La synthèse d'une protéine

Nous retrouvons Figure A.5 la représentation de la synthèse d'une protéine utilisée dans un biosystème, selon le code graphique détaillé en A. et selon le code graphique simplifié en B.. Les espèces intermédiaires comme l'ARN messager n'y figurent pas afin de simplifier la représentation d'un biosystème complet. Le raccourci entre le code ADN correspondant à la protéine et la protéine est aussi utilisé.

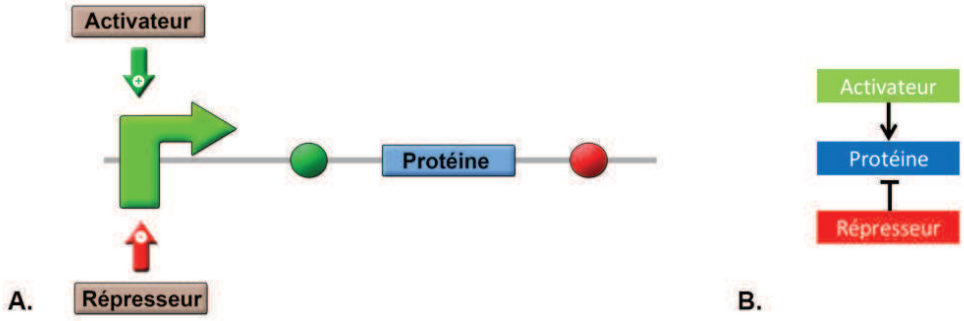


Figure A.5 : Synthèse d'une protéine, selon les deux représentations.

A.4 La complexation

Le dernier mécanisme représenté grâce aux deux codes graphiques est la liaison entre deux espèces. Pour le code détaillé, il est illustré Figure A.6.A et pour le code simplifié, il est illustré Figure A.6.B.

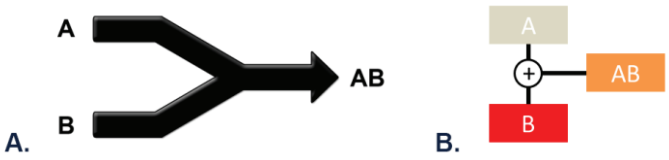


Figure A.6 : Complexation, selon les deux représentations.

Annexe B

Codes VHDL-AMS des modèles flux de signal

B.1 Complexation

```
library ieee;
use ieee.math_real.all,ieee.std_logic_1164.all;
use work.all;

entity complexation is

-- Coefficients de complexation et décomplexation

      generic (kp1,km1,kdegr : real := 0.0);

-- Espèces en entrée
-- Puis complexe produit et entrées recalculées

      port (quantity ProE1,ProE2 : in real;
            quantity ProS1,ProS2,Comp : out real);

end;

architecture biologique of complexation is

Begin

-- Calcul de la quantité de complexe formé

Comp'dot == kp1*((ProS1 + ProS2)/2.0 - abs((ProS1 - ProS2)/2.0))
            - km1*Comp - kdegr*Comp;

-- Correction des protéines en entrée après création de complexes

ProS1 == ProE1 - Comp*(ProE1/(ProE1+1.0e-9));
ProS2 == ProE2 - Comp*(ProE2/(ProE2+1.0e-9));

end;
```

B.2 Synthèse d'une protéine

```
library ieee;
use ieee.math_real.all,ieee.std_logic_1164.all;
use work.all;

entity synthese_proteine is

-- Paramètres de l'équation de synthèse
```

```
      generic (Kxa,Kxr,ktr,ktl,kdep,kcompl,dARNm,dp,a : real := 0.0;
               na,nr : integer := 0);

-- Entrées/Sorties de protéines

      port (quantity Act,Rep,DegrARN,DegrPro : in real;
            quantity ProS : out real);

end;

architecture biologique of synthese_proteine is

-- Molécule intermédiaire de la synthèse : l'ARNm

      quantity ARNm : real;


Begin

-- ODE pour calculer la quantité d'ARNm produite
-- Dépend de la quantité d'activateur et de répresseur

ARNm'dot == ktr*(a + 1.0/((1.0+(Kxa/(Act+1e-9))**na)*(1.0+(Kxr/(Rep+1e-9))**(-
nr))))
            - (dARNm+DegrARN)*ARNm;

-- ODE pour calculer la quantité de protéine
-- produite à partir de l'ARNm

ProS'dot == ktl*ARNm'delayed(kdep) - (dp+DegrPro+kcompl)*ProS;

end;
```

Annexe C

Codes VHDL-AMS des modèles conservatifs

C.1 Complexation

```
library ieee;
use ieee.math_real.all,ieee.std_logic_1164.all;

-- Chargement de la librairie définissant les natures biologiques

use work.biological_systems.all;
use work.all;

entity complexation_K is

-- Paramètres de la complexation

      generic (kp1,km1,kdegr : real := 0.0);

-- Noeuds de connexion du mécanisme

      port (terminal Pro1 : prot1;
              terminal Pro2 : prot2;
            terminal Comp : complexe);
end;

architecture biologique of complexation_K is

-- Définition des différentes quantités

      quantity Cpro1 across Fpro1 through cell_prot1 to Pro1;
      quantity Cpro2 across Fpro2 through cell_prot2 to Pro2;
      quantity Ccomp across Fcomp through Comp to cell_complexe;

Begin

-- Instanciation des différents résistances

r_pro1 : entity resistance_B(biologique)
             generic map (1.0/kp1)
             port map (a => Pro1_term, b => cell);
r_pro2 : entity resistance_B(biologique)
             generic map (1.0/kp1)
             port map (a => Pro2_term, b => cell);

r_comp1 : entity resistance_B(biologique)
              generic map (1.0/km1)
              port map (a => Comp_term, b => cell);
r_comp2 : entity resistance_B(biologique)
              generic map (1.0/kdegr)
              port map (a => Comp_term, b => cell);
```

```
-- Générateur du courant du complexe

Fcomp == kp1*Cpro1*Cpro2;

-- Générateurs des espèces, suite à la dissociation

Fpro1 == km1*Ccomp;
Fpro2 == km1*Ccomp;

end;
```

C.2 Synthèse d'une protéine

```
library ieee;
use ieee.math_real.all,ieee.std_logic_1164.all;

-- Chargement de la librairie définissant les natures biologiques

use work.biological_systems.all;
use work.all;

entity synthese_K is

-- Paramètres de la synthèse

      generic (Kxa,Kxr,ktr,ktl,kdep,kcompl,dARNm,dp,a,na,nr : real := 0.0);

-- Noeuds de connexion du mécanisme

      port (terminal Act : activateur;
              terminal Rep : represseur;
              terminal ProS : proteine);
end;

architecture biologique of synthese_K is

-- Constantes par défaut des composants

      constant valr_act,valr_rep :real := 10.0;
      constant valc_act,valc_rep,valc_arnm,valc_pros :real := 1.0;

-- Noeud intermédiaire de la synthèse : l'ARNm

      terminal ARNm_term : ARNm;

-- Définition des différentes quantités

      quantity Cact across Act to cell_activateur;
      quantity Crep across Rep to cell_represseur;
      quantity CARNm across FARNm through ARNm to cell_ARNm;
      quantity Cpros across FproS through ProS to cell_proteine;

Begin

-- Instanciation des différents composants
-- Une resistance et une capacité par espèce

r_act : entity resistance_B(biologique)
            generic map (valr_act)
            port map (a => Act, b => cell_activateur);
```

```
c_act : entity capacite_B(biologique)
            generic map (valc_act)
            port map (a => Act, b => cell_activateur);

r_rep : entity resistance_B(biologique)
            generic map (valr_rep)
            port map (a => Rep, b => cell_represseur);
c_rep : entity capacite_B(biologique)
            generic map (valc_rep)
            port map (a => Rep, b => cell_represseur);

r_arnm : entity resistance_B(biologique)
            generic map (1.0/dARNm)
            port map (a => ARNm, b => cell_ARNm);
c_arnm : entity capacite_B(biologique)
            generic map (valc_arnm)
            port map (a => ARNm, b => cell_ARNm);

r_pros : entity resistance_B(biologique)
            generic map (1.0/dp)
            port map (a => ProS, b => cell_proteine);
c_pros : entity capacite_B(biologique)
            generic map (valc_pros)
            port map (a => ProS, b => cell_proteine);

-- Générateur du courant "ARNm"

FARNm == ktr*(a + 1.0/((1.0+(Kxa/(Cact+1e-9))**na)*(1.0+(Kxr/(Crep+1e-9))**(-
nr))));

-- Générateur du courant "Protéine"

FproS == ktl*CARNm;

end;
```

C.3 Paquetage biological_systems

```
library IEEE;
    use IEEE.FUNDAMENTAL_CONSTANTS.all;

package BIOLOGICAL_SYSTEMS is

   subtype CONCENTRATION     is REAL tolerance "DEFAULT_CONCENTRATION";
   subtype FLUX     is REAL tolerance "DEFAULT_FLUX";

   -- Déclarations des différentes natures

   nature ARNm is
      CONCENTRATION        across
      FLUX        through
      ARNm_REF reference;

   -- ...

   -- Déclaration des unités

   attribute UNIT of CONCENTRATION       : subtype is "Concentration";
   attribute UNIT of FLUX                     : subtype is "Protein";
```

```
    attribute SYMBOL of CONCENTRATION     : subtype is "C";
    attribute SYMBOL of FLUX               : subtype is "P";

    -- Déclaration des alias

    alias CELL_ARNm is ARNm_REF;

    -- ...

end package BIOLOGICAL_SYSTEMS;
```

Annexe D

Code SystemC-AMS des modèles TDF

D.1 Complexation

```
#include "systemc-ams.h"

// Création du bloc complexation

SCA_TDF_MODULE(complexation)

{

// Entrées/sorties de la complexation (A, B, AB)
// Avec les rétroactions des autres blocs notées *_i
// et pour les autres blocs notées *_o

      sca_tdf::sca_in<double> A_i, B_i;
      sca_tdf::sca_out<double> AB_o;
      sca_tdf::sca_in<double> AB_i;
      sca_tdf::sca_out<double> A_o , B_o;

// Définition des paramètres de la complexation

      complexation( sc_core::sc_module_name nm, double kon_ ,double koff_):
                  AB_o("AB"), kon(kon_), koff(koff_)
                  {}

// Ajout de délais sur le flux de signal remontant

      void set_attributes()
      {
            AB_i.set_delay(1);
      }

// Initialisation du composant

      void initialize()
      {
            AB=0;
            dt = A_i.get_timestep().to_seconds();
            AB_i.initialize(0);

      }

// Calcul des différentes ODEs à partir du timestep

      void processing()
      {
            A=A_i.read();
            B=B_i.read();
```

```
            diff_AB=(kon*A*B-koff*AB)*dt;
            AB+=(diff_AB + AB_i.read());

            A_o.write((-kon*A*B+koff*AB)*dt);
            B_o.write((-kon*A*B+koff*AB)*dt);
            AB_o.write(AB);
      }

SCA_CTOR(complexation) {}

private:
      double diff_A,diff_B,diff_AB;
      double A,B,AB;
      double kon,koff;
      double dt;
};
```

D.2 Synthèse d'une protéine

```
#include "systemc-ams.h"
#include "math.h"

// Création du bloc synthèse

SCA_TDF_MODULE(synthese)

{

// Entrées/sorties de la synthèse (activateur, répresseur, protéine)
// Avec les rétroactions des autres blocs notées *_i
// et pour les autres blocs notées *_o

      sca_tdf::sca_in<double> A_i, R_i;
      sca_tdf::sca_out<double> X_o;
      sca_tdf::sca_in<double> X_i;
      sca_tdf::sca_out<double> A_o , R_o;

// Définition des paramètres de la synthèse

      synthese( sc_core::sc_module_name nm, double ktl_ ,double ktr_,
                  double ka_ ,double kr_,double na_ ,double nr_,
                  double dx_ ,double dmx_):
                  X_o("X"), ktl(ktl_), ka(ka_), kr(kr_), ktr(ktr_),
                  na(na_), nr(nr_),dx(dx_), dmx(dmx_)
                  {}

// Ajout de délais sur le flux de signal remontant

      void set_attributes()
      {
            X_i.set_delay(1);
            A_o.set_delay(1);
            R_o.set_delay(1);
      }
```

```
// Initialisation du composant

	void initialize()
	{
		mX=0;
		X=0;
		dt = A_i.get_timestep().to_seconds();
	}

// Calcul des différentes ODEs à partir du timestep

	void processing()
	{
		A_o.write(0.0);
		R_o.write(0.0);
		A=A_i.read();R=R_i.read();

		diff_mX=((ktr/((1+pow(ka/A,na))*(1+pow(R/kr,nr))))-dmx*mX)*dt;
		mX+=diff_mX;

		diff_X=(ktl*mX-dx*X)*dt;
		X+=diff_X+X_i.read();
		X_o.write(X);
	}

SCA_CTOR(synthese) {}

	private:
	double ktl,ktr,ka,kr,na,nr,dmx,dx;
	double mX,X;
	double A,R;
	double diff_X,diff_mX;
	double dt;

};
```

Annexe E

Code SystemC-AMS des modèles ELN

E.1 Complexation

```
#include "systemc-ams.h"

// Intégration des fonctions TDF nécessaires

#include "TDF/tdf_functs.h"

// Création du bloc complexation

SC_MODULE(complexation)
{

// Déclaration des noeuds de la complexation (A, B, AB)

sca_eln::sca_terminal A,B;
sca_eln::sca_terminal AB;

// Définition des paramètres de la complexation et déclaration des composants

      complexation( sc_core::sc_module_name nm, double kon_, double koff_, double
d_=0):
                  AB("AB"),kon(kon_),koff(koff_),
d(d_),aRead("aRead"),bRead("bRead"),

      iA("iA"),iB("iB"),iAB("iAB"),rKoff("rKoff"),rAB("rAB"),mult("mult"),cAB("CAB"
),abRead("abRead"),

      cA("cA"),cB("cB"),rA("rA"),rB("rB"),invAB("invAB"),rSubB("rSubB"),rSubA("rSub
A")
      {

// Instanciation des composants
// Pour les générateurs de flux (A consommé, B consommé et AB créé),
// nécessité de passer par un bloc TDF pour les équations.
// avec des blocs de convertion en amont et en aval.

            aRead.p(A);
            aRead.n(gnd);
            aRead.outp(sA);

            rSubA.p(A);
            rSubA.n(gnd);
            rSubA.inp(sInvAB);
            rSubA.scale=(1/kon);

            iA.p(A);
            iA.n(gnd);
```

```
            iA.inp(sAB);
            iA.scale=-koff;

            cA.p(A);
            cA.n(gnd);
            cA.value=1;

            rA.p(A);
            rA.n(gnd);
            rA.value=10;

// Mêmes composants pour B

            mult.A(sA);
            mult.B(sB);
            mult.out(sMultAB);

            iAB.p(AB);
            iAB.n(gnd);
            iAB.inp(sMultAB);
            iAB.scale=(-kon);

            rKoff.p(AB);
            rKoff.n(gnd);
            rKoff.value=1/koff;

            rAB.p(AB);
            rAB.n(gnd);
            rAB.value=1/d;

            cAB.p(AB);
            cAB.n(gnd);
            cAB.value=1;

            abRead.p(AB);
            abRead.n(gnd);
            abRead.outp(sAB);

            invAB.in(sMultAB);
            invAB.out(sInvAB);

      }

private:
      sca_eln::sca_node nIntA,nIntAB;
      sca_eln::sca_node_ref gnd;

      sca_tdf::sca_signal<double> sA,sB,sAB,sInvAB,sMultAB;

      sca_eln::sca_tdf_vsink aRead,bRead,abRead;
      sca_eln::sca_tdf_isource iAB,iA,iB;
      sca_eln::sca_tdf_r rSubA,rSubB;
      sca_eln::sca_r rAB,rA,rB;
      sca_eln::sca_c cAB,cA,cB;
      multAB mult;

      inv invAB;

      double kon,koff,d;

};
```

E.2 Synthèse d'une protéine

```
#include "systemc-ams.h"

// Intégration des fonctions TDF nécessaires

#include "TDF/tdf_functs.h"

// Création du bloc synthèse

SC_MODULE(synthese)
{

// Déclaration des noeuds de la synthèse (activateur, répresseur, protéine)

sca_eln::sca_terminal act,rep;
sca_eln::sca_terminal prot;

// Définition des paramètres de la synthèse et déclaration des composants

      synthese( sc_core::sc_module_name nm, double ktl_ ,double ktr_, double ka_ ,
                  double kr_,double na_ ,double nr_,double dx_ ,double dmx_):
prot("X"),
                  ktl(ktl_), ka(ka_), kr(kr_), ktr(ktr_),na(na_), nr(nr_),dx(dx_),
dmx(dmx_),

      actRead("actRead"),repRead("repRead"),cMx("cMx"),iARN_TDF("iARN_TDF",1,1),
                  iARN("iARN"),rDmx("rDmx"),iX("iX"),rDx("rDx"),cX("cX")
      {

// Instanciation des composants
// Pour les générateurs de flux (ARN, protéine),
// nécessité de passer par un bloc TDF pour les équations.
// avec des blocs de convertion en amont et en aval.

            actRead.p(act);
            actRead.n(gnd);
            actRead.outp(sAct);

            repRead.p(rep);
            repRead.n(gnd);
            repRead.outp(sRep);

            iARN_TDF.actsIn[0](sAct);
            iARN_TDF.repsIn[0](sRep);
            iARN_TDF.out(sIARN);
            iARN_TDF.setParameters(ka, na, kr, nr);

            iARN.inp(sIARN);
            iARN.p(nIARN);
            iARN.n(gnd);
            iARN.scale=ktr;

            rDmx.p(nIARN);
            rDmx.n(gnd);
            rDmx.value=1/dmx;

            cMx.p(nIARN);
            cMx.n(gnd);
```

```
            iX.ncp(nIARN);
            iX.ncn(gnd);
            iX.np(prot);
            iX.nn(gnd);
            iX.value=ktl;

            rDx.p(prot);
            rDx.n(gnd);
            rDx.value=1/dx;

            cX.p(prot);
            cX.n(gnd);

      }

private:
      sca_eln::sca_node nIARN;
      sca_eln::sca_node_ref gnd;

      sca_tdf::sca_signal<double> sAct,sRep,sIARN;

      sca_eln::sca_tdf_vsink actRead,repRead;
      sca_eln::sca_tdf_isource iARN;
      sca_eln::sca_r rDmx, rDx;
      sca_eln::sca_c cMx,cX;
      sca_eln::sca_vccs iX;

      fActsReps iARN_TDF;

      double ktl,ktr,ka,kr,na,nr,dmx,dx;
};
```

Annexe F

Manuel utilisateur du générateur automatique de modèles

L'interface principale du logiciel est constituée de trois parties, illustrées Figure F.1.

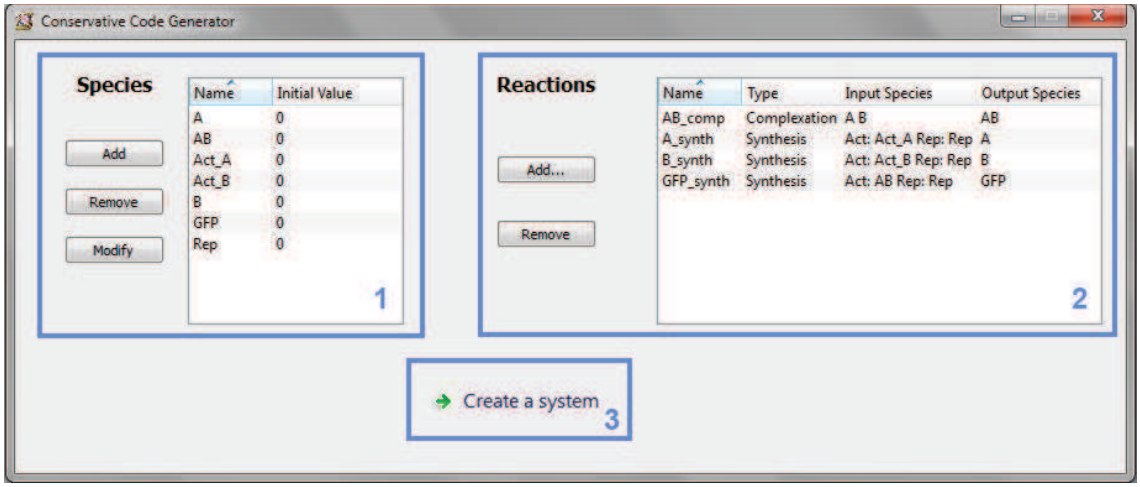


Figure F.1 : Interface du générateur automatique de modèles.

La partie 1 est consacrée à l'ajout, à la modification et à la suppression des différentes espèces impliquées dans le biosystème. En cliquant sur « Add », l'utilisateur se voit proposé la fenêtre Figure F.2, où il peut saisir le nom d'une espèce et sa concentration à l'état initial du système.

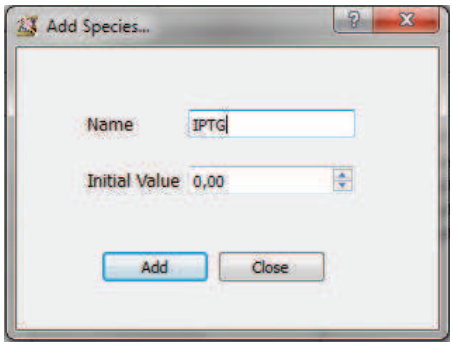


Figure F.2 : Fenêtre de saisie des espèces.

Une fois que toutes les espèces ont été saisies, l'utilisateur passe à la partie 2 du logiciel, correspondant à la saisie des différents mécanismes. Le choix des mécanismes est pour l'instant

limité à la synthèse des protéines et à la complexation. En cliquant sur « Add », l'utilisateur choisit le type de mécanisme et est amené à la fenêtre Figure F.3.

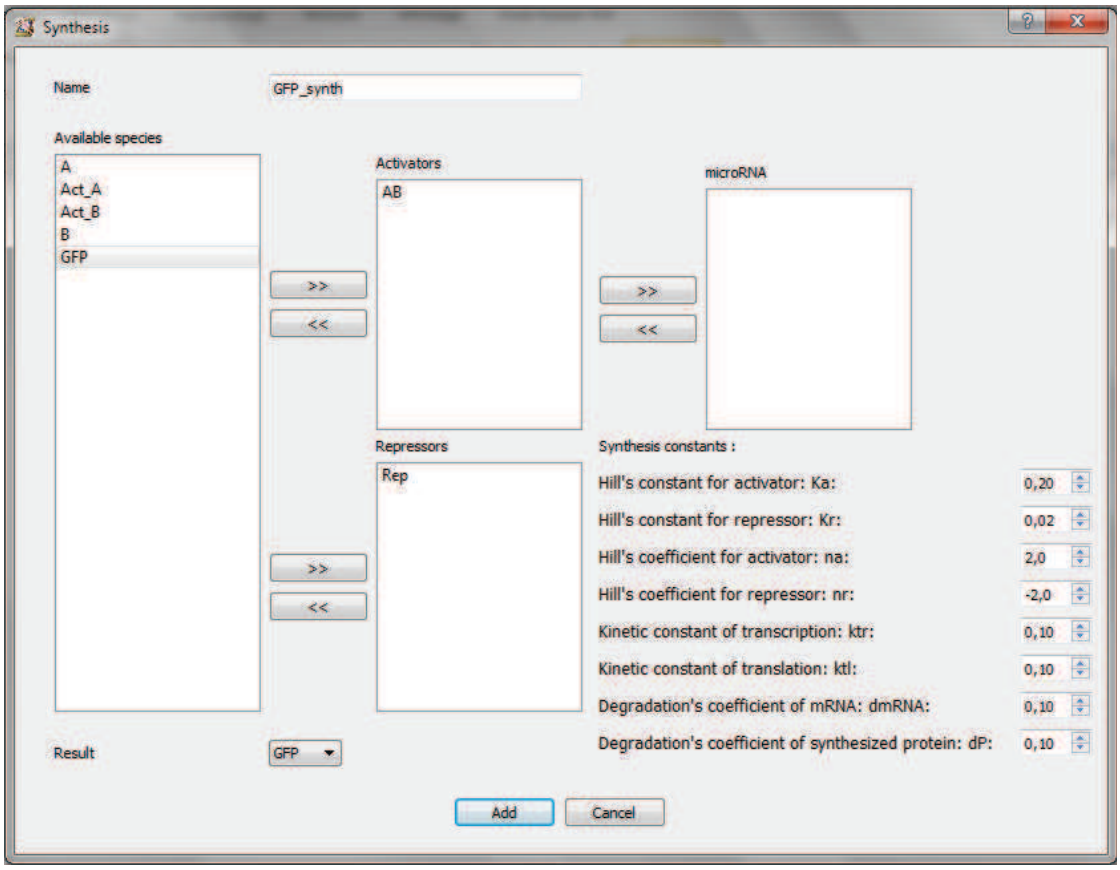


Figure F.3 : Fenêtre de saisie des mécanismes.

Dans cette fenêtre, il peut saisir le nom de la réaction, et les différentes espèces impliquées dans ce mécanisme, ainsi que les paramètres associés.

Une fois que toutes les réactions ont été saisies, l'utilisateur passe à la partie 3, qui correspond à la création du biosystème. En cliquant sur « Create a system », la fenêtre Figure F.4 s'ouvre.

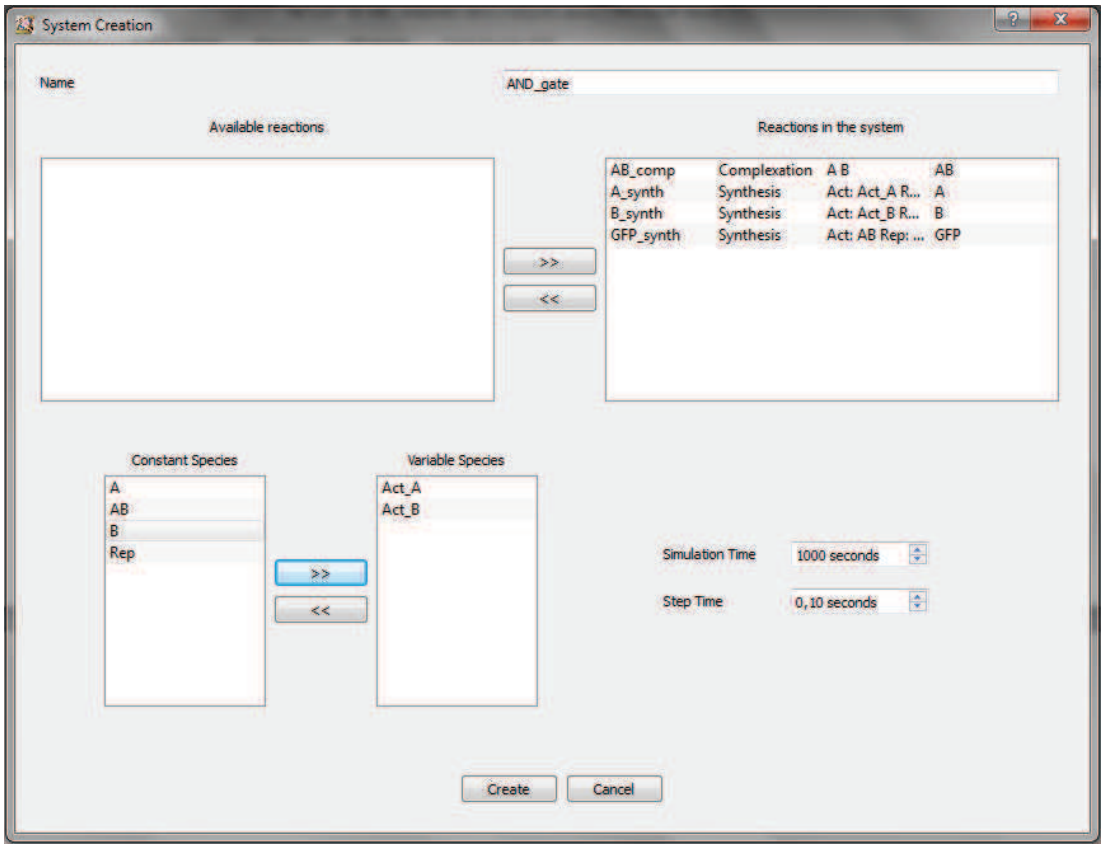


Figure F.4 : Fenêtre de saisie du biosystème.

Dans cette fenêtre, il peut saisir le nom du biosystème, les différentes réactions à intégrer, les espèces dont la concentration varie et les paramètres de simulation (le temps de simulation et le pas).

En cliquant sur « Create », les différents fichiers de modèles sont remplis en fonction des informations saisies et sont regroupés dans le dossier « generated_files ». Sous un ordinateur équipé du logiciel Smash, il ne reste plus qu'à exécuter le fichier .pat, permettant de lancer la simulation.

Annexe G

Formalisme de la netlist standardisée

G.1 Description générale

Plusieurs niveaux d'abstraction sont définis (Mécanisme biologique, Cellule, Système, Organe, Individu, …) et seuls les trois premiers sont traités actuellement. Chaque niveau d'abstraction est constitué de trois blocs :

- Le « bus biologique interne » constitué d'un condensateur, modélisant le stockage des espèces par la cellule et d'une résistance, modélisant son auto-dégradation.
- Une « netlist de mécanismes internes » tous constitués autour du même bloc élémentaire.
- Une « netlist de mécanismes d'entrée-sorties » permettant entre autre l'endocytose et l'exocytose d'espèces.

La description au niveau système se fait de la manière suivante :

```
#SYSTEM SYSTEM_LABEL {

      #SPECIES {
      …
      }

      #MECHANISMS {
      …
      }

      #CELLS {
      …
      }
}
```

G.2 Description des espèces en jeu

Quel que soit le niveau d'abstraction, les propriétés des espèces en jeu sont définies dans l'ensemble des mécanismes intervenant dans le système (les espèces utilisées pour les mécanismes internes mais aussi les espèces reçues par les sous-systèmes ou susceptibles d'y entrer). La description des espèces se fait de la manière suivante :

```
#SPECIES {

    Namespecies1 V d;
    Namespecies2 V d;
    …
}
```

Pour chaque espèce, deux paramètres sont définis :

- **V** qui correspond au volume normalisé. La valeur « 1 » correspond au volume de l'espace intracellulaire de référence. Lorsque les espèces se retrouvent dans l'espace intercellulaire, elles peuvent potentiellement se disperser mais, dû à leur mobilité réduite, elles restent néanmoins confinées dans un volume donné.
- **d** qui correspond au coefficient de dégradation de l'espèce, de base fixé à 0.1.

G.3 Description des mécanismes

A l'heure actuelle, cinq types de mécanismes peuvent être modélisés (Réaction ligand+proteine, Transcription, Traduction, Exocytose et Endocytose). La description des mécanismes se fait de la manière suivante :

```
#MECHANISM {

    Type Label_mecha1 Involved_species Parameters;
    Type Label_mecha2 Involved_species Parameters;
    …
}
```

La ligne de description diffère en fonction des mécanismes.

Réaction de complexation

La réaction de complexation est identifiée par le type (CX) et se limite dans un premier temps à un mécanisme simple A + B donne AB. Ainsi, il suffit de passer comme paramètres le nom des espèces A, B, AB ainsi que les coefficients k_{on} et k_{off} de la réaction.

```
CX Label A B AB kon koff TypeModel;
```

TypeModel permet de changer le modèle que l'on va choisir pour décrire la réaction. Par défaut, le modèle est un modèle conservatif d'ordre 1. Les autres modèles développées dans les versions futures seront : C pour conservatif, F pour fuzzy logic, N pour numérique et S pour stochastique.

Transcription génétique

Le mécanisme de transcription génétique est identifié par le type (TR). Il est plus compliqué car il peut y avoir plusieurs activateurs, plusieurs répresseurs et plusieurs ARN messager en sortie.

Pour simplifier l'utilisation du langage, la description peut se faire de deux manières : compacte ou développée.

Description développée :

```
TR Label {
    Species1 Type Param;
    Species2 Type Param;
    Species3 Type Param;
    …
} ModType;
```

Le type d'espèce est soit 'A' pour activateur, 'R' pour répresseur ou 'M' pour ARN messager. Dans les deux premiers cas les deux paramètres à passer sont les deux coefficients de l'équation de Hill, K et n. Dans le dernier cas, les deux paramètres sont k_{tr} (taux de transcription) et a (transcription libre). Tous les ARN messagers doivent avoir les même k_{tr} et a. Si ce n'est pas le cas dans la description, seules les dernières valeurs sont prises en compte.

Description compacte :

```
TR Label [A1 … An] [R1 … Rm] [M1 … Mp] [KA1 nA1, KA2 nA2, …, KR1 nR1 …] ktl a ModType;
```

Traduction protéique

La réaction de traduction est identifiée par le type (TL) et est très simple. Elle ne fait intervenir que deux espèces, l'ARN messager et la protéine codée. Elle n'est paramétrée, à l'ordre 1, que par le coefficient k_{tl}

```
TR Label M X ktl ModType;
```

Exocytose

L'exocytose est un mécanisme complexe comme la transcription défini par (EX). Elle est régulée par des activateurs et des répresseurs. Le même mécanisme de description compacte et développée que pour la traduction est défini.

Description développée :

```
EX Label {
    Species1 Type Param;
    Species2 Type Param;
    Species3 Type Param;
    …
} ModType;
```

Le type d'espèce peut donc cette fois être 'A' comme activateur, 'R' comme répresseur, 'S' comme signal (espèce à faire sortie de la cellule). Dans les deux derniers cas, il n'y a pas de paramètres (en tout cas dans la première version du modèle).

Description compacte :

```
EX Label [A1 … An] [R1 … Rm] S [KA1 nA1, KA2 nA2, …, KR1 nR1 …] ModType;
```

Endocytose

L'endocytose fonctionne exactement de la même manière que l'exocytose avec comme mécanisme (EN) à la place d'(EX).

G.4 Description des cellules

Les cellules sont des sous-blocs des systèmes. Chaque cellule se décrit avec exactement la même syntaxe que le système lui-même.

G.5 Exemple de la netlist

Le formalisme de la netlist a été appliqué sur le détecteur de cellules cancéreuses présenté dans la section 6.8.2.1. La netlist obtenue pour le sous-système permettant de détecter la présence des microARN est présentée ci-dessous :

```
#SYSTEM SYSTEM_LABEL {

   #SPECIES {
      rtTA 1 0.1;
      rtTA_RNA 1 0.1;
      LacI 1 0.1;
      LacI_RNA 1 0.1;
      miRNA_X 1 0.1;
      DsRed 1 0.1;
      DsRed_RNA 1 0.1;
      }

   #MECHANISMS {
      TR rtTA_RNA_synth [always] [miRNA_X] [rtTA_RNA] [0.1 2, 0.2 2] 0.1 0 C;
      TR rtTA_synth rtTA_RNA rtTA 0.1 C;
      TR LacI_RNA_synth [rtTA] [miRNA_X] [LacI_RNA] [0.1 2, 0.2 2] 0.1 0 C;
      TR LacI_synth LacI_RNA LacI 0.1 C;
      TR DsRed_RNA_synth [always] [LacI] [DsRed_RNA] [0.2 2, 0.2 2] 0.07 0.04 C;
      TR DsRed_synth DsRed_RNA DsRed 0.1 C;
      }

   #CELLS {
      empty
      }
   }
```

Annexe H

Code Matlab du cœur de calcul de logique floue

```
% Fonction fuzzy avec comme paramètres N, le nombre de MFs, regle, la matrice des
règles
% A et B les deux entrées avec leurs bornes et leurs échelles respectives

function [centro] = fuzzy_speed
(N,regle,A,A_min,A_max,A_scale,B,B_min,B_max,B_scale)

% Test de l'échelle et calcul du pourcentage de chaque entrée
if strcmp(A_scale,'log')
    A_pourc = (log(A)-log(A_min))/(log(A_max)-log(A_min));
else
    A_pourc = (A-A_min)/(A_max-A_min);
end

if strcmp(B_scale,'log')
    B_pourc = (log(B)-log(B_min))/(log(B_max)-log(B_min));
else
    B_pourc = (B-B_min)/(B_max-B_min);
end

x_tri = zeros(1,N+2);

MF_A = zeros(1,2);
MF_B = zeros(1,2);

% Calcul des points des MFs et enregistrement dans x_tri
for i=1:N+2
    x_tri(i)=(i-2)/(N-1);
end

% Test des MFs concernées et enregistrement dans MF[2]
n=0;
m=0;
for i=1:N
    if A_pourc > x_tri(i) && A_pourc < x_tri(i+2)
        n = n+1;
        MF_A(n)=i;
    end
    if B_pourc > x_tri(i) && B_pourc < x_tri(i+2)
        m = m+1;
        MF_B(m)=i;
    end
end

% Calcul des différentes fonctions de sortie
out_x = zeros((n*m),4);
out_y = zeros((n*m),4);

count = 0;
```

```
for i=1:m
    for j=1:n
        count=count+1;
        % Calcul du segment x entre MF et le point
        partA_x = min(A_pourc-x_tri(MF_A(j)),x_tri(MF_A(j)+2)-A_pourc);
        partB_x = min(B_pourc-x_tri(MF_B(i)),x_tri(MF_B(i)+2)-B_pourc);
        % Calcul des 4 points x pour chaque MF de sortie
        out_x(count,:)=[max(x_tri(regle(MF_B(i),MF_A(j))),0)
max((x_tri(regle(MF_B(i),MF_A(j)))+min(partA_x,partB_x)),0)
min((x_tri(regle(MF_B(i),MF_A(j))+2)-min(partA_x,partB_x)),1)
min(x_tri(regle(MF_B(i),MF_A(j))+2),1)];
        % Calcul de la valeur y associée au point x à travers la fonction triangle
        if out_x(count,2)==0
            Coo_x = -out_x(count,3);
        else
            Coo_x = out_x(count,2);
        end
        A = x_tri(regle(MF_B(i),MF_A(j)));
        B = x_tri(regle(MF_B(i),MF_A(j))+1);
        out_y(count,2) = (Coo_x-A)/(B-A);
        % Toutes les valeurs y = 0 sauf 2 (ou 1 si triangle plein)
        out_y(count,3) = out_y(count,2);
    end
end

for i=1:count
    signalx_pre = -1;
    numb(i) = 0;
    for j=1:4
        if out_x(i,j) == signalx_pre
            signaly(i,numb(i)) = max(signaly(i,numb(i)),out_y(i,j));
        else
            numb(i) = numb(i) + 1;
            signalx(i,numb(i)) = out_x(i,j);
            signaly(i,numb(i)) = out_y(i,j);
            signalx_pre = out_x(i,j);
        end
    end
end

% Boucle du calcul du max entre tous les points des différentes fonctions de sortie
l = 1;
cond = 0;
tst = 0;
minx=1.1;
for i=1:count
    xi(i) = 1;
    cond = cond + numb(i) +1;
    minx = min(minx,signalx(i,xi(i)));
end
signalfinalx(1)=minx;

while cond~=tst
    tst = 0;
    minx=1.1;
    for i=1:count
        tst = tst + xi(i);
        if xi(i)<=numb(i)
            minx = min(minx,signalx(i,xi(i)));
        end
```

```
    end

    for i=1:count
        if xi(i)<=numb(i) && minx == signalx(i,xi(i))
            xi(i) = xi(i) + 1;
        end
    end

    if signalfinalx(l) ~= minx && minx~=1.1
        l = l + 1;
        signalfinalx(l) = minx;
    end
end

for i=1:count
    ind(i)=1;
    for j=1:l
        if ind(i)>numb(i)
            signalycalc(i,j)=0;
        elseif signalfinalx(j)==signalx(i,ind(i))
            signalycalc(i,j)=signaly(i,ind(i));
            ind(i) = ind(i) + 1;
        elseif signalfinalx(j)<signalx(i,1)
            signalycalc(i,j)=0;
        elseif signalfinalx(j)<signalx(i,ind(i))
            signalycalc(i,j) = signaly(i,ind(i)-1) + (signaly(i,ind(i))-
signaly(i,ind(i)-1))/(signalx(i,ind(i))-signalx(i,ind(i)-1))*(signalfinalx(j)-
signalx(i,ind(i)-1));
        elseif signalfinalx(j)>signalx(i,ind(i))
            ind(i) = ind(i) + 1;
            j=j-1;
        end
    end
end

for i=1:l
    maxy(i) = 0;
    for j=1:count
        maxy(i) = max(maxy(i),signalycalc(j,i));
    end
    signalfinaly(i)=maxy(i);
end

oki=0;
for i=1:l
    prob=0;
    for j=1:count
        if maxy(i) == signalycalc(j,i)
            prob = prob + 1;
            sigtmp(prob)=j;
        end
    end
    if prob==1
        sigpre(i) = sigtmp(1);
    else
        for p=1:prob
            if i == 1
                if maxy(2) == signalycalc(sigtmp(p),2)
                    sigpre(i) = sigtmp(p);
                    oki=1;
                elseif oki == 0 && p==prob
```

```
                    sigpre(i) = sigtmp(p);
                end
            elseif maxy(i-1) == signalycalc(sigtmp(p),i-1) || maxy(min(i+1,l)) ==
signalycalc(sigtmp(p),min(i+1,l))
                sigpre(i) = sigtmp(p);
            end
        end
    end
    if sigpre(i)==sigpre(max(i-1,1))
        corr(i)=0;
    else
        corr(i)=1;
    end
end

k=0;
for i=1:l
    if corr(i)==1
        k = k+1;
        alpha = (signalycalc(sigpre(i),i)-signalycalc(sigpre(i),i-
1))/(signalfinalx(i)-signalfinalx(i-1));
        beta = (signalycalc(sigpre(i-1),i)-signalycalc(sigpre(i-1),i-
1))/(signalfinalx(i)-signalfinalx(i-1));
        signalfcx(k)=(signalfinalx(i-1)*alpha - signalycalc(sigpre(i),i-1) -
signalfinalx(i-1)*beta + signalycalc(sigpre(i-1),i-1))/(alpha-beta);
        signalfcy(k)= signalycalc(sigpre(i),i-1) + (signalfcx(k)-signalfinalx(i-
1))*alpha;
        k = k+1;
        signalfcx(k)=signalfinalx(i);
        signalfcy(k)=signalfinaly(i);
    else
        k = k+1;
        signalfcx(k)=signalfinalx(i);
        signalfcy(k)=signalfinaly(i);
    end
end

signalfcxtmp=signalfcx;
signalfcytmp=signalfcy;
nbt=2;

for i=3:k
    nbt=nbt+1;
    if roundn(signalfcytmp(i),-4)==roundn(signalfcytmp(i-1),-4) &&
roundn(signalfcytmp(i-1),-4)==roundn(signalfcytmp(i-2),-4)
        signalfcx([nbt-1])=[];
        signalfcy([nbt-1])=[];
        nbt=nbt-1;
    end
end

% Calcul des aires sommées à chaque point x
aire = zeros(1,(nbt-1));
for i=1:(nbt-1)
    aire(1,i) = aire(1,max(i-1,1)) + (signalfcx(i+1)-
signalfcx(i))*(signalfcy(i+1)+signalfcy(i))/2;
end
% Test pour déterminer les point x ou l'aire dépasse 50%
for pt=1:(nbt-1)
    if aire(1,pt)>(aire(1,(nbt-1))/2)
        break;
```

```
    end
end

% Calcul du poucentage d'aire du trapèze pour déterminer le centroide
if pt==1
    pourc = 0.5/((aire(1,pt))/aire(1,(nbt-1)));
else
    pourc = (0.5-(aire(1,pt-1)/aire(1,(nbt-1))))/((aire(1,pt)-aire(1,pt-
1))/aire(1,(nbt-1)));
end

coeff_a = (signalfcy(pt+1)-signalfcy(pt))/(signalfcx(pt+1)-signalfcx(pt));

if roundn(coeff_a,-4) == 0
    centro=pourc*(signalfcx(pt+1)-signalfcx(pt))+signalfcx(pt);
else
    coeff_b = 2*signalfcy(pt);

    coeff_c = -pourc*(signalfcx(pt+1)-
signalfcx(pt))*(signalfcy(pt)+signalfcy(pt+1));

    delta = coeff_b^2 - 4*coeff_a*coeff_c;

    centro1 = (-coeff_b-sqrt(delta))/(2*coeff_a) + signalfcx(pt);

    centro2 = (-coeff_b+sqrt(delta))/(2*coeff_a) + signalfcx(pt);

    if centro1 >= signalfcx(pt) && centro1 <= signalfcx(pt+1)
        centro = centro1;
    else
        centro = centro2;
    end
end

% Normalisation des bornes entre 0 et 1
norm = (2-sqrt(2))/(2*(N-1));

if centro <= 0.5
    centro = centro - norm*((0.5-centro)/(0.5-norm));
else
    centro = centro + norm*(1-(((1-norm)-centro)/((1-norm)-0.5)));
end
```

Annexe I

Liens entre le polynôme de liaison et l'équation de Hill

Dans cette annexe, l'approximation du polynôme de liaison afin de retrouver l'équation de Hill est généralisée pour r ligands. Cette approximation dépend de l'hypothèse effectuée sur la coopérativité entre les différents sites pour les ligands. Nous allons voir les cas où les sites ont une forte coopérativité positive et le cas où les sites sont indépendants.

I.1 Coopérativité positive forte

Nous repartons du polynôme de liaison en fixant tous les k identiques ($\forall x, k_x = k$). Nous avons la formule compacte permettant d'obtenir le polynôme de liaison pour n ligands du même type, défini par :

$$p(x) = 1 + \sum_{i=1}^{n} c_i \cdot (k \cdot x)^i \mathrm{C}_n^i \tag{I.1}$$

où c_i est le facteur de couplage entre le $i^{\text{ièmes}}$ sites ($c_i = c_{i_1,\dots,i_n} | \sum_k i_k = i$). L'hypothèse de forte coopérativité positive se traduit par la condition suivante :

$$[PL_{i+1}] \gg [PL_i] \tag{I.2}$$

L'équation (I.2) équivaut à :

$$\frac{[PL_{i+1}]}{P[L]^{i+1}} \gg \frac{[PL_i]}{P[L]^{i+1}} \Leftrightarrow \beta_{i+1} \gg \frac{\beta_i}{[L]} \tag{I.3}$$

Comme $\beta_{i+1} = c_{i+1} \cdot k^{i+1}$ et $\beta_i = c_i \cdot k^i$, l'équation (I.3) devient :

$$c_{i+1} \cdot k \cdot x \gg c_i \tag{I.4}$$

avec $x = [L]$. Le polynôme de liaison de l'équation (I.1) peut donc s'approximer comme :

$$p(x) \cong 1 + c_n \cdot (k \cdot x)^n \tag{I.5}$$

Le numérateur du signal total est donné par la formule compacte suivante :

$$\vartheta_i(x) = \sum_{i=1}^{n} i \cdot c_i \cdot (k \cdot x)^i C_n^i \tag{I.6}$$

De la même manière, avec l'hypothèse de coopération positive, nous obtenons l'approximation de l'équation (I.6) suivante :

$$\vartheta_i(x) \cong n \cdot c_n \cdot (k \cdot x)^n \tag{I.7}$$

Et finalement, en prenant le signal molaire $\sigma_i = 1$, le signal total S(x) est donné par :

$$S(x) = \frac{\vartheta_i(x)}{p(x)} = \frac{n \cdot c_n \cdot (k \cdot x)^n}{1 + c_n \cdot (k \cdot x)^n} = \frac{n}{1 + \frac{1}{c_n \cdot (k \cdot x)^n}} \tag{I.8}$$

S(x) peut être identifié avec l'équation de Hill définit par :

$$\frac{a}{1 + \left(\frac{K}{x}\right)^{n_H}} \tag{I.9}$$

avec *K* le paramètre de Hill valant :

$$K = \frac{1}{\sqrt[n]{c_n} \cdot k} \tag{I.10}$$

et le coefficient de Hill n_H valant donc n, le nombre de ligands, dans cette approximation. En pratique nous constatons qu'à partir du moment où le coefficient de Hill est supérieur à 1, cela correspond à une coopérativité positive. Le cas où le coefficient de Hill tend vers n correspond à une coopérativité très fortement positive.

I.2 Sites indépendants

Dans cette hypothèse, nous repartons des formules (I.1) et (I.6) mais cette fois-ci en ayant les facteurs de couplages c_i égaux à 1, car la coopérativité entre les sites est nulle. Le polynôme de liaison peut donc s'approximer de la façon suivante :

$$p(x) \cong (1 + k \cdot x)^n \tag{I.11}$$

Pour le numérateur du signal total $\vartheta_i(x)$ l'approximation est légèrement plus complexe à réaliser. Nous partons donc de l'équation (I.6) qui peut se simplifier :

$$\begin{aligned}\vartheta_i(x) &= \sum_{i=1}^{n} i \cdot (k \cdot x)^i \mathrm{C}_n^i = \sum_{i=0}^{n-1} (i+1) \cdot (k \cdot x)^{i+1} \mathrm{C}_n^{i+1} \\ &= n \cdot k \cdot x \cdot \sum_{i=0}^{n-1} \frac{(i+1)}{n} \cdot \mathrm{C}_n^{i+1} \cdot (k \cdot x)^i\end{aligned} \tag{I.12}$$

Or nous avons la propriété sur les combinaisons suivante :

$$\frac{(i+1)}{n} \cdot \mathrm{C}_n^{i+1} = \frac{n!}{(i+1)! \cdot \big(n-(i+1)\big)!} \cdot \frac{(i+1)}{n} = \frac{(n-1)!}{i! \cdot (n-1-i)!} = \mathrm{C}_{n-1}^i \tag{I.13}$$

L'équation (I.13) devient donc :

$$\vartheta_i(x) = n \cdot k \cdot x \cdot \sum_{i=0}^{n-1} \mathrm{C}_{n-1}^i \cdot (k \cdot x)^i \cong n \cdot k \cdot x \cdot (1 + k \cdot x)^{n-1} \tag{I.14}$$

Le signal total S(x), en prenant le signal molaire $\sigma_i = 1$, équivaut à :

$$S(x) = \frac{\vartheta_i(x)}{p(x)} = \frac{n \cdot k \cdot x \cdot (1 + k \cdot x)^{n-1}}{(1 + k \cdot x)^n} = \frac{n \cdot k \cdot x}{1 + k \cdot x} = \frac{n}{1 + \frac{1}{k \cdot x}} \tag{I.15}$$

S(x) peut à nouveau être identifié avec l'équation de Hill, avec K le paramètre de Hill valant :

$$K = \frac{1}{k} \tag{I.16}$$

Le coefficient de Hill n_H vaut donc 1 dans cette approximation, ce qui correspond bien à une coopérativité nulle soit des sites indépendants.

Annexe J

Manuel utilisateur du générateur de polynôme

L'interface principale du générateur de polynômes de liaison est divisée en six parties, illustrées Figure J.1.

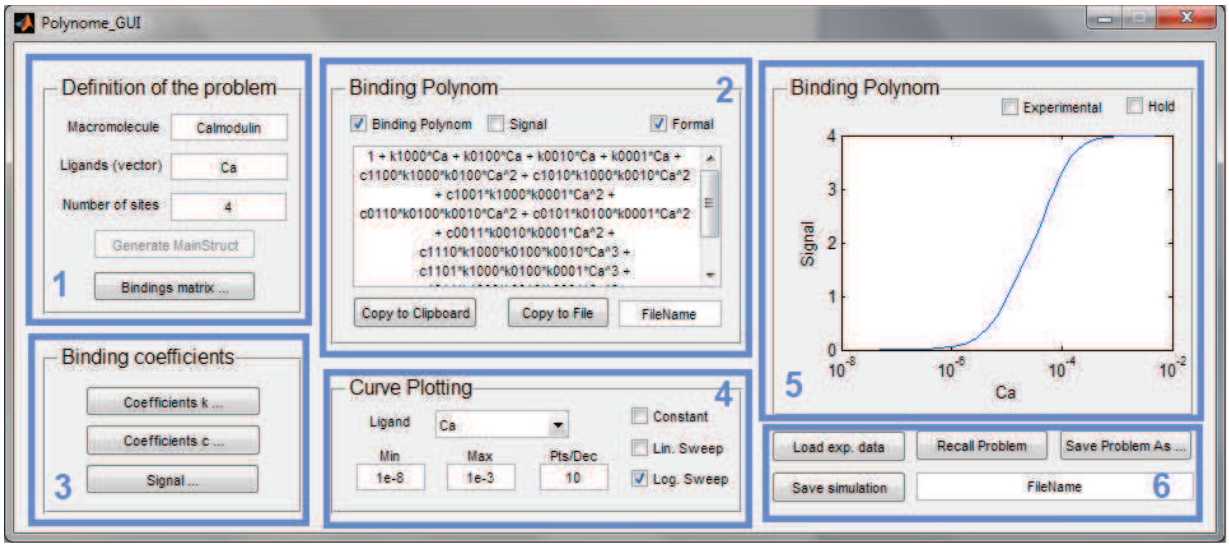


Figure J.1 : Interface principale du générateur de polynômes.

Pour obtenir le polynôme, l'utilisateur doit dans un premier temps, définir le système qu'il veut modéliser. Pour ce faire il saisit le nom de la macromolécule, les noms des différents ligands ainsi que le nombre de sites présents sur la macromolécule dans la partie 1 « Definition of the problem ». Dans le cas de la calmoduline, nous saisissons la calmoduline comme macromolécule, avec 4 sites de liaison, et comme ligands les ions calcium. En pressant sur le bouton « Bindings matrix », l'utilisateur peut modifier les sites sur lesquels les différents ligands peuvent se lier, ce qui est illustré Figure J.2. Dans l'exemple utilisé, les ions calcium peuvent se lier sur les quatre sites.

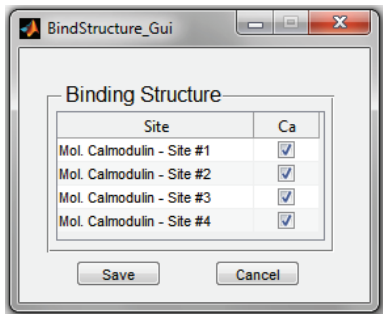


Figure J.2 : Sélection des sites de liaison.

Enfin, pour obtenir le polynôme, l'utilisateur clique sur « Generate MainStruct », ce qui génère le polynôme dans l'encadré 2, « Binding Polynom ». L'utilisateur peut ensuite récupérer le polynôme généré ainsi que le signal, afin de les exploiter dans un programme externe en les copiant dans le presse-papier ou en les sauvegardant dans un fichier.

Pour visualiser la courbe extraite à partir du polynôme représentant une fonction dépendant des concentrations des différents ligands dans la partie 5, l'utilisateur doit d'abord remplir les valeurs des coefficients du polynôme, à savoir les constantes d'association, les facteurs de couplage et les signaux associés à chaque complexe. Pour ce faire, et pour ensuite pouvoir les modifier, il faut cliquer sur les boutons correspondant dans l'encadré 3, « Binding coefficients ». Par exemple pour les facteurs de couplage, nous obtenons la fenêtre Figure J.3.

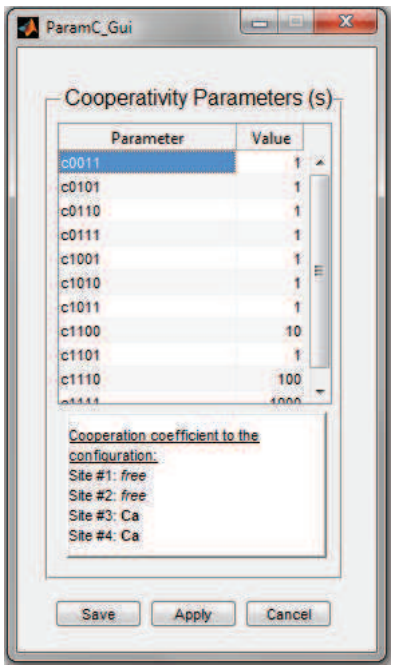


Figure J.3 : Fenêtre de saisie des facteurs de couplage.

En sélectionnant un coefficient, sa correspondance avec les sites affectés est toujours rappelée en bas de la fenêtre pour simplifier la compréhension. Les différents coefficients sont saisis en

fonction des hypothèses sur le modèle. Dans le cas présenté, nous saisissons l'hypothèse de liaison séquentielle.

Ensuite dans l'encadré 4 « Curve plotting » il reste à sélectionner le ligand en fonction duquel la courbe sera tracée (ici les ions calcium) et à ajuster l'évolution de la concentration de celui-ci, qui peut être constante ou avec une évolution linéaire ou logarithmique.

Dans les encadrés 5 et 6, nous obtenons la courbe du signal total. Les options disponibles sont le maintien de la courbe en cours afin de voir l'impact de la modification des coefficients, le chargement et l'affichage d'une série de points extérieurs au programme ainsi que la sauvegarde ou le chargement du système saisi, ou uniquement les résultats de simulation, dans un fichier.

Structuration d'un flot de conception pour la biologie synthétique

-

Structuring the design flow for synthetic biology

Résumé : La biologie synthétique est une science issue du rapprochement entre les biotechnologies et les sciences pour l'ingénieur. Elle consiste à créer de nouveaux systèmes biologiques par une combinaison rationnelle d'éléments biologiques standardisés, découplés de leur contexte naturel. L'environnement, l'agroalimentaire et la santé figurent parmi ses principaux domaines d'application. Cette thèse s'est focalisée sur les aspects liés à la conception ex-vivo de ces biosystèmes artificiels. A partir des analogies réalisées entre les processus biologiques et certaines fonctions électroniques, l'accent a été mis sur la réutilisation et l'adaptation des outils de conception numériques, supportant l'approche de conception « top-down ». Ainsi, une adaptation complète des méthodes de CAO de la microélectronique a été mise en place pour la biologie synthétique. Dans cette optique, les mécanismes biologiques élémentaires ont été modélisés sous plusieurs niveaux d'abstraction, allant de l'abstraction numérique à des modèles flux de signal et des modèles conservatifs. Des modèles en logique floue ont aussi été développés pour faire le lien entre ces niveaux d'abstraction. Ces différents modèles ont été implémentés avec deux langages de description matérielle et ont été validés sur la base de résultats expérimentaux de biosystèmes artificiels parmi les plus avancés. Parallèlement au travail de formalisation des modèles destinés au flot de conception, leur amélioration a aussi été étudiée : la modélisation des interactions entre plusieurs molécules a été rendue plus réaliste et le développement de modèles de bruits biologiques a également été intégré au processus. Cette thèse constitue donc une contribution importante dans la structuration et l'automatisation d'étapes de conception pour les biosystèmes synthétiques. Elle a permis de tracer les contours d'un flot de conception complet, adapté de la microélectronique, et d'en mettre en évidence les intérêts.

Mots-clés : biologie synthétique ; biosystèmes ; modélisation compacte ; simulation de circuits ; langages de description matérielle ; flot de conception ; multiples niveaux d'abstraction.

Abstract: Synthetic biology is a science derived from the rapprochement between biotechnology and engineering science. It aims to create new biological systems through a rational combination between standardized biological elements which are disconnected from their natural context. Its main areas of application are the environment, the food-processing industry and the health sector. This thesis focuses on the ex vivo design aspects of these artificial biosystems. Thanks to analogies between biological processes and some electronic functions, the emphasis was put on reusing and adapting digital design tools that are fitting the top-down design approach. Thus, microelectronics CAD methods have been completely adapted to synthetic biology. In this regard, basic biological mechanisms have been modelled with various levels of abstraction, from digital abstraction to signal flow and conservative models. Fuzzy logic models have also been developed as a link between these levels of abstraction. These models have been implemented with two hardware description languages. They have been proven correct thanks to experimental results from state-of-the-art artificial biosystems. Concurrently to their formalization, improvements of design flow models have been studied: the modelling of interactions between several molecules have been made more realistic and the development of models for biological noise have been integrated to the process. This thesis is an important contribution to the structuring and the automation of some design steps for synthetic biosystems. It has made possible to highlight and to trace the outlines of a complete design flow, adapted from microelectronics.

Keywords: synthetic biology; biosystems; compact modeling; circuit simulation; hardware description languages; design flow; multiple levels of abstraction.