

N° d'ordre :

Habilitation à Diriger des Recherches

Spécialité : Informatique

présentée devant

L'Université de Rennes I

**Institut de Formation Supérieure
en Informatique et en Communication**

par

Vincent RODIN

Équipe d'Accueil 3883

Laboratoire d'Informatique des Systèmes Complexes (LISyC)

Ecole Nationale d'Ingénieurs de Brest (ENIB)

Centre Européen de Réalité Virtuelle (CERV)

Titre :

Contribution à l'utilisation de l'informatique en biologie

Soutenue le 6 décembre 2004 devant le jury composé de :

<i>Président :</i>	Mr	Jean-François	ABGRALL	Pr. Université de Brest, UBO/CHU Brest
<i>Rapporteurs :</i>	Mme	Catherine	GARBAY	DR CNRS, TIMC Grenoble
	Mr	Bernard	MANDERICK	Pr. Vrije Universiteit Brussel
	Mr	Christian	PELLEGRINI	Pr. Université de Genève
<i>Examineurs :</i>	Mr	Stéphane	DONIKIAN	CR CNRS – HDR, IRISA Rennes
	Mr	Jean	LE BIHAN	Pr. ENI Brest
	Mr	Jacques	TISSEAU	Pr. ENI Brest

Contribution à l'utilisation de l'informatique en biologie

Document de synthèse

VINCENT RODIN

Équipe d'Accueil 3883

Laboratoire d'Informatique des Systèmes Complexes (LISyC)

Ecole Nationale d'Ingénieurs de Brest (ENIB)

Centre Européen de Réalité Virtuelle (CERV)

**École Nationale d'Ingénieurs de Brest
Laboratoire d'Ingénierie Informatique
Centre Européen de Réalité Virtuelle**

Technopôle Brest-Iroise
Site de la Pointe du Diable
Parvis Blaise PASCAL
Adresse postale : C.S. n° 73862, 29238 Brest Cedex 3, France

Vincent Rodin

e-mail : rodin@enib.fr
tel : +33 (0)2 98 05 89 50
fax : +33 (0)2 98 05 89 79

Remerciements

A Claudine, David et Mickaël.

Je remercie vivement Catherine Garbay, Directrice de Recherches au CNRS, co-responsable du laboratoire TIMC de Grenoble, d'avoir accepté d'être rapporteur de ce document et d'avoir participé à mon jury.

Je remercie tout aussi vivement, et pour les mêmes raisons, Bernard Manderick, Professeur co-responsable du *Computational Modeling Lab* de l'Université flamande *Vrije Universiteit Brussel* et Christian Pellegrini, Professeur responsable du Laboratoire d'Intelligence Artificielle de l'Université de Genève.

Je remercie également Stéphane Donikian, Chargé de Recherches au CNRS-HDR, membre de l'équipe SIAMES de l'IRISA de Rennes, d'avoir accepté d'être membre de mon jury.

Mes remerciements vont également, et tout naturellement, à Jean Le Bihan, Professeur responsable du Laboratoire RESO de l'ENI de Brest et Jean-François Abgrall, Professeur d'Hématologie au CHU de Brest. Ils ont en effet accepté de participer à mon jury et c'est avec eux que j'ai effectué la majorité de mes travaux.

Je tiens tout particulièrement à remercier Jacques Tisseau, responsable du Centre Européen de Réalité Virtuelle et Professeur à l'ENI de Brest, de m'avoir accueilli dans son laboratoire il y a maintenant environ dix ans, et d'avoir participé à mon jury. Je le remercie bien évidemment aussi pour sa confiance et ses nombreux conseils toujours très opportuns.

Merci également à tous mes collègues de l'ENI de Brest et du CERV.

Table des Matières

Remerciements	iii
Table des Matières	v
Liste des Figures	vii
1 Introduction	1
2 Modélisation et simulation de systèmes physiologiques humains	7
2.1 La simulation comme outil d'investigation du vivant	7
2.2 Notre démarche de modélisation	10
2.2.1 Caractéristiques des systèmes biologiques	10
2.2.2 Caractéristiques des systèmes multi-agents	11
2.2.3 De l'agent-cellule aux phénomènes	13
2.3 Agent-cellule	14
2.3.1 Construction d'un agent-cellule	14
2.3.2 Comportement d'un agent-cellule	16
2.3.3 Interactions entre agents-cellules, agents-molécules	16
2.3.4 Exemple de comportement interne d'un agent-cellule	18
2.3.5 Intérêt et utilisation des agents-cellules	23
2.4 Agent-réaction	23
2.4.1 Cinétique des réactions biochimiques	24
2.4.2 Comportement des agents-réactions	27
2.4.3 Intérêt et utilisation des agents-réactions	31
2.5 Agent-interaction et phénomène biologique	33
2.5.1 Réification des interactions	33
2.5.2 Composition de phénomènes	34
2.5.3 Intérêt et utilisation des agents-interactions	35
2.6 Applications	37
2.6.1 Applications en immunologie	37

2.6.2	Applications en hématologie	43
2.6.3	Application en cancérologie : le cas du myélome multiple	50
2.7	Conclusion sur la modélisation et la simulation de systèmes biologiques	56
3	Système multi-agents <i>immunitaire</i> auto-régulé pour le traitement d'images biologiques	57
3.1	Introduction	57
3.2	Langage et simulateur oRis	61
3.2.1	Concepts principaux d'oRis	61
3.2.2	Communication entre agents	63
3.2.3	Simulateur oRis	65
3.3	Applications multi-agents de traitement d'images	68
3.3.1	Système multi-agents pour la détection de stries concentriques	68
3.3.2	Système multi-agents pour la détection de spots	72
3.4	Influence des paramètres des agents	73
3.4.1	Paramètre θ	75
3.4.2	Utilisation des critères de Canny	76
3.4.3	Paramètre L	77
3.4.4	Robustesse de la méthode et application aux images d'otolithes	79
3.5	Arrêt de notre système multi-agents par auto-régulation	81
3.5.1	Principes immunitaires pour la régulation de systèmes multi-agents	83
3.5.2	Structure générale de notre système multi-agents immunitaire	86
3.5.3	Application au traitement d'images d'otolithes	86
3.5.4	Intérêt de la régulation	88
3.6	Conclusion sur le traitement d'images par agents auto-régulés	89
4	Conclusion et perspectives	91
	Références Bibliographiques	93

Liste des Figures

1.1	Images numériques endoscopiques : un utérus à gauche et un colon à droite.	1
2.1	Création d'un agent-cellule vide.	15
2.2	Création d'un agent-cellule complet.	15
2.3	Comportements de base prédéfinis.	16
2.4	Fusion et division d'agents composés.	17
2.5	Déplacement des agents : translation, rotation.	17
2.6	Exemple de carte cognitive.	19
2.7	Définition d'une carte cognitive floue [145].	20
2.8	Association d'un agent à un Graphe d'Influence Flou.	21
2.9	a: Modèle de Huang et Ferell ; b: Modèle de Kholodenko.	21
2.10	a: GIF du modèle de Huang et Ferell ; b: GIF du modèle de Kholodenko.	21
2.11	Résultats obtenus par Blüthgen [18]. a: modèle de Huang et Ferell ; b: modèle de Kholodenko.	22
2.12	Résultats obtenus avec des GIF. a: modèle de Huang et Ferell avec N=1 agent-cellule ; b: modèle de Kholodenko avec N=1 agent-cellule ; c: modèle de Huang et Ferell avec N=1000 agents-cellules (moyenne des concentra- tions) ; d: modèle de Kholodenko avec N=1000 agents-cellules (moyenne des concentrations).	22
2.13	Modèle de Henri-Michaelis-Menten.	25
2.14	Vitesse initiale d'une réaction enzymatique en fonction de la concentration en substrat. La vitesse initiale évolue selon une branche d'hyperbole en fonction de la concentration en substrat.	26
2.15	Cycle de vie de chaque agent-réaction.	28
2.16	Code associé au cycle de vie de chaque agent-réaction de type réaction enzymatique.	29
2.17	Code associé au cycle de vie de chaque agent-réaction de type formation de complexes.	29
2.18	Simulation du modèle de Kholodenko : (a) agents-réactions ; (b) ODE.	30

2.19	Evolution de la concentration en MAPK-PP avec suppression, puis restauration, de la rétro-inhibition.	31
2.20	Modélisation à base d'ODE <i>vs.</i> modélisation à base d'agents-réactions. . .	32
2.21	Réification des interactions : le schéma de gauche représente la méthode classique de modélisation d'un système avec des nœuds "composants" et des arcs "interactions" ; le schéma de droite représente notre approche dans laquelle les arcs sont transformés en nœuds et inversement. Sur cet exemple, il est possible qu'une interaction ne soit liée qu'à un seul composant. . . .	34
2.22	Diagramme UML (Unified Modeling Language) décrivant la notion de phénomène et d'organisation.	35
2.23	Un phénomène est un ensemble quelconque de composants, d'interactions et donc potentiellement de sous-phénomènes. Ici, Phénomène B est composé de Phénomènes A.1 et A.2.	36
2.24	Association non hiérarchique de phénomènes.	36
2.25	De phénomènes pères en en phénomènes fils, il est possible de décrire un biotope. Chaque nœud du graphe représente les interactions du sous-ensemble avec l'environnement.	36
2.26	Modélisation d'un morceau de peau en vue de simuler un urticaire.	37
2.27	Modèle biologique coopératif de la réponse humorale [49].	39
2.28	Modèle multi-agents coopératif de la réponse humorale.	40
2.29	Comparaison entre l'expérimentation <i>in vitro</i> et <i>in virtuo</i> de l'évolution au cours du temps du nombre d'antigènes et du nombre d'anticorps lors d'une réponse humorale.	41
2.30	Modèle du lymphocyte B-CD5 et des agents molécules.	42
2.31	Evolution des populations des agents B-CD5 avec et sans introduction d'anti-CD5.	43
2.32	Les interactions mises en œuvre dans nos modèles de coagulation.	45
2.33	Courbe de génération de thrombine obtenue <i>in vitro</i> par sous-échantillonnage, d'après Hendrik Coenraad Hemker [62].	46
2.34	Vue de dessus de la paroi de la veine.	47
2.35	Diverses simulations de coagulation : patient sain, patients malades (hémophilie A, hémophilie B, maladie de Leiden).	47
2.36	Diverses simulations de traitement : hémophilie B traitée avec du NovoSeven et maladie de Leiden traitée avec de l'héparine.	48
2.37	Diverses simulations de coagulation : patient sain, patients malades (hémophilie A, hémophilie B et maladie de Leiden).	49
2.38	Diverses simulations de coagulation : hémophilie B traitée avec du NovoSeven et maladie de Leiden traitée avec de l'héparine.	50
2.39	Evolution de la maladie en fonction de l'expression de CD45.	51
2.40	Rôle du CD45 dans les myélomes.	52
2.41	Modèle de cellule myélomateuse : GIF pour le <i>signalling</i> IGF-1/Il6.	53
2.42	Connexion d'un agent à un GIF.	54
2.43	Modèle de cellule myélomateuse : dimérisation.	54
2.44	Evolution de la tumeur.	55

2.45	Evolution des populations de cellules exprimant plus ou moins CD45. Cette figure présente différentes courbes : le nombre total de cellules myéломateuses, le nombre de cellules exprimant fortement CD45 (CD45 Bright), le nombre de cellules exprimant faiblement CD45 (CD45 Low) et le nombre de cellules n'exprimant pas CD45 (CD45 Neg).	55
3.1	(a) Modèle BRIC de notre système multi-agents pour le traitement d'images, (b) Modèle BRIC de nos agents.	58
3.2	(a) Un otolithe de plie de 8 ans – (b) Un gel d'électrophorèse 2D.	59
3.3	Un très simple programme orienté agent en oRis.	63
3.4	Deux exemples de redéfinition de méthode au niveau instance.	64
3.5	Dédoublément du flot d'exécution en oRis.	66
3.6	Structure de la machine virtuelle oRis.	67
3.7	Exemples de textures présentes dans un anneau de croissance.	69
3.8	(a) Modèle BRIC de notre système multi-agents, (b) Modèle BRIC d'un agent.	70
3.9	(a) Capteurs d'un agent, (b) Automate décrivant le comportement d'un agent.	70
3.10	Exemples de résultats obtenus avec des agents simples et des agents de "haut niveau".	72
3.11	(a) Les capteurs d'un agent, (b) Automate décrivant le comportement d'un agent.	73
3.12	Une image et les spots détectés ; les spots sont remplis pour une meilleure visibilité.	74
3.13	Deux combinaisons différentes de (L, θ) pour le même w	75
3.14	Chemin d'un agent. (a) $\theta = \pi$, $L = 2$; (b) $\theta = \pi/4$, $L = 2$	76
3.15	(a) Image d'un contour en toit (largeur 3 pixels, amplitude 60 niveaux de gris) noyé dans un bruit gaussien d'écart-type 2 (après une égalisation d'histogramme pour une meilleure visibilité) ; (b) Chemins des agents sur cette image avec $L = 2$ (niveaux de gris inversés).	77
3.16	Histogramme de l'image des chemins des agents.	77
3.17	Chemins des agents. (a) $L = 2$; (b) $L = 6$	78
3.18	(a) Image bruitée définie par Ziou (après une égalisation d'histogramme) ; (b) Chemins des agents, $L = 2$, niveaux de gris inversés ; (c) Chemins des agents après un seuillage automatique.	79
3.19	(a) Chemins des agents pour différentes valeurs de L ; (b) Chemins après seuillage.	80
3.20	Résultats obtenus avec des otolithes de 1 à 8 ans.	80
3.21	Les deux principales phases d'une réponse immunitaire.	84
3.22	Schéma logique du déclenchement d'un comportement de type activation ou apoptose. Le seuil de maturité est ajusté au cours du temps en fonction de l'utilité de l'agent dans le système.	86
3.23	Système de régulation immunitaire pour systèmes multi-agents.	87
3.24	Evolution du nombre d'agents en fonction du temps.	88
3.25	Comparaison de la qualité du traitement avec et sans régulation.	89
3.26	Durée du traitement avec et sans régulation.	89

Chapitre 1

Introduction

Au début des années 1990, alors que je préparais mon doctorat, je travaillais sur le traitement informatique d'images endoscopiques. L'endoscopie a débuté durant les années 1960. A cette époque, la fibroscopie avec des endoscopes à fibres de verre donnait des résultats tout juste acceptables. Depuis la fin des années 1980, l'utilisation des endoscopes électroniques avec leurs capteurs CCD au bout de la sonde, permet de saisir des images directement à l'intérieur du corps humain au lieu de simplement les transmettre avec toute la déperdition de résolution qui en résultait. La figure 1.1 présente deux exemples d'images obtenues avec un endoscope électronique. L'acquisition numérique des images endoscopiques a alors ouvert la voie de l'utilisation de l'informatique comme aide lors de l'endoscopie opératoire. Lors de mes travaux de recherche pour ma thèse, soutenue en 1993, j'ai plus particulièrement travaillé sur la reconstruction tridimensionnelle de la cavité observée par endoscopie et sur l'aide au diagnostic avec la détection automatique de zones cancéreuses par traitement d'images et apprentissage [124, 125]. Parallèlement à ces travaux, les premiers simulateurs d'opérations endoscopiques apparaissent afin de contribuer à la formation des chirurgiens [52]. Aujourd'hui, la tendance est à l'utilisation de techniques opératoires utilisant la réalité augmentée (*i.e.* superposition d'images de synthèse sur les images issues de l'endoscope) afin de rendre l'acte chirurgical encore plus fiable et, donc, de sauver encore plus de vies humaines [26, 110].

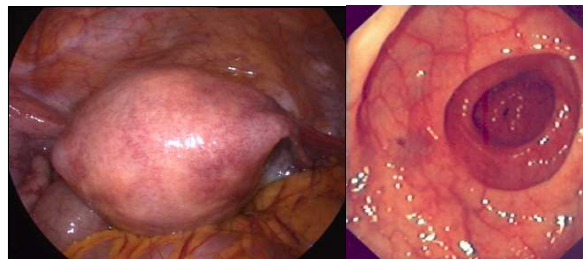


Figure 1.1: Images numériques endoscopiques : un utérus à gauche et un colon à droite.

Depuis 1995, mes travaux de recherche se déroulent dans le Laboratoire d'Ingénierie Informatique de l'Ecole Nationale d'Ingénieurs de Brest. Ils se situent dans la continuité de mes travaux de thèse, c'est-à-dire sur l'utilisation de l'informatique en biologie.

L'informatique et la biologie sont deux domaines qui semblent *a priori* décorrelés. L'un est formel, manipulant des données numériques et symboliques, l'autre est expérimental et sujet aux aléas du vivant. Contrairement aux apparences, des liens étroits existent entre ces deux disciplines. Ces liens sont très probablement aussi forts que ceux qui lient la physique et les mathématiques [70]. Comme nous le verrons dans la suite de ce document, une avancée de la recherche en informatique peut avoir des répercussions en biologie et inversement, la biologie peut fournir un large éventail de métaphores afin de trouver de nouvelles solutions à des problèmes purement informatiques. Les réseaux de neurones, les algorithmes génétiques et la programmation par essaim (*ant programming*) sont certainement les métaphores biologiques les plus connues et les plus utilisées en informatique.

L'association de l'informatique et de la biologie n'est pas nouvelle. Elle a en effet débuté dans les années 1960 avec l'utilisation d'ordinateurs en biomathématique. Le terme *biomathématique* lui-même est bien plus ancien et date de 1923. A cette époque, le Docteur William Moses Feldman a constaté un manque flagrant d'outils de calcul et d'analyse de données pour les chercheurs en médecine et en biologie. Il a alors écrit un traité intitulé *Biomathematics* à leur intention [41]. Ainsi, l'objectif premier des biomathématiques est le traitement de l'information dans les sciences de la vie et de la santé en employant des outils mathématiques.

Aujourd'hui, les domaines d'applications des biomathématiques sont très variés. Citons par exemple, et de façon non exhaustive, l'épidémiologie, la modélisation d'activités médicales, l'étude de systèmes dynamiques, les sciences cognitives, la phylogénie moléculaire ou encore l'analyse d'images biologiques. Les biomathématiques visent donc actuellement à résoudre des problèmes en biologie en utilisant les mathématiques, les statistiques et l'informatique. Les champs d'applications des biomathématiques recouvrent ainsi partiellement ceux d'une autre discipline plus récente, la bioinformatique.

Le terme *bioinformatique* n'est pas, remarquons le, d'origine anglo-saxonne. Ce mot semble en effet être passé du français à l'anglais en donnant le mot *bioinformatics*. Il est difficile de dater avec précision l'apparition de ce mot. Certains auteurs le datent du milieu des années 1980 [9, 53]. Il désigne alors toute utilisation de l'informatique en biologie [45, 94]. D'autres auteurs estiment au contraire que le mot bioinformatique n'est pas apparu avant le début des années 1990 [19, 29]. Il n'était alors utilisé que dans le contexte de l'édition électronique et de la recherche d'informations [46].

Tout comme l'apparition du mot bioinformatique, les débuts de la bioinformatique en tant que discipline à part entière sont difficiles à cerner. Ceci est probablement dû aux frontières relativement floues entre les biomathématiques et la bioinformatique. L'origine de la bioinformatique est parfois vue comme remontant aux premiers travaux

sur la génétique avec, par exemple, les lois sur l'hérédité de Gregor Mendel en 1865 [103] ou la découverte de la structure en double hélice de l'ADN par James Watson et Francis Crick en 1953 [151]. Un autre point de vue place l'origine de la bioinformatique au milieu des années 1960 avec des méthodes de taxonomie numériques en phylogénie moléculaire [141] et la visualisation tridimensionnelle de molécules sur ordinateur [93]. Cette divergence d'opinions sur l'origine de la bioinformatique est due aux multiples définitions du mot bioinformatique. Ceci est d'autant plus vrai que pour les anglo-saxons le mot *bioinformatics* n'a pas véritablement le même sens que le mot *bioinformatique* pour les francophones.

Pour les anglo-saxons, les termes *bioinformatics* et *computational biology* se côtoient pour désigner l'utilisation des technologies de l'information en biologie et en médecine. Notons que le mot *biocomputing* est souvent employé comme synonyme de *computational biology*. Pour certains auteurs, les termes *bioinformatics* et *computational biology* sont interchangeables, tandis que pour d'autres ils ne recouvrent absolument pas les mêmes domaines. Au cours de l'année 2000, le National Institute of Health (NIH), situé près de Washington (USA), en donne les définitions suivantes [112] :

Bioinformatics :

Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.

Computational Biology :

The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems.

Ces définitions du NIH ne font malheureusement pas l'unanimité. Par exemple, toujours aux USA, le National Center for Biotechnology Information (NCBI) donne une autre définition du mot *bioinformatics* [111] :

Bioinformatics is the field of science in which biology, computer science, and information technology merge into a single discipline. The ultimate goal of the field is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be discerned.

There are three important sub-disciplines within bioinformatics:

- *the development of new algorithms and statistics with which to assess relationships among members of large data sets;*
- *the analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures; and*
- *the development and implementation of tools that enable efficient access and management of different types of information.*

Ces différentes définitions montrent un manque de consensus sur les domaines d'applications liés au mot *bioinformatics*. Ceci est vrai au point que le mot *bioinformatics* ne recouvre parfois que l'analyse informatique des séquences d'ADN et de protéines.

Dans le monde francophone, le mot *bioinformatique* fait référence à l'interprétation de l'information biologique plutôt qu'à son traitement sur ordinateur. Aujourd'hui, l'ordinateur tient bien évidemment une grande place dans le domaine de la bioinformatique. Il contribue largement à l'analyse des séquences d'ADN et des structures 3D de molécules, à l'interprétation de l'information biologique, au développement de techniques de prédictions fonctionnelles ainsi qu'à la simulation de processus biologiques. La définition du mot *bioinformatics* du NCBI est assez proche de la définition du mot *bioinformatique* qu'il serait possible de donner en français. Néanmoins, le mot *bioinformatique* ne fait que rarement référence à l'archivage de données biologiques et à leur accès via des bases de données dont les plus connues sont certainement EMBL, GenBank, ou SWISS-PROT.

Ce n'est que vers le milieu des années 1990 que le grand public a eu connaissance de la bioinformatique avec notamment les différents travaux sur le décryptage du génome de différents êtres vivants (la *génomique*) et sur les manipulations génétiques qui pouvaient en découler à travers les travaux liés aux biotechnologies. Citons par exemple, les travaux :

- sur les plantes, et tous les problèmes liés aux incertitudes sur l'effet sur la santé de la consommation d'Organismes Génétiquement Modifiés (OGM),
- sur les animaux, avec notamment l'utilisation de souris génétiquement modifiées pour étudier en laboratoire certaines maladies et, naturellement,
- sur l'homme avec tous les espoirs liés à de nouveaux traitements ainsi qu'avec tous les problèmes d'éthique rencontrés lors de la sélection d'embryons, du stockage d'informations génétiques, et de bien d'autres types de manipulations [6, 73].

Aujourd'hui, les travaux en bioinformatique sur la génomique sont complétés par les travaux sur le *transcriptome* et sur le *protéome*. Le terme de bioinformatique *post-génomique* est alors employé pour qualifier les projets visant à déterminer la partie du génome transcrite en ARN messager (le *transcriptome*) et les projets visant à connaître les protéines exprimées à partir de ces messagers (le *protéome*). A travers ces travaux, c'est la prédiction de la fonction des différents gènes qui est envisagée. La compréhension de la fonction biologique des gènes est devenue le "nouveau Graal" des biologistes [144], cette compréhension devrait permettre de tester virtuellement l'efficacité d'un médicament sur une maladie ou même de mettre au point un traitement adapté à un malade.

La connaissance des gènes et des protéines produites à partir de ces gènes est un point important mais ne suffit pas pour comprendre dans sa totalité le fonctionnement des organismes vivants. En effet, connaître les constituants élémentaires est bien, mais il faut comprendre leurs interactions. La biologie systémique répond à ces besoins en associant l'informatique et la biologie, domaines auxquels viennent bien évidemment s'ajouter les mathématiques et la systémique. La compréhension des systèmes biologiques dans leur ensemble est un projet à long terme sur lequel travaillent un certain nombre de laboratoires de recherche à travers le monde depuis la fin des années 1990 [82, 131]. Aujourd'hui, le développement de cellules et d'organes virtuels est certainement un des plus grands défis de la biologie systémique.

Les principaux centres d'intérêt en biologie systémique consistent, d'après Hiroaki Kitano [82], en la compréhension de la structure de systèmes biologiques, de leur dynamique, des processus de régulation les contrôlant et des méthodes de conception permettant de les construire et de les modifier afin qu'ils aient les propriétés désirées. La modélisation de systèmes biologiques est devenue indispensable; elle permet, en effet, un meilleur dialogue entre biologistes, informaticiens, mathématiciens et, bien sûr, systémiciens en leur fournissant des outils facilitant l'interdisciplinarité [48]. Parmi ces outils, nous pouvons bien évidemment citer CellDesigner [47, 84], SBML [69] ou SB-UML [130]. Les différents objectifs de la biologie systémique laissent à penser que ce nouveau paradigme apportera à la recherche en biologie un nouvel élan aussi fort que l'a été la découverte de la structure en double hélice de l'ADN par Watson et Crick.

Pour résumer, l'utilisation de l'informatique en biologie est aujourd'hui devenue incontournable. Ses champs d'application sont, comme nous venons de le voir, très variés et couvrent la quasi totalité des recherches en biologie. L'informatique joue même un rôle central dans certains travaux. En effet, la simulation informatique permet de compléter la simple preuve *a priori* fournie par une modélisation mathématique en autorisant des validations expérimentales *a posteriori*. De plus, les informaticiens impliqués dans ces recherches se sont aperçus qu'ils pouvaient trouver dans la biologie une source d'inspiration pour résoudre des problèmes liés uniquement à l'informatique. Citons à titre d'exemples, la détection de virus informatiques et les méthodes d'optimisation comme les réseaux de neurones, les algorithmes génétiques ou bien encore les systèmes immunitaires artificiels. Nous pouvons aujourd'hui affirmer que l'informatique et la biologie sont indissociablement liées et probablement pour longtemps car chaque discipline a besoin de l'autre pour avancer.

Dans la suite de ce document, nous détaillerons nos travaux de recherche sur l'utilisation de l'informatique en biologie. Effectués dans le cadre de plusieurs thèses [11, 16, 38, 56, 119, 120], ces travaux concernent essentiellement l'utilisation de systèmes multi-agents dans les domaines de la simulation de systèmes physiologiques humains et du traitement d'images biologiques. Une métaphore biologique utilisée en informatique sera également présentée. Issue du système immunitaire, cette métaphore permet de résoudre un problème purement informatique : l'arrêt d'un système multi-agents spécialisé dans la détection de stries concentriques sur des images d'otolithes de poissons. Ce document s'articule autour de deux parties principales :

1. La première partie (chapitre 2) expose nos travaux menés en collaboration avec plusieurs laboratoires de recherche médicale dans le cadre de thèses de doctorat en informatique ou en biologie. Nous rappelons notre démarche de modélisation de phénomènes biologiques complexes à l'aide de systèmes multi-agents. La complexité de ces phénomènes provient essentiellement de la diversité des composants (cellules, molécules, complexes, ...) et de leurs interactions. Nous présentons également diverses applications dans les domaines de l'immunologie, de l'hématologie et de la cancérologie.

2. Dans la deuxième partie (chapitre 3), nous décrivons nos contributions, tirées de plusieurs thèses, dans le domaine du traitement d'images biologiques. Ces travaux, menés en collaboration avec le centre IFREMER de Brest, visent à déterminer l'âge et la croissance d'animaux aquatiques par analyse d'images de pièces calcifiées. L'acquisition de ce type de données est essentielle pour une gestion rationnelle des stocks halieutiques. L'approche présentée se base sur l'utilisation de systèmes multi-agents permettant de rechercher des informations pertinentes dans les images. Nous utilisons également cette approche pour traiter un autre type d'images biologiques utilisées en protéomique, des images de gels d'électrophorèse 2D. Finalement, nous terminons par un retour sur l'immunologie avec l'utilisation de concepts immunitaires pour auto-réguler une population d'agents chargés de trouver des informations dans des images.

Chapitre 2

Modélisation et simulation de systèmes physiologiques humains

La recherche en biologie utilise depuis fort longtemps l'expérimentation *in vivo* et *in vitro*. Plus récemment, l'expérimentation *in silico* a fait son apparition. Il est ainsi possible de modéliser et de simuler sur ordinateur un certain nombre de phénomènes biologiques. Les calculs obtenus *in silico* peuvent être confrontés aux résultats d'expériences réelles.

Depuis peu, la possibilité d'interagir avec une simulation en cours d'exécution a ouvert la voie à un nouveau type d'expérimentation : l'expérimentation *in virtuo* [145]. Il est désormais possible, à tout moment, de perturber un modèle, de supprimer ou d'ajouter des éléments ou de modifier les conditions aux limites. L'intérêt principal de l'expérimentation *in virtuo* est que toutes ces modifications apportées sur une simulation peuvent être introduites sans arrêter le cours de cette simulation. Nous sommes alors très proches des conditions expérimentales des approches *in vivo* et *in vitro* où le biologiste peut, par exemple, introduire de nouvelles molécules, modifier une concentration, ajouter ou détruire des cellules.

Dans ce chapitre, nous utilisons l'approche multi-agents afin de développer un véritable laboratoire d'expérimentation *in virtuo* en biologie. Nous avons recours à cette approche car les systèmes multi-agents et les systèmes biologiques ont bon nombre de caractéristiques communes. Citons, par exemple, la robustesse, l'émergence, l'auto-organisation et l'adaptabilité. Nous présentons plusieurs niveaux de modélisation multi-agents et nous décrivons plusieurs applications dans les domaines de l'immunologie, de l'hématologie et de la cancérologie. Ces différents travaux sont décrits dans un chapitre de livre [1] ou bien sont majoritairement issus de plusieurs thèses [11, 38, 119].

2.1 La simulation comme outil d'investigation du vivant

Au cours du temps, différents paradigmes se sont développés en biologie. Dans un premier temps le *réductionnisme* proposait de modéliser un processus biologique par

sa structure réelle et invariante, puis le *holisme* issu des théories de l'évolution et de la thermodynamique, modélisait l'évolution et les transformations de ces structures au cours du temps. Ces deux paradigmes ont donné naissance au structuralisme (*i.e.* structure et évolution). Dans les années 1960, des biologistes commencent à s'intéresser à l'approche *systémique* afin d'y trouver des lois générales capables d'appréhender la complexité biologique. La *biologie systémique* se propose en effet d'étudier les processus biologiques comme des systèmes composés d'éléments en interaction [83] plutôt que d'étudier toutes les entités du système indépendamment les unes des autres.

Pour comprendre un processus en tant que système il est nécessaire d'étudier la structure de ce système et de sa dynamique. L'étude de la dynamique conduit tout naturellement à la simulation informatique.

La simulation, c'est l'expérimentation sur un modèle. Un modèle représente une simplification d'un phénomène réel. Seuls certains paramètres pertinents sont donc pris en compte et de nombreux autres sont négligés. La simulation d'un modèle n'a de sens que si son comportement se rapproche du phénomène réel, c'est-à-dire s'il est possible d'approcher la réalité. Ainsi, en faisant varier les paramètres d'un modèle, la simulation permet d'induire ce qui se passerait dans la réalité sous l'influence d'actions analogues.

La simulation possède différents objectifs. Premièrement, lorsque le modèle est validé par l'expérimentation réelle, la simulation permet d'effectuer "en vitesse accélérée" de nombreuses expériences, et ce en contrôlant très précisément chacun des paramètres. Deuxièmement, si le modèle est incomplet ou insuffisant, la simulation permet de tester des hypothèses pour en extraire celles se rapprochant le plus de la réalité. Dans ce cas, elle participe à la mise au point du modèle biologique. La simulation peut s'effectuer sur une maquette ou bien, depuis l'avènement de l'informatique, sur un ordinateur grâce à un modèle numérique. De nombreux domaines de la recherche et de l'industrie font appel à la simulation, soit pour le développement de modèles, soit pour la mise au point de produits ou la prévision de phénomènes complexes. L'aéronautique, l'automobile, l'économie, la chimie, la météorologie, l'astrophysique, la cosmologie, la physique nucléaire ou encore la biologie en font partie.

L'expérimentation *in vitro* constitue pour la biologie la forme la plus répandue de simulation. Des cellules sont prélevées chez une personne, puis sont étudiées à l'extérieur du corps humain. L'immunologie comme l'hématologie en sont de grandes consommatrices.

C'est en 1966 que le premier modèle mathématique décrivant un phénomène immunitaire voit le jour [61]. Cette description formelle, à base d'équations différentielles, est suffisamment simple pour être manipulée à l'aide de l'outil mathématique. Ainsi, par la résolution d'équations, le modèle peut être exploré et des solutions à différents problèmes peuvent être trouvées. Il s'agissait de la première alternative à l'expérimentation *in vitro*.

Un peu plus tard, vers 1970, les modèles immunitaires à base d'équations différentielles se sont complexifiés. Par exemple, c'est en 1974 que Niels Jerne [76] décrit son modèle du réseau idiotypique. Plus précisément, sa théorie explique comment une réponse immunitaire en initie une autre dirigée non plus contre l'antigène, mais contre les anticorps de la précédente. Son modèle est hautement non linéaire et aucune solution analytique simple ne peut être trouvée. Le recours à la simulation numérique devient alors indispensable.

Au cours des années 1980, les premiers simulateurs de réactions biochimiques apparaissent. Citons, par exemple, le simulateur KinSim [15] qui permet, grâce à la génération de courbes obtenues par résolution d'équations différentielles, d'analyser la cinétique de réactions enzymatiques. L'utilisateur peut interactivement modifier des paramètres cinétiques afin d'ajuster les courbes avec les résultats expérimentaux. Citons également des travaux plus récents comme Gepasi [104, 105] (approche déterministe) ou StochSim [91, 108] (approche stochastique) qui permettent, toujours à l'aide d'équations différentielles, de simuler et de modéliser des réactions biochimiques au niveau moléculaire. D'autres types de simulateurs existent et sont plutôt orientés vers la modélisation de processus cellulaires. A titre d'exemples, mentionnons E-Cell [142, 147], Virtual Cell [134] et BioDrive [90] qui s'attachent à modéliser l'intérieur d'une cellule. Dans ces trois simulateurs, les mécanismes de base sont décrits par des réactions enzymatiques ou des réactions de formation de complexes modélisés par des équations différentielles.

Depuis peu, un langage de description de données biologiques appelé *System Biology Markup Language* (SBML) est apparu [69]. Fondé sur XML, ce langage permet la standardisation et l'échange de données biologiques entre chercheurs. SBML permet de décrire essentiellement des réseaux biochimiques mais aussi des modèles quantitatifs d'interaction entre ces réseaux et d'autres phénomènes. SBML permet également de décrire des événements discrets qui sont déclenchés par des changements d'état du système modélisé. Notons que parmi les personnes impliquées dans le consortium SBML, nous trouvons les concepteurs des principaux simulateurs actuels, simulateurs bien évidemment compatibles avec SBML.

En parallèle, l'informatique évolue, tant du point de vue de la puissance de calcul que des langages de développement. Différents paradigmes apparaissent comme ceux de la vie artificielle, des automates cellulaires, de la programmation par objet ou des systèmes multi-agents. Il devient alors possible de modéliser et de simuler des phénomènes biologiques autrement que par des équations différentielles, c'est-à-dire en ne pensant plus seulement aux influences entre populations de cellules.

Nous le voyons, l'informatique est un outil très intéressant pour la biologie. Nous sommes convaincus, et ce chapitre tente de le démontrer, que la recherche en biologie ainsi que la mise au point de médicaments peuvent se faire en partie à l'aide de l'ordinateur. Dans la suite de ce chapitre, nous présentons notre démarche de modélisation multi-agents de phénomènes biologiques ainsi que plusieurs applications dans les domaines de l'immunologie, de l'hématologie et de la cancérologie.

2.2 Notre démarche de modélisation

Afin de bien comprendre notre démarche de modélisation multi-agents, nous devons tout d’abord étudier les différentes caractéristiques des systèmes biologiques qui, selon Hiroaki Kitano [83], sont la robustesse, l’émergence, l’auto-organisation et l’adaptabilité.

2.2.1 Caractéristiques des systèmes biologiques

La robustesse est une des caractéristiques essentielles des systèmes biologiques. Ainsi, de nombreux auteurs ont montré que les systèmes biologiques sont capables de conserver les propriétés fonctionnelles du système malgré différentes perturbations et ceci peut être réalisé par quatre stratégies différentes. Premièrement, la mise en place de schémas de contrôle (par exemple la rétroaction) limite l’effet de la modification de paramètres cinétiques de voies de transmission [17]. Deuxièmement, la redondance des gènes codant pour des fonctions biologiques importantes permet de conserver ces fonctions en cas de mutation d’un exemplaire de ce gène. *A contrario*, la non redondance rend certaines cellules hypersensibles aux mutations. Par exemple, la mutation sur un gène pléiotropique perturbe de nombreux processus, ce qui permet dans un système multi-cellulaire d’éliminer rapidement les cellules touchées [88]. Enfin, une architecture modulaire telle que la cellule évite l’effondrement du système si l’un de ses modules est touché.

L’émergence est également un point important des systèmes biologiques [129]. Nous pouvons caractériser l’émergence par l’apparition d’une propriété dans un système alors qu’aucun composant de ce système ne contient ni n’exprime cette propriété. Ceci a été observé lors de la modélisation de différentes voies de transduction en interaction [17]. L’expression “le tout est plus que la somme des parties” semble donc tout à fait adaptée pour décrire les systèmes biologiques.

La notion d’auto-organisation émergente est une conséquence directe de la notion d’émergence; ainsi l’auto-organisation ne serait que la résultante de l’ensemble des interactions existantes entre les entités d’un système. L’auto-organisation émergente est observée par exemple dans le cas de formation de complexes tels que les complexes immuns. L’auto-organisation peut être téléologique, c’est-à-dire que l’organisation est préprogrammée dans le système. L’embryologie est un exemple marquant d’auto-organisation dans la matière vivante [7].

Deux niveaux d’adaptation peuvent être définis. Premièrement, l’adaptation au niveau du système peut être associée à l’auto-organisation. L’environnement agissant sur l’organisation d’un système, tout changement de l’environnement peut modifier l’organisation de ce système. Par exemple, Amund Ringvold [123], a montré qu’il existe une variation saisonnière de la cornée. Il a en effet observé que le nombre de cellules

de la cornée varie en fonction du rayonnement ambiant. Deuxièmement, l'adaptation existe au niveau de l'entité. Ceci a été observé par Rosamiria Greco-Perotto [55] qui a montré une modification des cellules du tissu adipeux de rats modifiés pour résister au froid.

Aujourd'hui, la simulation sur ordinateur est devenue incontournable pour étudier la dynamique de systèmes biologiques. Dans ce chapitre, nous décrivons l'utilisation de systèmes multi-agents pour modéliser et simuler ce type de systèmes.

2.2.2 Caractéristiques des systèmes multi-agents

Les premiers travaux sur les systèmes multi-agents datent de la fin des années 1970 [64]. Ils sont motivés par la constatation suivante : "Il existe dans la nature des systèmes capables d'accomplir des tâches collectives complexes dans des environnements dynamiques, sans contrôle externe ni coordination centrale" [145]. Les colonies d'insectes sont certainement les exemples les plus notoires d'organisations collectives [21]. Les notions de robustesse, d'émergence, d'auto-organisation et d'adaptabilité sont donc sous-jacentes dans les systèmes multi-agents et font même partie de leurs fondements. Ceci fait donc de la méthodologie multi-agents un bon candidat pour la modélisation et la simulation de phénomènes biologiques.

La robustesse a été décrite comme inhérente aux systèmes multi-agents [2]. Elle est alors associée à la redondance des agents et à leur autonomie. Cependant certains auteurs estiment que ces phénomènes ne suffisent pas à résoudre tous les problèmes [135], notamment dans les systèmes utilisant des agents-cognitifs ou des agents avec un comportement complexe. Ils insistent donc sur le fait que les modèles basés agents nécessitent des structures particulières pour améliorer la robustesse [117]. D'après le collectif "M.R. Jean" [75], c'est la dynamique des interactions entre chaque entité et son environnement qui est la condition de base pour l'émergence d'une structure ou d'une fonctionnalité globale. Ainsi, chaque agent possède un environnement local sur lequel il peut agir et qu'il partage avec d'autres agents. La dynamique du système procède d'une itération entre interprétation de leur environnement local par les agents, action des agents sur cet environnement, nouvelle interprétation de cet environnement modifié, nouvelle action et ainsi de suite. Quand une telle dynamique se stabilise on peut parler d'émergence de structure ou de fonctionnalité globale. L'auto-organisation est liée à la précédente notion. En effet, l'auto-organisation est l'apparition de structures émergentes [100]. Les agents vivent dans un milieu ouvert, dynamique. Ceci implique que les agents soient capables de s'adapter à leur environnement malgré des changements de ce dernier [117]. L'adaptabilité d'un agent peut être associée à sa capacité d'apprentissage afin qu'il soit capable d'atteindre ses objectifs malgré des changements de l'environnement. On parle alors d'agent adaptatif [2, 99]. L'imprévu étant inhérent à la vie de ces systèmes, l'auto-organisation, qui correspond à un changement décidé de manière autonome, devient un moyen pour parvenir à surmonter les perturbations éventuelles de l'environnement [24, 101]. Par le biais de l'auto-organisation, un système

doit conserver son adéquation fonctionnelle malgré les variations de son environnement.

Pour résumer, la méthodologie multi-agents s'organise autour de différents concepts. Yves Demazeau définit un système multi-agents comme étant composé de quatre concepts clefs que sont les Agents, l'Environnement, les Interactions et l'Organisation. C'est l'approche *Vowels* ou AEIO [34].

Un agent (A) est une entité qui est plongée dans un environnement. Il interagit avec cet environnement et est doté d'une certaine autonomie. Il n'y a donc pas de contrôle global du système. L'agent est l'élément de base d'un système multi-agents. Il peut être réactif [42] ou cognitif [154] selon son degré d'intelligence. Qu'il soit réactif ou cognitif, un agent évolue toujours selon un cycle à trois temps :

- perception : ses capteurs lui fournissent une vision de l'environnement local,
- décision : suivant ses intentions, son état interne et sa perception, l'agent choisit une action à effectuer,
- action : il modifie l'environnement par son action.

L'environnement (E) peut être divisé en deux parties, l'environnement du système multi-agents et l'environnement de l'agent. Le premier est le support des actions des agents. Le second contient à la fois l'environnement du système multi-agents et les autres agents de l'environnement. Par extension, l'environnement est l'ensemble des conditions extérieures susceptibles d'agir sur le fonctionnement d'un agent.

La notion d'interaction (I) correspond à la mise en relation de deux ou plusieurs agents à travers une série d'actions. Ces actions peuvent être directes ou indirectes, c'est-à-dire que l'interaction peut se produire soit directement entre deux agents, soit par l'intermédiaire d'un autre agent ou de l'environnement. C'est cette dynamique des interactions entre composants qui permet l'émergence de nouvelles fonctionnalités plus complexes.

L'organisation (O) est l'agencement qui se produit entre les agents pour former un système. Les agents vont créer un réseau d'interactions qui va former un tout qui sera capable de résister – dans une certaine mesure – à des perturbations. L'interaction est donc le composant de base de toute organisation, organisation nécessairement dynamique.

Le modèle *Vowels* peut être enrichi de la dernière voyelle, la lettre U comme Utilisateur [145]. En effet, celui-ci doit être capable d'interagir avec le modèle en cours de simulation. Ceci ouvre la voie à l'expérimentation *in virtuo*. Cette approche complète les approches classiques : *in vivo*, *in vitro* et *in silico*. Dépassant la simple observation du modèle numérique en cours d'utilisation sur un ordinateur, l'utilisateur peut tester la réactivité et l'adaptabilité du modèle en cours de fonctionnement.

Comme nous venons de le voir, la méthodologie multi-agents semble tout à fait adaptée à la modélisation et à la simulation de phénomènes biologiques pour l'expérimentation *in virtuo*. Décrivons brièvement maintenant les différents niveaux de modélisation que nous verrons beaucoup plus en détail par la suite.

2.2.3 De l'agent-cellule aux phénomènes

Les premiers travaux *in virtuo* ont utilisé l'analogie immédiate entre la cellule biologique et l'agent informatique – agent de type réactif – pour modéliser le rôle de récepteurs CD5 de certains lymphocytes B vis-à-vis de l'apoptose [11]. La première étape dans la conception d'un agent-cellule consiste à choisir la forme géométrique qui constituera la membrane cellulaire. Il faut ensuite placer sur cette membrane des récepteurs qui permettront à l'agent de percevoir l'environnement. Enfin, il est nécessaire de définir le comportement de cet agent-cellule. Ce comportement peut être régi de différentes manières. Citons, par exemple, les algorithmes “classiques”, les équations différentielles, les outils de l'intelligence artificielle (réseaux de neurones, cartes cognitives, ...) ou encore les diagrammes d'états-transitions.

Lorsque le nombre d'agents-cellules devient trop important, il faut trouver un moyen pour réduire les temps de calcul. Nous avons alors proposé la notion d'agent-réaction.

Les modèles d'expériences *in vitro* sont souvent constitués de réactions chimiques entre des espèces moléculaires en solution dans un milieu (réaction enzymatique, oxydo-réduction, dimérisation...). Dans ce genre de cas, les concentrations des différentes molécules peuvent être considérées comme étant homogènes. Dès lors, il est possible de modéliser les populations de molécules de même type plutôt que de modéliser chaque molécule individuellement. Il est alors possible de réifier les interactions, réactions chimiques, entre ces populations. Le verbe réifier signifie ici que l'on considère les interactions comme des agents à part entière. Un agent-réaction représente donc une réaction chimique entre plusieurs espèces dans un milieu homogène. Un agent-réaction effectue son cycle en trois temps : lecture des concentrations des espèces dans le milieu (perception), calcul de la quantité de réactifs (décision), mise à jour des concentrations (action). Un ensemble d'agents-réactions peut former un réseau biochimique complexe et permet de le simuler très précisément. Ainsi, par exemple, un agent-cellule peut contenir plusieurs agents-réactions pour décrire ses voies de transduction.

Lorsque l'on désire modéliser un phénomène complexe (cellule, organe, ...), il peut être nécessaire de regrouper les différentes entités qui composent un mécanisme particulier. Nous utilisons pour cela des agents-phénomènes qui représentent des organisations au sens systémique. Nous pouvons, par exemple, regrouper différentes réactions chimiques qui constituent une voie de transduction.

Détaillons maintenant ces différents niveaux de modélisation permettant la simulation de phénomènes biologiques pour l'expérimentation *in virtuo*. Notons que ces travaux ont été développés pour la majorité d'entre eux avec le langage orienté agent oRis (voir la section 3.2 du chapitre 3). Ce langage est associé à un simulateur qui assure l'équité entre les agents. De plus, grâce aux propriétés dynamiques de ce langage, il est possible d'intervenir sur le modèle en cours de simulation. Cette caractéristique répond tout à fait aux besoins de l'expérimentation *in virtuo*.

2.3 Agent-cellule

Dans notre modélisation multi-agents de systèmes biologiques, il semblait assez naturel dans un premier temps d'associer un agent à une cellule. Par similarité avec une cellule, un agent est doté d'un ensemble de récepteurs à sa surface. Un agent a également un comportement interne éventuellement complexe et multi-modèles. Les agents sont soumis aux règles de l'environnement. Ces règles consistent à soumettre les agents, via leurs récepteurs, à l'influence des autres agents. Ces influences sont la cause de leurs mouvements. Ce déplacement est à la fois linéaire et angulaire. Lors des déplacements, des liaisons entre agents peuvent survenir. En fonction des stimuli reçus grâce à leurs récepteurs, les agents modifient leurs comportements et leurs états internes. Quand un agent-cellule est activé, il peut augmenter son nombre de récepteurs et inversement lorsqu'il est désactivé, un certain nombre de ses récepteurs sont éliminés par endocytose.

Les stimuli reçus grâce aux récepteurs correspondent, soit à la liaison avec les récepteurs d'autres agents-cellules soit, à la liaison avec des molécules. Lors de la modélisation d'une population de molécules, il est actuellement impossible, pour des raisons de temps de simulation, d'associer une seule molécule à un agent. Afin de résoudre ce problème, un agent-molécule représentera dans notre modélisation un ensemble plus ou moins grand de molécules. De même, un agent-cellule sera le représentant d'un certain nombre de cellules. Il sera ainsi possible de passer de grandeurs quantitatives de simulation à des grandeurs quantitatives plus proches de la réalité.

Les liaisons entre agents peuvent aboutir à la formation de complexes. Un complexe, est lui-même un agent composé d'autres agents. Il est doté d'une masse et d'un centre de gravité. La masse du complexe est la somme des masses des agents qui le composent. Au sein du complexe, chaque agent conserve son indépendance comportementale mais est soumis aux lois physiques imposées par le collectif. Cela signifie donc qu'un agent composé est une seule et même entité au vu des lois physiques de l'environnement. Ces lois physiques imposées aux agents (cellules ou complexes) seront expliquées plus loin mais, commençons par la description précise de la construction d'un agent-cellule.

2.3.1 Construction d'un agent-cellule

La construction d'un agent-cellule débute par le choix d'une forme représentant la membrane cellulaire. Celles retenues, les plus simples et les plus naturelles, sont le cercle en 2D et la sphère en 3D. D'autres formes sont également possibles mais plus complexes à gérer comme, par exemple, les ellipses ou les ovoïdes. Certains travaux font également état de membranes flexibles par déformation du squelette de la cellule [118]. Ensuite, des récepteurs choisis dans une bibliothèque sont placés autour de la forme-

membrane (voir la figure 2.1). Ils peuvent être uniformément répartis à la surface de la cellule ou être dans des positions déterminées par leur position angulaire et leur distance par rapport au centre de l'agent. Cette disposition est importante dans la mesure où, en réalité, deux récepteurs proches à la surface d'une cellule peuvent interagir. Ainsi, les phénomènes d'auto-interaction entre récepteurs sont modélisables.

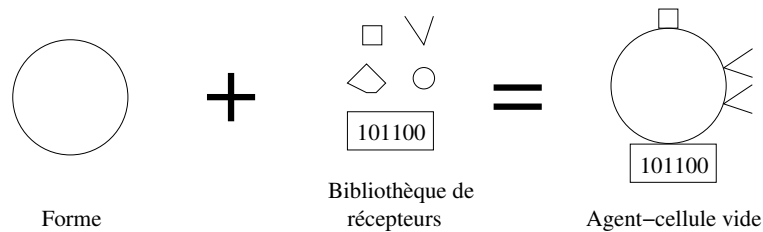


Figure 2.1: Création d'un agent-cellule vide.

La troisième étape de la construction consiste à inclure les comportements internes. Ces comportements sont décrits par des algorithmes, des équations différentielles, des fonctions mathématiques, des systèmes multi-agents ou tout autre formalisme permettant de décrire des comportements (voir la figure 2.2).

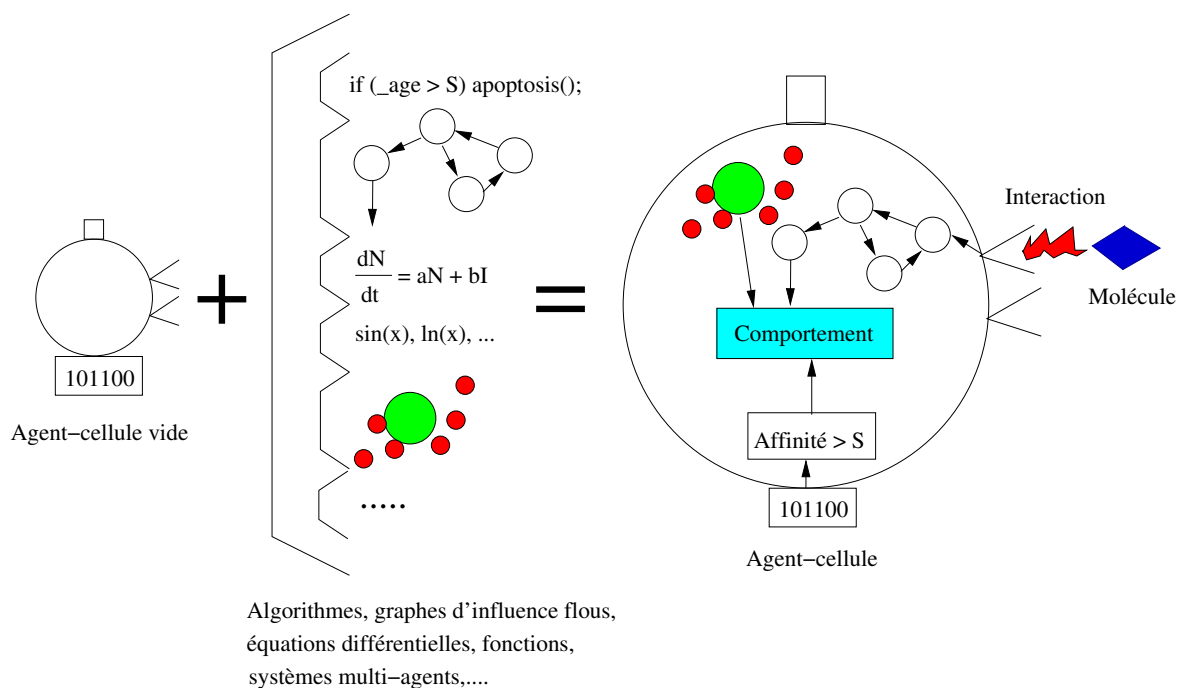


Figure 2.2: Création d'un agent-cellule complet.

2.3.2 Comportement d'un agent-cellule

Le choix entre algorithmes, graphes d'influence flous, équations différentielles, fonctions, systèmes multi-agents ou d'autres types de formalisme dépend de la nature de chaque problème au sein de la cellule, des connaissances acquises sur le problème ainsi que de l'habitude du modélisateur à manipuler l'un des outils plutôt qu'un autre.

Concernant les algorithmes, le modèle inclut un certain nombre de comportements prédéfinis. Par exemple, un agent a la possibilité de produire des récepteurs à sa surface (pour devenir sensible à un type de molécule), d'internaliser ses récepteurs lors de liaisons avec des molécules, de se diviser (mitose), de créer de nouveaux agents-molécule (agent-interleukine, interféron,...) ou de se détruire (apoptose). D'autres comportements spécifiques sont intégrables afin d'améliorer et d'affiner le modèle (voir la figure 2.3).

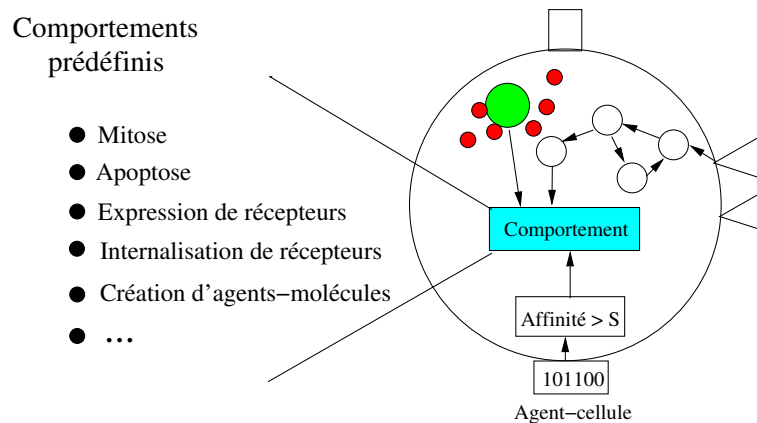


Figure 2.3: Comportements de base prédéfinis.

Rappelons que pendant la simulation, des agents peuvent se regrouper, formant alors un agent composé. Un tel agent est considéré par l'environnement comme une seule et même entité pour ses déplacements, mais reste une mosaïque de comportements individuels. Un agent composé peut se fractionner ou fusionner avec un autre agent composé (voir la figure 2.4). Cela survient lors de création ou de destruction de liaisons.

2.3.3 Interactions entre agents-cellules, agents-molécules

Les interactions entre agents se font donc à travers les récepteurs des agents. Les liaisons entre récepteurs ne sont pas définitives, elles peuvent se défaire dans trois cas. Premièrement, lorsqu'une des cellules d'un complexe meurt, les différents récepteurs sont libérés et sont donc à nouveau capables de se lier à d'autres récepteurs. Le second

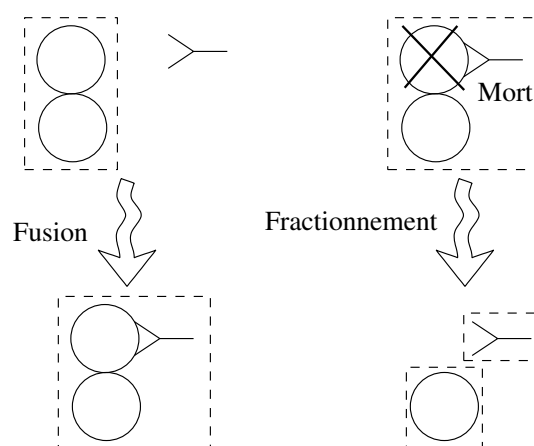


Figure 2.4: Fusion et division d'agents composés.

cas survient lorsque la durée de la liaison atteint une valeur déterminée. En effet, les liaisons moléculaires sont temporaires : leur durée est mesurée en demi-durée de vie. C'est-à-dire que pour un ensemble de liaisons de même type, au bout de la demi-durée de vie, la moitié des liaisons intermoléculaires est détruite. Le dernier cas correspond à la liaison entre un agent-cellule et un agent-molécule. L'agent-cellule internalise alors l'entité représentant la molécule. L'internalisation élimine la molécule au sein de la cellule après que celle-ci ait entraîné un signal.

Les forces d'attraction entre les agents prennent en compte la taille et la masse des objets (cellules, molécules) ainsi que l'affinité entre les récepteurs [113]. Notons que dans le cas de complexes, les forces totales d'attraction doivent être calculées. Ainsi, à partir de ces forces d'attraction, les agents se déplacent alors en translation et en rotation (voir la figure 2.5).

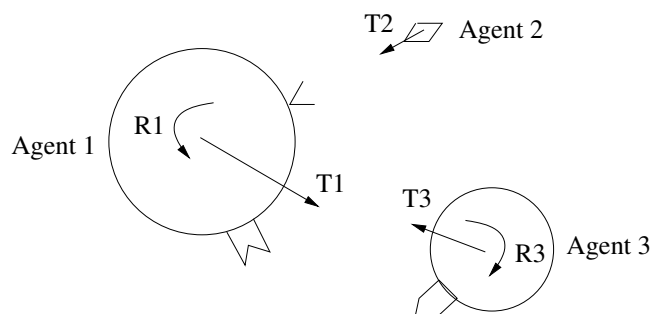


Figure 2.5: Déplacement des agents : translation, rotation.

2.3.4 Exemple de comportement interne d'un agent-cellule

Nous détaillons maintenant un exemple de modélisation de comportement interne d'une cellule à base de Graphe d'Influence Flou (GIF). Cette modélisation sera utilisée plus loin lors de la description d'une application en cancérologie où les différentes voies de transduction d'une cellule seront spécifiées à l'aide d'un GIF (voir la section 2.6.3).

Une solution pour modéliser le comportement d'une cellule dans toute sa complexité consiste à modéliser les réseaux intracellulaires qui la contrôlent [8]. La méthodologie utilisée doit tout d'abord permettre une représentation sémantique facilitant le dialogue avec les biologistes. Elle doit également permettre de représenter des boucles de rétroaction présentes dans les systèmes biologiques. Enfin, cette méthodologie doit se suffire de données incomplètes, rendant difficiles des simulations quantitatives.

Des travaux ont été menés pour modéliser des réseaux intracellulaires avec différentes méthodes, soit dans le cas d'une seule cellule en utilisant des réseaux de neurones [150] ou des réseaux bayésiens [133], soit avec un ensemble de cellules en utilisant des algorithmes "simples" [89] ou l'architecture *Belief, Desires, Intentions* (BDI) [80]. La plupart de ces travaux ne permettent pas facilement de simuler des réseaux intracellulaires complexes en respectant les caractéristiques énoncées au paragraphe précédent.

Dans la suite de cette section, nous présentons donc une méthode de modélisation de réseaux intracellulaires à base de Graphe d'Influence Flou. Cette méthode de modélisation est très proche des concepts manipulés par les biologistes (voies de transduction) et permet de concevoir des réseaux avec rétroaction ne nécessitant que peu, voire pas, de données biologiques précises.

2.3.4.1 Graphe d'Influence Flou

Les Graphes d'Influence Flous (GIF) appelés également Cartes Cognitives Floues (CCF) sont issus de travaux de psychologues. Les cartes cognitives ont été introduites afin de décrire des comportements complexes de mémorisation topologiques chez les rats [146]. Dans les années 1970, Robert Axelrod formalisa ces cartes cognitives sous forme de graphes orientés et les utilisa en théorie de la décision appliquée au domaine politico-économique [10]. En 1986, Bart Kosko intégra la logique floue aux graphes d'Axelrod qui devinrent alors les Cartes Cognitives Floues [87]. En 2001, dans le cadre de la réalité virtuelle, Marc Parenthoën propose de délocaliser les cartes cognitives floues au sein de chaque agent autonome afin de modéliser leur comportement perceptif [114]. Le lecteur intéressé peut trouver une vue d'ensemble de cet outil de formalisation dans [3].

Les Cartes Cognitives Floues sont essentiellement des graphes d'influence orientés. Les nœuds sont des concepts (C_i) regroupés dans un ensemble $\mathcal{C} = \{C_1, \dots, C_n\}$. Les arcs (C_i, C_j) sont orientés et représentent des liens de causalité entre les concepts. Un degré d'activation (a_i) est associé à chaque concept, alors que le poids L_{ij} d'un arc traduit le degré d'excitation ou d'inhibition du concept C_i vers le concepts C_j . Un exemple de Carte Cognitive Floue est donné figure 2.6. La dynamique de la carte est calculée par produit matriciel normalisé comme indiqué par la définition formelle des Cartes Cognitives Floues donnée figure 2.7.

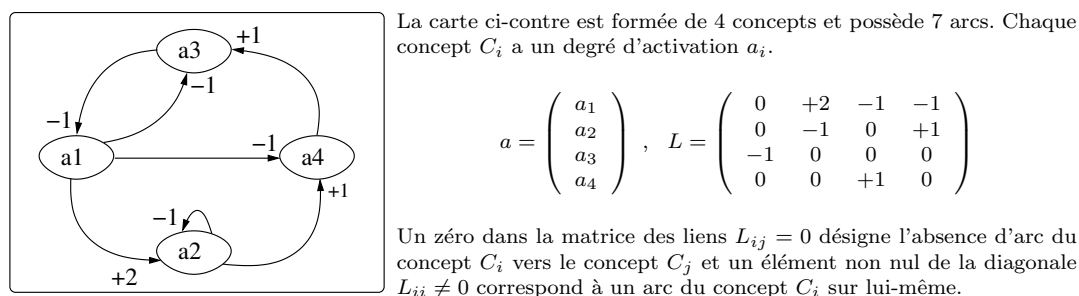


Figure 2.6: Exemple de carte cognitive.

2.3.4.2 Eléments de validation : la voie des MAPK

En reprenant l'idée de Parenthoën [114], nous modélisons le comportement d'un agent-cellule à l'aide d'une Carte Cognitive Floue que nous appellerons dorénavant Graphe d'Influence Flou (GIF). Un agent-cellule possède donc un GIF ayant des concepts sensitifs (*i.e.* concepts d'entrée) dont les degrés d'activation sont obtenus par *fuzzification* des données issues de ses récepteurs de surface (ou capteurs). Inversement, les actions de l'agent-cellule sont obtenues par *défuzzification* des degrés d'activation de concepts "moteurs" (*i.e.* concepts de sortie) du GIF (voir la figure 2.8).

Afin de valider notre approche, nous avons choisi de modéliser la voie des "Mitogen Activated Proteine Kinases" (MAPK) du fait de leur rôle dans le cycle et la croissance cellulaire. La voie des MAPK est composée de trois kinases conservées au cours de l'évolution de nombreuses espèces : Raf (MAPKKK), MEK (MAPKK) et ERK (MAPK). Un modèle de référence (voir la figure 2.9-a) a été proposé par Chi-Ying Huang et James Ferrel [68]. Ce modèle décrit l'activation de la voie des MAPK dans l'ovocyte de xénope. L'entrée de ce modèle est un enzyme théorique E1 qui est la cible de la rétro-inhibition (voir la figure 2.9-b) proposée ensuite par Boris Kholodenko [81]. Cette dernière hypothèse peut provoquer l'apparition d'oscillations de la concentration de MAPK activée (MAPK-PP).

La figure 2.10 présente les GIF associés aux modèles de Huang-Ferrel (a) et de Kholodenko (b). On peut observer que le poids des arcs est de +1 pour l'activation et de -1 pour l'inhibition. Chaque concept représente la forme activée de la molécule

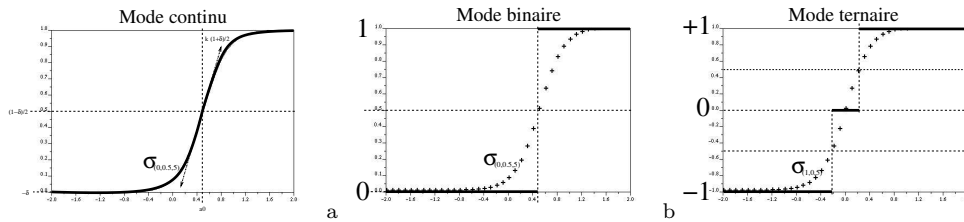
On désigne par K l'un des anneaux \mathbb{Z} ou \mathbb{R} , par δ l'un des nombres 0 ou 1, par \mathcal{V} l'un des ensembles $\{0, 1\}$, $\{-1, 0, 1\}$, ou $[-\delta, 1]$. Soient $(n, t_0) \in \mathbb{N}^2$ et $k \in \mathbb{R}_+^*$.

Une **carte cognitive floue** \mathcal{F} est un sextuplet $(\mathcal{C}, \mathcal{A}, L, A, f_a, \mathcal{R})$ où :

1. $\mathcal{C} = \{C_1, \dots, C_n\}$ est l'ensemble des n concepts formant les noeuds d'un graphe.
2. $\mathcal{A} \subset \mathcal{C} \times \mathcal{C}$ est l'ensemble des arcs (C_i, C_j) orientés de C_i vers C_j .
3. $L : \begin{cases} \mathcal{C} \times \mathcal{C} & \rightarrow K \\ (C_i, C_j) & \mapsto L_{ij} \end{cases}$ est une fonction de $\mathcal{C} \times \mathcal{C}$ vers K associant L_{ij} à un couple de concepts (C_i, C_j) , avec $L_{ij} = 0$ si $(C_i, C_j) \notin \mathcal{A}$, ou avec L_{ij} égale le poids de l'arc orienté de C_i vers C_j si $(C_i, C_j) \in \mathcal{A}$. $L(\mathcal{C} \times \mathcal{C}) = (L_{ij}) \in K^{n \times n}$ est une matrice de $\mathcal{M}_n(K)$. C'est la matrice des liens de la carte \mathcal{F} que, pour simplifier, on note L sauf indication contraire.
4. $A : \begin{cases} \mathcal{C} & \rightarrow \mathcal{V}^N \\ C_i & \mapsto a_i \end{cases}$ est une fonction qui à chaque concept C_i associe la suite de ses degrés d'activation telle que pour $t \in \mathbb{N}$, $a_i(t) \in \mathcal{V}$ soit son degré d'activation à l'instant t . On notera $a(t) = [(a_i(t))_{i \in \llbracket 1, n \rrbracket}]^T$ le vecteur des activations à l'instant t .
5. $f_a \in (\mathbb{R}^n)^N$ une suite de vecteurs d'activations forcées tels que pour $i \in \llbracket 1, n \rrbracket$ et $t \geq t_0$, $f_{a_i}(t)$ soit l'activation forcée du concept C_i à l'instant t .
6. \mathcal{R} est une relation de récurrence sur $t \geq t_0$ entre $a_i(t+1)$, $a_i(t)$ et $f_{a_i}(t)$ pour $i \in \llbracket 1, n \rrbracket$ traduisant la dynamique de la carte \mathcal{F} .

$$\forall i \in \llbracket 1, n \rrbracket, a_i(t_0) = 0; \forall i \in \llbracket 1, n \rrbracket, \forall t \geq t_0, a_i(t+1) = \sigma \circ g \left(f_{a_i}(t), \sum_{j \in \llbracket 1, n \rrbracket} L_{ji} a_j(t) \right)$$

où $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ est une fonction de \mathbb{R}^2 vers \mathbb{R} , par exemple: $g(x, y) = \min(x, y)$ ou $\max(x, y)$ ou $\alpha x + \beta y$, et où $\sigma : \mathbb{R} \rightarrow \mathcal{V}$ est une fonction de \mathbb{R} vers l'ensemble des degrés d'activation \mathcal{V} normalisant les activations comme suit :



- (a) En mode continu, $\mathcal{V} = [-\delta, 1]$, σ est la fonction sigmoïde $\sigma_{(\delta, a_0, k)}$ centrée en $(a_0, \frac{1-\delta}{2})$, de pente $k \cdot \frac{1+\delta}{2}$ en a_0 et de limites en $\pm\infty$ respectivement 1 et $-\delta$:

$$\sigma_{(\delta, a_0, k)} : \begin{cases} \mathbb{R} & \rightarrow [-\delta; 1] \\ a & \mapsto \frac{1 + \delta}{1 + e^{-k(a-a_0)}} - \delta \end{cases}$$

- (b) En mode binaire, $\mathcal{V} = \{0, 1\}$, $\sigma : a \mapsto \begin{cases} 0 & \text{si } \sigma_{(0, 0.5, k)}(a) \leq 0.5 \\ 1 & \text{si } \sigma_{(0, 0.5, k)}(a) > 0.5 \end{cases}$.

- (c) En mode ternaire, $\mathcal{V} = \{-1, 0, 1\}$, $\sigma : a \mapsto \begin{cases} -1 & \text{si } \sigma_{(1, 0, k)}(a) \leq -0.5 \\ 0 & \text{si } -0.5 < \sigma_{(1, 0, k)}(a) \leq 0.5 \\ 1 & \text{si } \sigma_{(1, 0, k)}(a) > 0.5 \end{cases}$.

Figure 2.7: Définition d'une carte cognitive floue [145].

(e.g. le nœud MAPK représente la forme activée de MAPK : MAPK-PP). La donnée d'entrée est la concentration de E1 donnée par l'agent-cellule. Cette concentration doit être *fuzzyfiée* avant d'être utilisée par le graphe d'influence. Les données de sortie MAPKK (concentration de MAPKK-PP) et MAPK (concentration de MAPK-PP) sont *dé-fuzzyfiées* avant d'être utilisables par l'agent-cellule.

Pour chacun des deux modèles, nous avons effectué deux simulations. La première

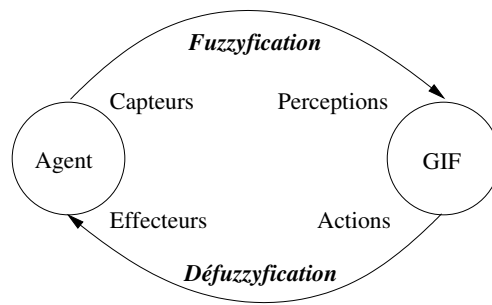


Figure 2.8: Association d'un agent à un Graphe d'Influence Flou.

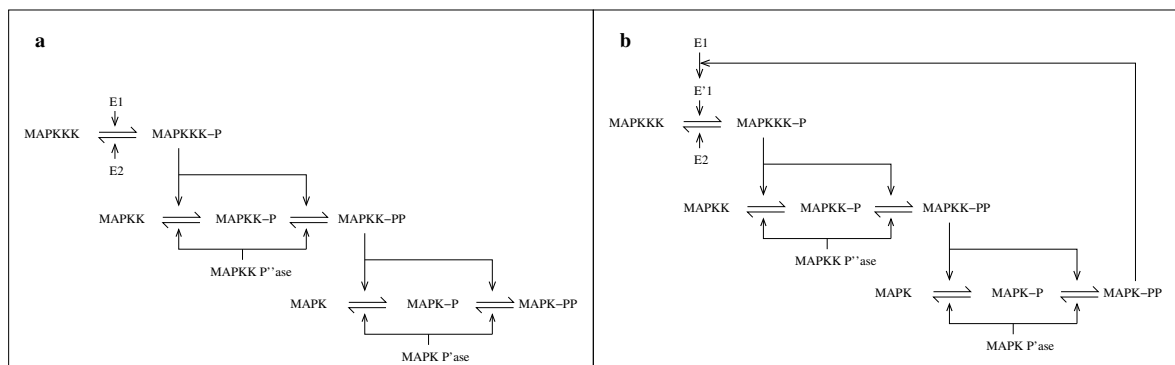


Figure 2.9: a: Modèle de Huang et Ferell ; b: Modèle de Kholodenko.

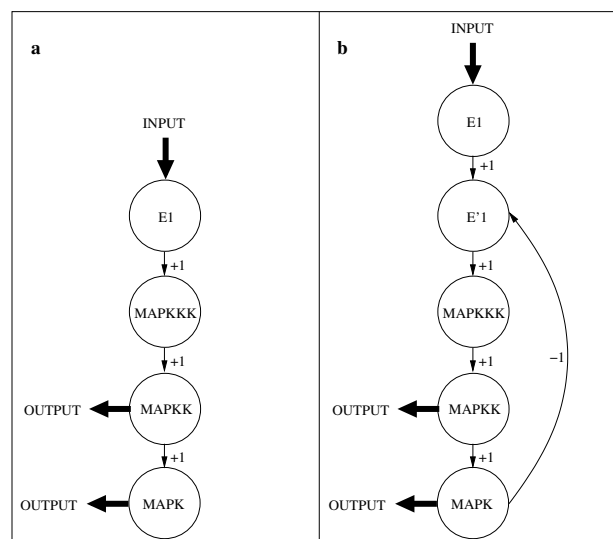


Figure 2.10: a: GIF du modèle de Huang et Ferell ; b: GIF du modèle de Kholodenko.

avec 1 agent-cellule et la seconde avec 1000 agents-cellules. Pour évaluer notre approche, nous avons utilisé les courbes obtenues par Nils Blüthgen [18] grâce à des équations différentielles ordinaires (voir la figure 2.11). Nous avons pour cela tracé l'évolution des concentrations de MAPK, MAPKK et MAPKKK (voir la figure 2.12). Pour le modèle de Kholodenko, plus le nombre d'agents-cellules est grand, plus les courbes obtenues

sont périodiques. Un agent-cellule pris individuellement n'a peut-être pas tout à fait le comportement désiré, par contre, le comportement d'un ensemble d'agents-cellules représente davantage la réalité. Et donc, l'individuel, c'est bien, le collectif, c'est mieux.

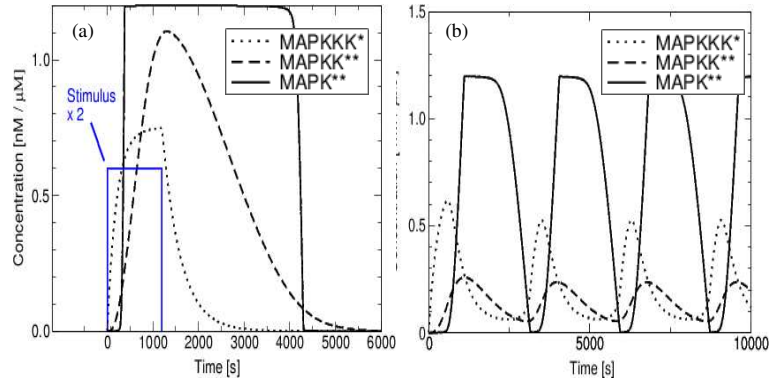
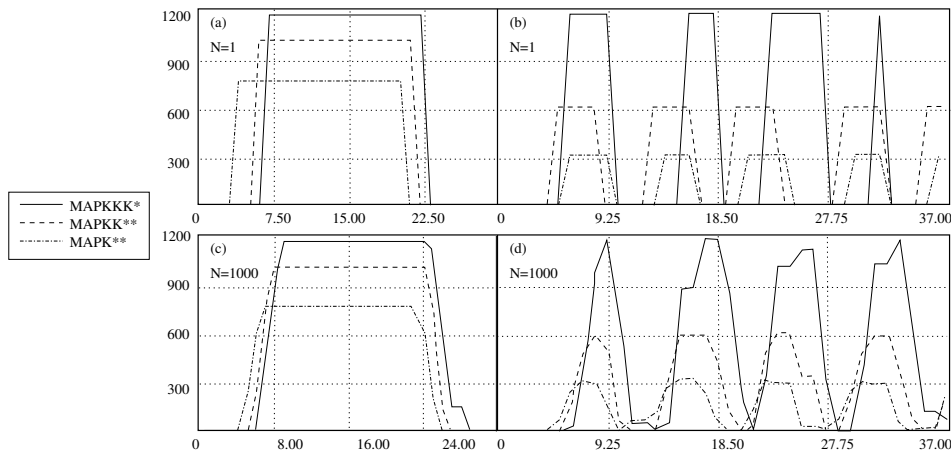


Figure 2.11: Résultats obtenus par Blüthgen [18]. a: modèle de Huang et Ferell ; b: modèle de Kholodenko.



Les deux simulations du modèle de Huang et Ferell (voir les figures a et c) montrent que les concentrations de MAPK-PP (MAPK) sont très proches en amplitude avec celles obtenues par Blüthgen. Notons que, contrairement à Blüthgen, nous n'utilisons pas de donnée cinétique et donc que l'échelle de temps dans nos simulations n'a pas de signification biologique. La première simulation du modèle de Kholodenko avec 1 agent-cellule (voir la figure b) montre des signaux qui ne sont pas véritablement périodiques avec une "periode" moyenne : 7.6 unités de temps et un écart-type : 0.55 unités de temps. Par contre, lorsque le nombre d'agents-cellules augmente, les signaux deviennent davantage périodiques (voir la figure d) avec, par exemple, pour 100 agents-cellules une période moyenne : 8 unités de temps et un écart-type : 0.47 et, pour 1000 agents-cellules une période moyenne : 7.9 unités de temps et un écart-type : 0.45.

Figure 2.12: Résultats obtenus avec des GIF. a: modèle de Huang et Ferell avec N=1 agent-cellule ; b: modèle de Kholodenko avec N=1 agent-cellule ; c: modèle de Huang et Ferell avec N=1000 agents-cellules (moyenne des concentrations) ; d: modèle de Kholodenko avec N=1000 agents-cellules (moyenne des concentrations).

Signalons que dans ces simulations, contrairement à Blüthgen, nous n'utilisons volontairement pas les données cinétiques des réactions biochimiques intervenant dans la voie des MAPK, données disponibles dans la littérature. Nous avons adopté cette démarche car toutes les voies d'activation intracellulaires ne sont pas aussi bien connues que celle des MAPK et donc les données cinétiques ne sont pas toujours accessibles.

2.3.5 Intérêt et utilisation des agents-cellules

L'utilisation des agents-cellules nous semble appropriée dans essentiellement deux cas. Tout d'abord, lorsque l'on cherche à modéliser un phénomène où les cellules impliquées sont relativement bien connues dans leurs comportements de base et dans leurs interactions (immunologie, hématologie, ...). Ensuite, comme nous venons de le voir dans l'exemple précédent, cette approche peut être intéressante lorsque l'on cherche à modéliser l'intérieur d'une cellule et que l'on veut comprendre l'impact d'une modélisation intracellulaire sur un ensemble de cellules (cancérologie, ...).

Nous avons utilisé cette approche agent-cellule dans plusieurs domaines. Citons, notamment, les domaines de l'immunologie (voir la section 2.6.1), de l'hématologie avec la modélisation de la coagulation du sang (voir la section 2.6.2.1) ou de la cancérologie avec la modélisation de voies d'activation de la cellule (voir la section 2.6.3).

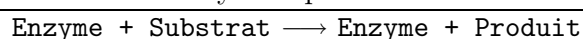
2.4 Agent-réaction

La modélisation d'une cascade de réactions biochimiques commence par le choix du niveau de modélisation :

- Le niveau *microscopique* est le plus bas niveau et le plus général. Il permet facilement d'intégrer des informations spatiales par une équation différentielle déterministe utilisant les lois de Newton. Par contre, celle-ci nécessite six variables par molécule (trois pour la position et trois pour le moment) ce qui rend difficile la modélisation de grands systèmes.
- Le niveau *mésoscopique* tient toujours compte du nombre de molécules mais ne se préoccupe plus ni de leur position ni de leur moment cinétique. Cette variable varie discrètement. On calcule la probabilité de collision entre deux molécules par unité de temps et de volume. On parle alors de probabilité de réaction. A ce niveau, la modélisation de larges systèmes est plus aisée.
- Le niveau *macroscopique* est le plus haut niveau de modélisation d'une cascade biochimique. La concentration de chaque molécule varie de façon continue. On ne parle plus de probabilité de réaction mais de vitesse de réaction. On garde toujours l'hypothèse du milieu homogène.

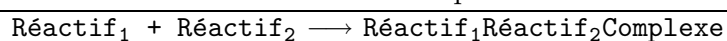
C'est à ce dernier niveau que nous nous plaçons. Plus exactement, nous proposons une méthode multi-agents pour modéliser et simuler la cinétique de réactions biochimiques de deux types :

- les réactions enzymatiques :



(Exemple : un facteur de coagulation en activant un autre, ...).

- les réactions de formation de complexes :



(Exemple : un enzyme qui se lie à son cofacteur, ...).

Plusieurs méthodes ont déjà été utilisées pour simuler des réseaux biochimiques. Citons, par exemple, l'utilisation d'équations différentielles ordinaires avec les travaux de Upinder Bhalla et Ravi Iyengar qui ont modélisé un réseau de signalisation intracellulaire [17] ou, ceux de John Tyson et Bela Novak qui ont proposé un modèle du cycle cellulaire chez les eucaryotes [149]. Citons également l'utilisation de méthodes stochastiques qui tiennent compte d'une probabilité et non d'une vitesse de réaction comme dans les méthodes à base d'équations différentielles ordinaires. Parmi ces méthodes stochastiques, nous trouvons les travaux de Daniel Gillespie qui, en 1976, propose une méthode qui évalue quelle va être la prochaine réaction et le temps pendant lequel elle sera active [50, 51]. Nous trouvons aussi l'utilisation de réseaux de Petri qui peuvent être simplement qualitatifs [122] ou quantitatifs [54, 67, 86]. Citons également le logiciel StochSim [109, 108] qui utilise un algorithme individu-centré stochastique pour la simulation de voies de transduction.

Dans ce qui suit, nous exposons les différents aspects de notre méthode où une réaction biochimique est représentée par un agent : l'individu est la réaction. Une modélisation de ce type est très facilement envisageable compte tenu de la puissance des ordinateurs actuels mais nécessite la connaissance des constantes cinétiques des vitesses de réaction.

2.4.1 Cinétique des réactions biochimiques

La cinétique des réactions enzymatiques et des réactions de formation de complexes se décrit par leur vitesse initiale, c'est-à-dire la vitesse à laquelle les réactifs vont être consommés à $t = 0$ (et ainsi la vitesse à laquelle les produits apparaissent). La vitesse initiale des réactions enzymatiques repose sur le modèle de Henri-Michaelis-Menten, tandis que la vitesse initiale des réactions de formation de complexes suit la cinétique de second ordre.

2.4.1.1 Modèle de Henri-Michaelis-Menten

Vers la fin du XIX^{ième} siècle, des chimistes étudiant certaines réactions enzymatiques n'observaient pas d'évolution de la vitesse initiale de réaction en fonction de la concentration en substrat, alors que d'autres constataient une variation de cette vitesse, mais qui n'était pas en accord avec la loi d'action de masse sous l'hypothèse d'une cinétique du premier ordre [79]. C'est alors qu'apparut l'idée selon laquelle l'enzyme agissait en formant un complexe avec le substrat : à des concentrations faibles de ce dernier, les molécules d'enzyme ne seraient pas toutes mobilisées à la fois pour participer à la réaction, et la vitesse de formation du produit pourrait alors augmenter par addition de substrat jusqu'à ce qu'il soit présent en une quantité telle que l'enzyme soit saturé, la vitesse de réaction ayant ainsi atteint sa valeur maximale. En 1902, Victor Henri proposa la première équation générale de la vitesse initiale des réactions

impliquant des enzymes selon ces hypothèses [63]. Onze ans plus tard, en 1913, Leonor Michaelis et Maud Menten confirmaient les travaux expérimentaux de Henri et présentaient une version légèrement modifiée de son équation [107].

Dans le modèle de Henri-Michaelis-Menten, l'enzyme **E** est supposé réagir avec le substrat **S** pour donner le produit **P** selon le schéma suivant [136] : chaque molécule de substrat se combine avec une molécule d'enzyme pour former un complexe **E·S**. Ce complexe peut soit se dissocier, redonnant une molécule d'enzyme et une molécule de substrat, soit se décomposer, donnant alors une molécule d'enzyme et une molécule de produit **P** (voir la figure 2.13).

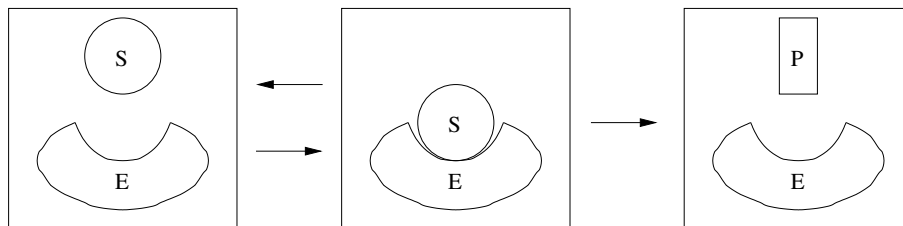


Figure 2.13: Modèle de Henri-Michaelis-Menten.

Une telle réaction se caractérise par l'équation chimique (2.1).



La vitesse initiale de cette réaction est décrite par l'équation (2.2).

$$v = \frac{k_{cat} \times [\text{E}] \times [\text{S}]}{K_M + [\text{S}]} \quad (2.2)$$

En posant $V_{max} = k_{cat} \times [\text{E}]$, la vitesse initiale s'écrit sous la forme de l'équation (2.3).

$$v = \frac{V_{max} \times [\text{S}]}{K_M + [\text{S}]} \quad (2.3)$$

k_{cat} est appelée constante catalytique. K_M est appelée constante de Michaelis. V_{max} est la vitesse de réaction maximale.

La figure 2.14 représente la vitesse initiale de la réaction en fonction de la concentration en substrat. Cette vitesse est donc représentée par une branche d'hyperbole, représentation conforme à l'observation expérimentale.

2.4.1.2 Cinétique de second ordre

L'étude cinétique d'une réaction au cours de laquelle les réactifs **A** et **B** conduisent aux produits **C** et **D** ($\alpha\text{A} + \beta\text{B} \longrightarrow \gamma\text{C} + \delta\text{D}$), où α , β , γ et δ sont les coefficients stoechiométriques, permet de déterminer expérimentalement une vitesse de réaction

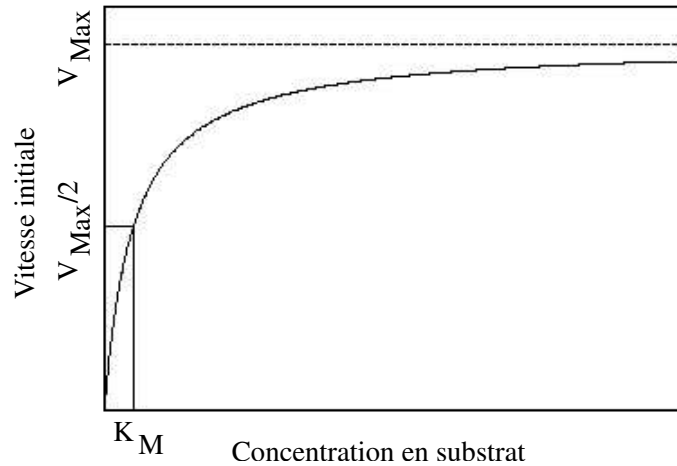


Figure 2.14: Vitesse initiale d'une réaction enzymatique en fonction de la concentration en substrat. La vitesse initiale évolue selon une branche d'hyperbole en fonction de la concentration en substrat.

(dont l'expression est donnée dans l'équation (2.4)) proportionnelle à la concentration de A élevée à la puissance p et à la concentration de B élevée à la puissance q [43].

$$v = k \times [A]^p \times [B]^q \quad (2.4)$$

Les valeurs des puissances p et q sont données par la *mesure expérimentale* de la vitesse de réaction; la somme $p + q$ est l'ordre global de la réaction. Cette mesure expérimentale suit le protocole suivant : la vitesse de la réaction est mesurée en fonction de la concentration de chacun de ces réactifs, pris indépendamment. On fait donc varier la concentration de A en maintenant la concentration de B constante; on établit ainsi la vitesse de formation des produits en fonction de la variation de la concentration en A. Il suffit maintenant de regarder à quoi est proportionnelle cette vitesse :

- si elle est directement proportionnelle à A alors $v = k_1 \times [A]$ et l'ordre p vaut 1,
- si elle est proportionnelle au carré de A alors $v = k_1 \times [A]^2$ et l'ordre p vaut 2,
- si elle est proportionnelle au cube de A alors $v = k_1 \times [A]^3$ et l'ordre p vaut 3,
- ...

Ainsi on dit que l'ordre cinétique de la réaction est p par rapport à A, ou encore que cette réaction est du $p^{ième}$ ordre par rapport à A. On fait ensuite varier la concentration de B, en gardant cette fois la concentration de A constante, et l'on mesure la vitesse de formation des produits en fonction de la variation de concentration de B. De même que pour A, on établit la relation $v = k_2 \times [B]^q$. La vitesse globale de la réaction s'écrit donc $v = k \times [A]^p \times [B]^q$ et l'ordre global de cette réaction est $p + q$.

Une cinétique de second ordre s'écrit donc de manière générale comme indiqué dans l'équation (2.5).

$$v = k \times [A] \times [B] \quad (2.5)$$

A titre d'exemple, dans la coagulation plasmatique, les réactions de formation de complexes sont de second ordre et leur cinétique s'écrit comme dans l'équation (2.6).

$$v = k_{on} \times [R_1] \times [R_2] \quad (2.6)$$

où k_{on} est la constante d'association, R_1 le réactif 1 et R_2 le réactif 2.

2.4.2 Comportement des agents-réactions

Chaque individu (agent) représente donc une réaction du système à modéliser. Le comportement de chaque agent est défini par le cycle de vie suivant :

- **Perception** : lecture des concentrations des composants de la réaction (*i.e.* les réactifs et les produits),
- **Décision** : calcul de la quantité de réactifs et, par conséquent, de la quantité de produits formés,
- **Action** : écriture des nouvelles concentrations des composants de la réaction.

2.4.2.1 Perception

La concentration actuelle des composants impliqués dans la réaction est lue par celle-ci en mémoire. Chaque réaction enzymatique lit la concentration actuelle de l'enzyme, du substrat et du produit, et chaque réaction de formation de complexes lit la concentration actuelle du premier réactif, du deuxième réactif et du complexe.

2.4.2.2 Décision

La réaction va maintenant décider de la quantité de réactif qui va réagir selon ses perceptions. Pour cela, sa vitesse initiale (v) est calculée. La vitesse initiale dépend des concentrations des composants impliqués dans la réaction, ainsi dépend-elle des perceptions de la réaction. La vitesse initiale est conséquemment multipliée par un temps infime (δt , le pas d'intégration) pour déduire la quantité de réactifs (q) à réagir durant δt . C'est une technique d'intégration numérique dérivée de la méthode d'Euler [33], utilisée pour résoudre numériquement des systèmes d'équations différentielles. L'idée est la suivante : $q(t + \delta t) = q(t) + \delta t \times v(t)$. La valeur de δt doit être assez faible pour admettre l'hypothèse que durant δt la vitesse de réaction est constante et égale à la vitesse initiale. En réalité, la vitesse de réaction décroît dans le temps (la vitesse est proportionnelle à la quantité de réactifs présents et ceux-ci sont consommés lors des réactions), mais si la valeur de δt est assez faible on obtient une approximation de bonne qualité et l'erreur numérique reste très faible. Toutefois

des valeurs trop faibles de δt augmentent certes la qualité de l'approximation mais impliquent un allongement important du temps de calcul. Il faut donc équilibrer la balance : qualité de l'approximation/temps de calcul. Enfin, δt introduit la notion de temps dans la simulation : la quantité de réactifs durant δt est calculée à chaque cycle de vie d'une réaction, c'est-à-dire qu'il vient de s'écouler δt secondes une fois le cycle de vie exécuté. Le système évolue ainsi en fonction du temps.

2.4.2.3 Action

La quantité de réactifs que la réaction a consommé vient d'être calculée. L'action de la réaction consiste à mettre à jour les concentrations de ses composants compte tenu de l'action qu'elle a sur ceux-ci. Ainsi la quantité consommée est soustraite aux concentrations originelles des réactifs et est ajoutée aux concentrations originelles des produits.

2.4.2.4 Implémentation du cycle perception-décision-action

L'ensemble de ces concepts est récapitulé figure 2.15 et le code informatique associé est donné figures 2.16 et 2.17. Le langage utilisé ici est le langage orienté agent oRis [59, 60]. L'activation des agents est contrôlée par un ordonnanceur pouvant être placé en mode aléatoire et assurant l'équité entre les agents (voir la section 3.2.3 du chapitre 3).

Cycle	Réaction enzymatique	Réaction de formation de complexes
Perception	$[E]_t = \dots$ $[S]_t = \dots$ $[P]_t = \dots$	$[R_1]_t = \dots$ $[R_2]_t = \dots$ $[R_1R_2\text{Complexe}]_t = \dots$
Décision	$v = \frac{k_{cat} \times [E]_t \times [S]_t}{K_M + [S]_t}$ $\Rightarrow q = v \times \delta t$	$v = k_{on} \times [R_1]_t \times [R_2]_t$ $\Rightarrow q = v \times \delta t$
Action	$[S]_{t+\delta t} = [S]_t - q$ $[P]_{t+\delta t} = [P]_t + q$	$[R_1]_{t+\delta t} = [R_1]_t - q, [R_2]_{t+\delta t} = [R_2]_t - q$ $[R_1R_2\text{Complexe}]_{t+\delta t} = [R_1R_2\text{Complexe}]_t + q$

Le Cycle de vie d'un agent-réaction est du type : Perception-Décision-Action. La perception donne la concentration actuelle de chaque composant (*i.e.* réactifs et produits) impliqué dans la réaction (**E** = enzyme, **S** = substrat, **P** = produit, **R₁** = réactif₁, **R₂** = réactif₂, **R₁R₂Complexe** = le complexe composé du réactif₁ et du réactif₂). La décision calcule la vitesse initiale (v) compte tenu des concentrations des réactifs. La quantité de réactifs qui réagit (q) est déduite en multipliant la vitesse initiale par un temps infime (δt , le pas d'intégration numérique). L'action met à jour les concentrations des composants de la réaction en soustrayant q aux concentrations des réactifs et en ajoutant q aux concentrations des produits.

Figure 2.15: Cycle de vie de chaque agent-réaction.


```

/* Perception */
enzymeConcentration    = milieu->getConcentration(enzyme);
substrateConcentration = milieu->getConcentration(substrate);
productConcentration   = milieu->getConcentration(product);

/* Decision-making */
reactionRate = (kcat * enzymeConcentration * substrateConcentration)
               / (km + substrateConcentration);
involvedConcentration = reactionRate * delta_t;

/* Action */
milieu->setConcentration(substrate, substrateConcentration - involvedConcentration);
milieu->setConcentration(product, productConcentration + involvedConcentration);

```

Cette figure présente le code associé au cycle de vie de chaque agent-réaction enzymatique, Perception : lecture de la concentration de l'enzyme, du substrat, et du produit (la variable `milieu` contient la liste de chaque corps impliqué dans le système à modéliser), Décision : calcul de la vitesse initiale en utilisant la cinétique de Henri-Michaelis-Menten et déduction de la quantité de réactifs par intégration numérique, Action : mise à jour de la concentration du substrat et du produit.

Figure 2.16: Code associé au cycle de vie de chaque agent-réaction de type réaction enzymatique.

```

/* Perception */
reagent1Concentration = milieu->getConcentration(reagent1);
reagent2Concentration = milieu->getConcentration(reagent2);
complexConcentration  = milieu->getConcentration(complex);

/* Decision-making */
reactionRate = kon * reagent1Concentration * reagent2Concentration;
involvedConcentration = reactionRate * delta_t;

/* Action */
milieu->setConcentration(reagent1, reagent1Concentration - involvedConcentration);
milieu->setConcentration(reagent2, reagent2Concentration - involvedConcentration);
milieu->setConcentration(complex, complexConcentration + involvedConcentration);

```

Cette figure présente le code associé au cycle de vie de chaque agent-réaction de formation de complexes, Perception : lecture de la concentration du réactif₁, du réactif₂, et du complexe (la variable `milieu` contient la liste de chaque corps impliqué dans le système à modéliser), Décision : calcul de la vitesse initiale en utilisant la cinétique de second ordre et déduction de la quantité de réactifs par intégration numérique, Action : mise à jour de la concentration du réactif₁, du réactif₂, et du complexe.

Figure 2.17: Code associé au cycle de vie de chaque agent-réaction de type formation de complexes.

2.4.2.5 Eléments de validation : la voie des MAPK

Comme dans le cas des agents-cellules (voir la section 2.3.4.2), nous utilisons la voie intracellulaire des MAPK pour évaluer notre méthode. Pour cela, nous implémentons le modèle de Kholodenko avec deux types d'agents-réactions où toutes les données

biochimiques sont connues :

- des agents-réactions enzymatiques

$$V = \frac{V_{max} * [P]}{K_M + [S]}$$

où V : la vitesse de réaction, V_{max} : vitesse maximale, $[S]$: la concentration en substrat et K_m les constantes de Michaelis-Menten.

- et des agents-réactions rétro-inhibition qui découlent du modèle de Kholodenko :

$$V = \frac{[E1]}{1 + \frac{[MAPK^{**}]}{K_{Loop}}}$$

où K_{Loop} : une constante, $[MAPK^{**}]$: la concentration en MAPK activée et $[E1]$: la concentration en E1.

La figure 2.18 montre que, sur le modèle de Kholodenko, les courbes obtenues par résolution d'équations différentielles ordinaires (Blüthgen [18]) et par agents-réactions sont très similaires que ce soit en amplitude ou en fréquence.

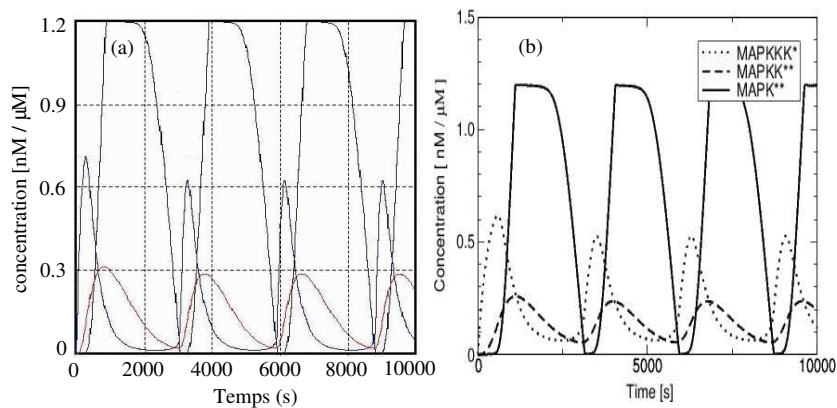


Figure 2.18: Simulation du modèle de Kholodenko : (a) agents-réactions ; (b) ODE.

Grâce aux propriétés dynamiques de notre langage multi-agents oRis (voir la section 3.2.1.3 du chapitre 3), il est possible d'intervenir sur le modèle en cours de simulation. Nous avons donc testé la suppression de la rétro-inhibition sur le modèle de Kholodenko. Ceci se fait simplement par la destruction de l'agent-réaction associé à cette rétro-inhibition. La figure 2.19 montre l'évolution de la concentration en MAPK-PP lors de l'élimination (au temps $T_1=5500$), puis du rétablissement (au temps $T_2=7000$), de la rétro-inhibition. Nous observons une disparition du signal cyclique, puis une réapparition de celui-ci.

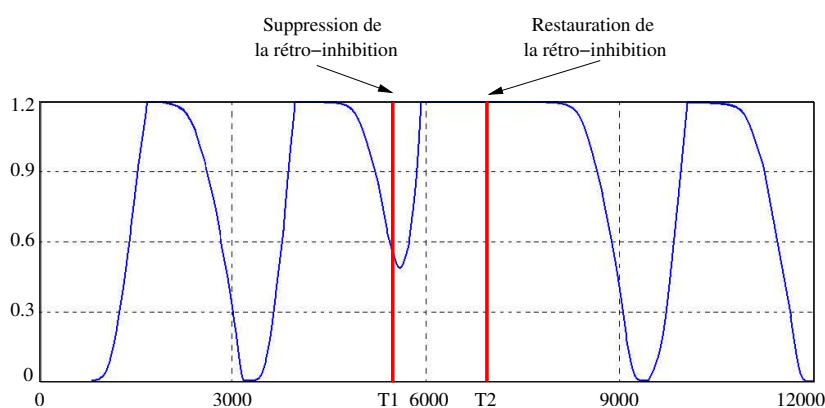


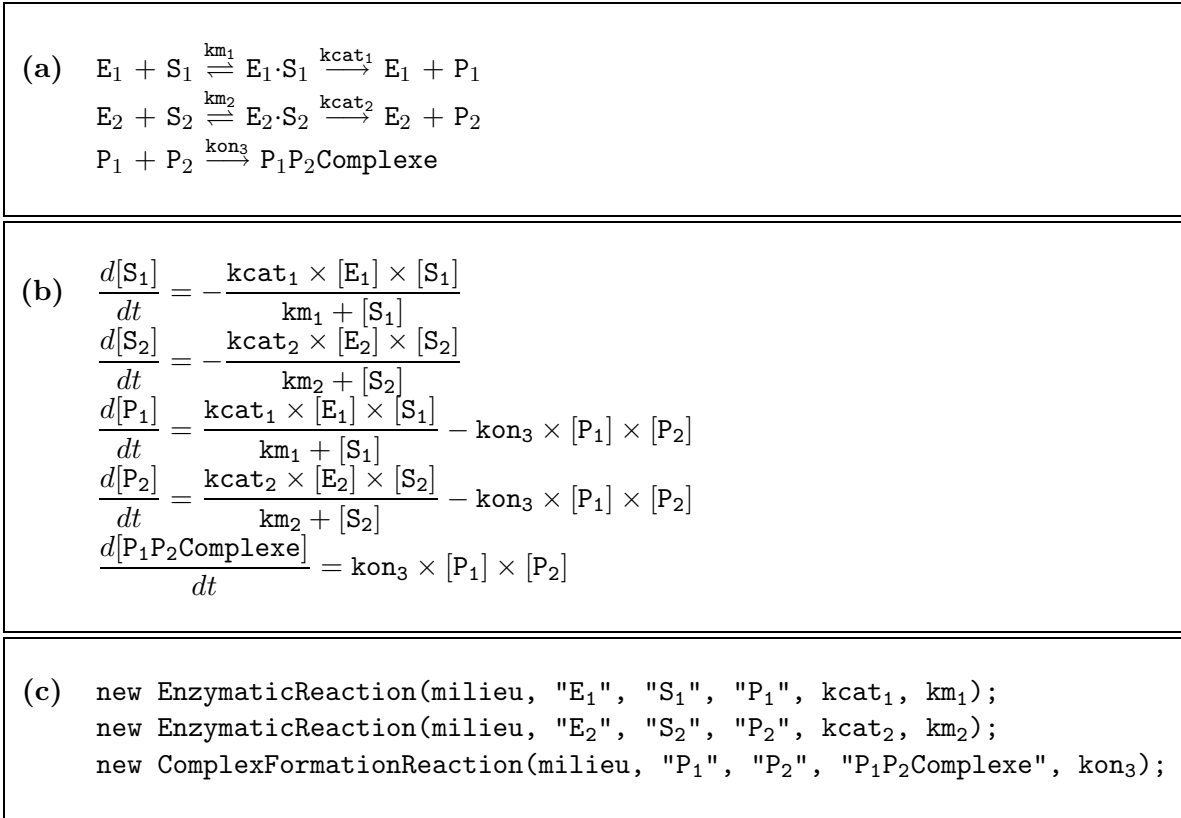
Figure 2.19: Evolution de la concentration en MAPK-PP avec suppression, puis restauration, de la rétro-inhibition.

2.4.3 Intérêt et utilisation des agents-réactions

La question qui vient immédiatement à l'esprit est : quel est l'intérêt de cette approche à base d'agents-réactions par rapport à des méthodes "classiques" utilisant des équations différentielles ordinaires?

Les deux approches simulent un système en construisant un modèle et en l'exécutant sur un ordinateur (calculs *in silico*). Les différences entre les deux méthodes résident dans la forme du modèle et la façon selon laquelle les modèles sont exécutés [115]. En utilisant des agents, le modèle est composé d'un ensemble d'entités décrivant le comportement de chaque individu constituant le système, et l'exécution consiste à émuler ces comportements. En utilisant les équations différentielles ordinaires (ou ODE : Ordinary Differential Equations), le modèle est composé d'un système d'équations différentielles, et l'exécution consiste à le résoudre. Nous avons comparé ces deux méthodes. Notre méthode par agent-réaction donne des résultats très proches de ceux obtenus par la résolution d'un système d'équations différentielles avec un algorithme Runge-Kutta d'ordre 4. L'écart relatif moyen est de 0.015 % avec un écart relatif maximal de 0.035 %.

Afin de bien comprendre les différences, considérons l'exemple présenté figure 2.20. La complexité du modèle à base d'agents est seulement définie par le nombre d'interactions entre les différents corps (3 réactions = 3 agents), alors que la complexité du modèle à base d'ODE est définie par le nombre de corps ainsi que le nombre d'interactions entre eux (5 corps = 5 équations différentielles, P_1 est impliqué dans deux réactions donc deux expressions de vitesses initiales dans son équation différentielle, de même pour P_2).



La figure a représente un système composé de deux réactions enzymatiques et d'une réaction de formation de complexes. La figure b présente la modélisation de ce système sous forme d'ODE. La figure c décrit la modélisation de ce système à base d'agents-réactions. (E_1 = enzyme₁, S_1 = substrat₁, P_1 = produit₁, E_2 = enzyme₂, S_2 = substrat₂, P_2 = produit₂ et $P_1P_2\text{Complexe}$ = le complexe formé par produit₁ et produit₂).

Figure 2.20: Modélisation à base d'ODE *vs.* modélisation à base d'agents-réactions.

Maintenant, supposons que nous voulions ajouter au modèle la réaction enzymatique suivante : $E_3 + S_1 \xrightleftharpoons[km_3]{} E_3 \cdot S_1 \xrightarrow{kcat_3} E_3 + P_1$.

Dans le modèle ODE, il faut modifier le système d'équations et relancer le programme de calcul. Plus exactement, les modifications à apporter sont les suivantes :

$$\frac{d[S_1]}{dt} = -\frac{kcat_1 \times [E_1] \times [S_1]}{km_1 + [S_1]} \implies$$

$$\frac{d[S_1]}{dt} = -\frac{kcat_1 \times [E_1] \times [S_1]}{km_1 + [S_1]} - \frac{kcat_3 \times [E_3] \times [S_1]}{km_3 + [S_1]}$$

et

$$\frac{d[P_1]}{dt} = \frac{kcat_1 \times [E_1] \times [S_1]}{km_1 + [S_1]} - kon_3 \times [P_1] \times [P_2] \implies$$

$$\frac{d[P_1]}{dt} = \frac{kcat_1 \times [E_1] \times [S_1]}{km_1 + [S_1]} - kon_3 \times [P_1] \times [P_2] + \frac{kcat_3 \times [E_3] \times [S_1]}{km_3 + [S_1]}$$

Dans le modèle à base d'agents-réactions, il suffit d'ajouter :

```

new EnzymaticReaction(milieu, "E3", "S1", "P1", kcat3, km3);
                    
```

Grâce aux propriétés dynamiques de notre langage multi-agents oRis, cette modification du système peut se faire à tout moment, sans arrêter la simulation (voir la section 3.2.1.3). C'est ici que se situe l'intérêt principal de notre approche. Il est en effet possible, à tout instant, d'enlever ou d'ajouter une réaction et de voir l'effet immédiat sur les résultats. Ceci peut servir, par exemple, à estimer le moment optimal de la prise d'un médicament.

Nous avons utilisé cette approche agent-réaction dans plusieurs domaines. Citons, notamment, les domaines de l'hématologie avec la modélisation de la coagulation du sang (voir la section 2.6.2.2), de l'allergologie avec la modélisation du récepteur à l'histamine permettant de tester l'effet de différentes molécules [37, 39], ou encore la cancérologie avec la modélisation de voies d'activation de la cellule [119].

2.5 Agent-interaction et phénomène biologique

Dans cette section, nous présentons une démarche beaucoup plus systémique permettant de regrouper nos différentes modélisations de phénomènes biologiques (agent-cellule, agent-réaction). Cette démarche passe par la réification des interactions entre éléments d'un système biologique, c'est-à-dire que nous considérons les interactions comme des agents.

2.5.1 Réification des interactions

La méthode des agents-réactions décrite dans la section précédente fait appel à une approche particulière de la simulation multi-agents de phénomènes physiques. Cette approche part du principe qu'un phénomène est la conséquence des interactions entre ses différents composants, plutôt que le fait des composants eux-mêmes. Les composants sont réduits à de simples variables d'état et les interactions sont réifiées en agents autonomes et concurrents (voir la figure 2.21).

Ceci nous amène à avoir une approche plus systémique concernant les interactions. Cette notion d'interaction est une des plus riches et des plus importantes de la systémique puisqu'elle s'intéresse aux relations entre les constituants d'un système. D'après Joël de Rosnay, "un système est un ensemble d'éléments en interaction dynamique, organisés en fonction d'un but" [32]. Pour modéliser un système complexe, il est nécessaire de considérer les interactions dans ce système comme des échanges portant aussi bien sur de l'énergie, de la matière ou de l'information. Il est ainsi souvent plus utile de connaître la nature des interactions que celle des composants du système.

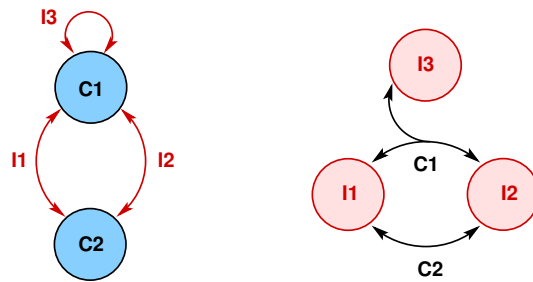


Figure 2.21: Réification des interactions : le schéma de gauche représente la méthode classique de modélisation d'un système avec des nœuds "composants" et des arcs "interactions" ; le schéma de droite représente notre approche dans laquelle les arcs sont transformés en nœuds et inversement. Sur cet exemple, il est possible qu'une interaction ne soit liée qu'à un seul composant.

Figure 2.22, nous décrivons la modélisation d'un phénomène, qui représente un modèle à expérimenter, c'est-à-dire un système. Un phénomène, que nous appellerons également une organisation, est un ensemble composé de deux types d'objets : les composants et les interactions. Les composants représentent un volume délimité dans lequel les entités (cellules, concentration de molécules, ...) sont localisées. Les interactions peuvent être de plusieurs sortes : agents-interfaces pour le transport de matières ou d'énergie, agents-réactions biochimiques. Il est bien sûr possible de considérer d'autres types d'interactions. Notons que les interactions peuvent également être des phénomènes. Nous considérons, en effet, un phénomène comme étant un type particulier d'interaction.

Nous introduisons figure 2.22 des agents-interfaces qui représentent les équations phénoménologiques de la physique des processus irréversibles. Nous considérons ici plus particulièrement les phénomènes de transport. La loi de la conduction thermique de Fourier, la loi de diffusion de Fick et la loi de conduction électrique d'Ohm sont des cas particuliers de ces phénomènes de transport. Les agents-interfaces nous permettent de gérer les échanges de matières entre phénomènes – ou organisations –.

2.5.2 Composition de phénomènes

Cette modélisation par interactions réifiées permet de considérer qu'un phénomène est décomposable en sous-phénomènes eux-mêmes décomposables. De plus, nous avons vu précédemment qu'un phénomène peut correspondre à un ensemble quelconque de composants et d'interactions (voir la figure 2.23). Nous pouvons aussi associer des phénomènes sans qu'il y ait de relation hiérarchique entre eux (voir la figure 2.24).

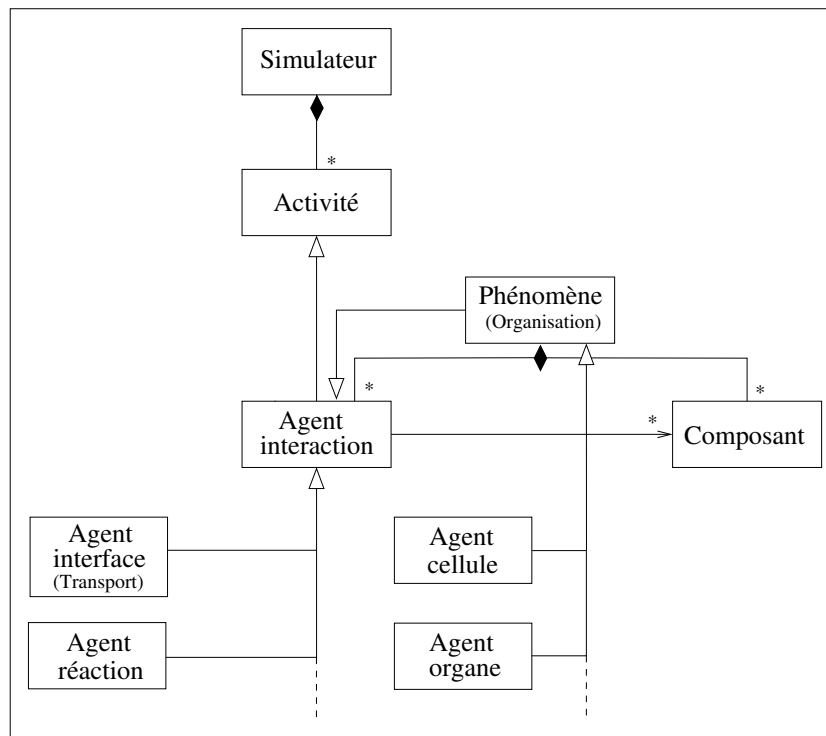


Figure 2.22: Diagramme UML (Unified Modeling Language) décrivant la notion de phénomène et d'organisation.

2.5.3 Intérêt et utilisation des agents-interactions

Le fait de considérer qu'un sous-phénomène est une interaction constitue l'originalité de notre approche. Cela conduit, par exemple, à l'idée qu'une cellule biologique est un type particulier de phénomène et donc un type particulier d'interaction. Cette approche permet de modéliser simplement qu'une cellule est en interaction avec son milieu. Nous avons ainsi une approche multi-modèles dans laquelle des agents de niveaux différents (agent-cellule, agent-réaction, ...) peuvent coopérer.

De cette manière, il est possible de composer à volonté les phénomènes – ou organisations – et ainsi de modéliser un univers complexe avec une vision interaction (voir la figure 2.25).

A l'heure actuelle, nous n'en sommes qu'au début de nos travaux utilisant cette approche par interactions réifiées.

Nous avons utilisé la notion d'interaction entre organisations – ou milieux – dans lesquelles les concentrations de molécules peuvent être différentes. Citons, par exemple, une application en cancérologie (voir la section 2.6.3) où, la concentration en Il6 est un facteur très important sur le développement d'une tumeur. Une cellule cancéreuse pourra ainsi réagir différemment en fonction de l'organisation dans laquelle elle se

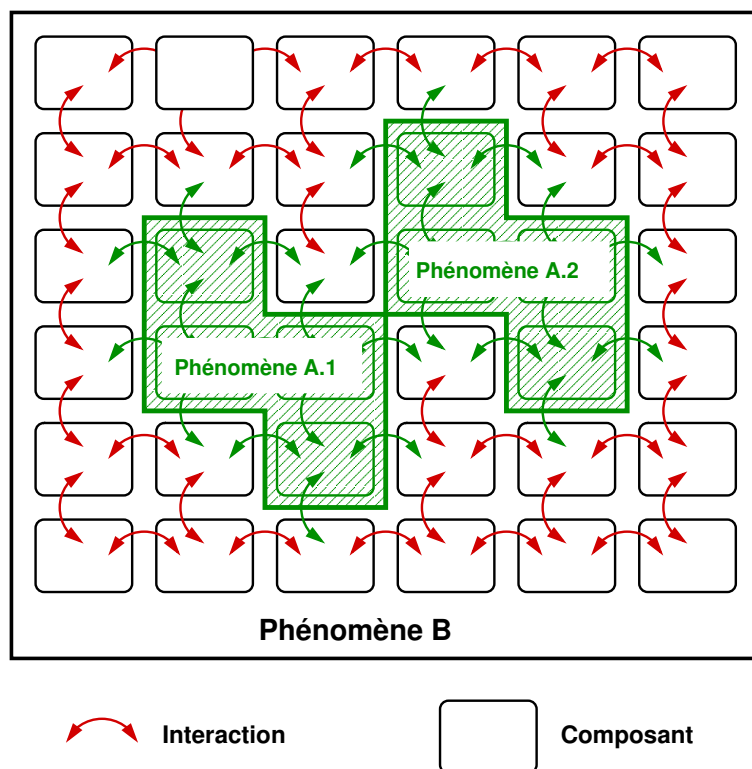


Figure 2.23: Un phénomène est un ensemble quelconque de composants, d'interactions et donc potentiellement de sous-phénomènes. Ici, Phénomène B est composé de Phénomènes A.1 et A.2.

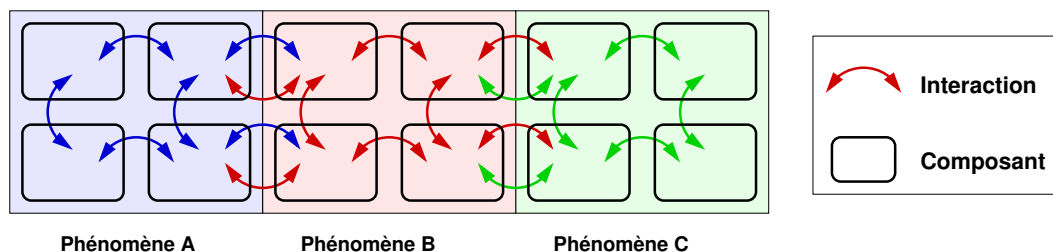


Figure 2.24: Association non hiérarchique de phénomènes.

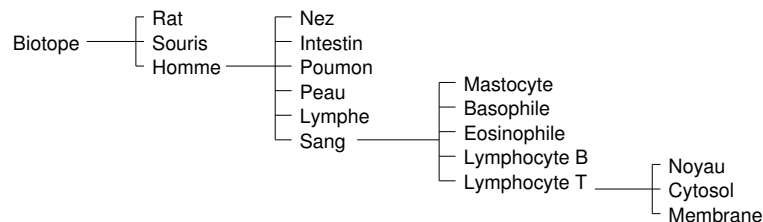
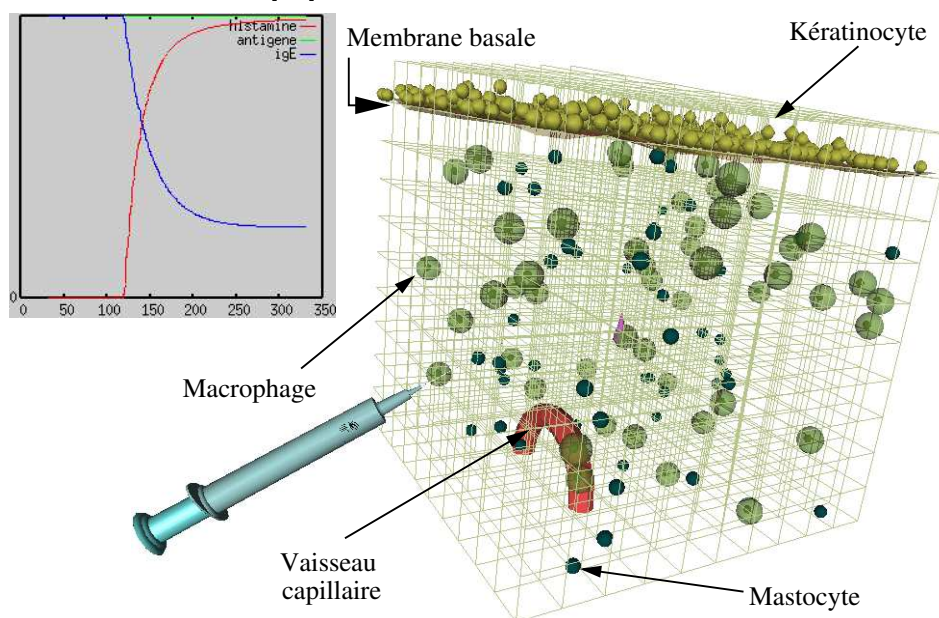


Figure 2.25: De phénomènes pères en en phénomènes fils, il est possible de décrire un biotope. Chaque nœud du graphe représente les interactions du sous-ensemble avec l'environnement.

trouve, organisation qui a sa propre concentration en Il6. Nous envisageons également d'utiliser la composition d'organisations (*i.e.* de phénomènes) afin de modéliser différentes zones de la peau (voir la figure 2.26) intervenant lors d'un urticaire provoqué par une réaction allergique [38].



Cette figure représente un morceau de peau virtuelle correspondant à un cube de $600 \mu\text{m}$ de côté. Une certaine quantité d'antigène peut y être injectée à l'aide d'une seringue virtuelle. Cette injection provoque une réaction allergique libérant de l'histamine. Lorsque l'histamine est au contact d'un vaisseau capillaire, il y a vasodilatation. Cette augmentation du calibre du vaisseau engendre une fuite de plasma entraînant rougeurs et démangeaisons.

Figure 2.26: Modélisation d'un morceau de peau en vue de simuler un urticaire.

2.6 Applications

Nous venons d'étudier différents niveaux de modélisation multi-agents pour l'expérimentation *in virtuo*. Voyons maintenant un certain nombre d'applications dans les domaines de l'immunologie, de l'hématologie et de la cancérologie.

2.6.1 Applications en immunologie

L'immunologie est la discipline qui prend en charge l'étude de la structure du système immunitaire (constitution organique et moléculaire) et de son fonctionnement lors de la pénétration d'un antigène [49]. Le système immunitaire est un système extrêmement complexe dont le rôle est de maintenir l'intégrité de l'individualité biologique d'un être donné, c'est-à-dire le "soi". Ainsi le système immunitaire est capable d'éliminer le "non-soi", encore dénommé antigène, par la mise en œuvre d'une réponse immunitaire.

Cette réponse peut être artificiellement divisée en deux sous-réponses : la réponse humorale et la réponse cellulaire. La réponse humorale entraîne la produc-

tion d'anticorps spécifiques dirigés contre l'antigène. La réponse cellulaire génère des lymphocytes détruisant les cellules de l'organisme infectées par l'antigène. Le système immunitaire est capable de reconnaître plusieurs millions de configurations moléculaires distinctes. Le développement d'une réponse immunitaire repose sur la coopération entre plusieurs types cellulaires. Cette coopération est soit directe, c'est-à-dire par contact membranaire entre deux cellules, soit indirecte, par l'intermédiaire de messagers moléculaires. Ce développement est soumis à de nombreuses régulations qui, soit amplifient la réponse, soit la freinent. Cette notion de régulation pourra être utilisée comme métaphore biologique afin de réguler une population d'agents (voir la section 3.5 du chapitre 3).

Les mécanismes immunitaires sont modélisés à l'aide d'équations différentielles depuis 1966 [61]. En 1974, Niels Jerne propose le modèle de réseau idiotypique qu'il formalise à l'aide d'un ensemble d'équations différentielles [76, 77]. Un système d'équations différentielles permet de décrire l'évolution d'une population de cellules [116] ou de nombreux types d'interactions entre plusieurs populations de cellules. Ce type de modélisation est actuellement largement répandu. Il comporte deux avantages essentiels : simplicité et formalisation. Des solutions analytiques peuvent être trouvées et si ce n'est pas le cas, des simulations numériques peuvent être effectuées. Cependant, cette approche possède ses limites. En effet, une fois un modèle écrit sous forme d'équations différentielles, une modification ou une amélioration de ce modèle engendre la réécriture de la plupart des équations. Cela montre la faible modularité de cette approche. De plus, ce type de modélisation représente le comportement de populations de cellules et non de cellules prises individuellement.

L'approche agent est une autre méthode permettant de modéliser et de simuler des réponses immunitaires. Citons, par exemple, les travaux de Franco Celada et Philip Seiden qui, à partir de 1992 [27, 85, 137], utilisent un automate à états finis généralisé pour simuler une réponse humorale, ainsi que les recherches de l'équipe de Stephanie Forrest [28, 66, 139] orientées sur la modélisation des récepteurs de surface des cellules. L'approche multi-agents est particulièrement adaptée à la modélisation et à la simulation de phénomènes immunitaires. Les notions d'environnement, d'entité autonome, de distribution spatiale, de répartition de rôles, d'interaction, de coopération, de comportements individuels ou collectifs et de récepteurs sont en effet présentes à la fois en immunologie et dans les systèmes multi-agents. L'ajout ou le retrait d'une cellule peut se faire à tout moment par l'apport ou l'élimination d'un agent dans la simulation.

Notre objectif est d'arriver à un modèle multi-agents qui soit le plus proche possible des modèles développés par les immunologistes pour reproduire le plus précisément les phénomènes observés *in vitro* et *in vivo*. Ainsi, l'ordinateur pourra devenir un véritable laboratoire d'expérimentation dans lequel toutes sortes d'observations et de tests pourront s'effectuer *in virtuo*. Afin d'illustrer ces propos, nous donnons dans la suite de cette section deux exemples de modélisation et de simulation en immunologie : la réponse humorale et l'étude de la résistance à l'apoptose du lymphocyte B-CD5. Ces deux exemples sont issus de [11].

2.6.1.1 Modélisation de la réponse humorale

Une des premières expériences que nous avons menées avait pour but de simuler la réponse immunitaire humorale en utilisant plusieurs types d'agents-cellules. Dans ces travaux, cités dans [31], nous avons choisi de simuler la réponse humorale car les cellules entrant en compte sont relativement bien connues dans leurs comportements essentiels et dans leurs interactions de base.

La réponse humorale est un mécanisme qui permet au système immunitaire de générer des anticorps dirigés spécifiquement contre une substance antigénique. Les anticorps ainsi générés se fixent spécifiquement sur les antigènes et en annulent l'activité. Cette réponse utilise différentes cellules du système immunitaire, mais aussi des signaux moléculaires qui permettent aux cellules de communiquer entre elles afin de coordonner leurs actions (voir la figure 2.27). Lorsqu'un antigène pénètre chez un individu, le système immunitaire commence par détecter sa présence grâce essentiellement aux macrophages (phagocytose) et aux lymphocytes B (endocytose), c'est la phase 1. Ces cellules s'activent et deviennent capables de présenter la substance étrangère à d'autres cellules comme les lymphocytes T-CD4 (appelés aussi T4) ou B. D'autres cellules, non modélisées ici permettent de présenter l'antigène. Il s'agit de cellules épithéliales ou dendritiques, mais la voie essentielle est celle des macrophages (monocyte). Juste après la détection, les lymphocytes T-CD4 activés génèrent des messagers chimiques de prolifération et de différenciation appelés interleukines (II), c'est la phase 2. En créant ces molécules, les cellules B et T-CD4 se multiplient (prolifération) et se spécialisent (différenciation), c'est la phase 3. Sous l'action de l'interleukine 4, 5 et 6, les lymphocytes B deviennent des plasmocytes, c'est la phase 4. Ces plasmocytes ont pour rôle la production de molécules d'anticorps se liant spécifiquement à la substance étrangère détectée, c'est la phase 5.

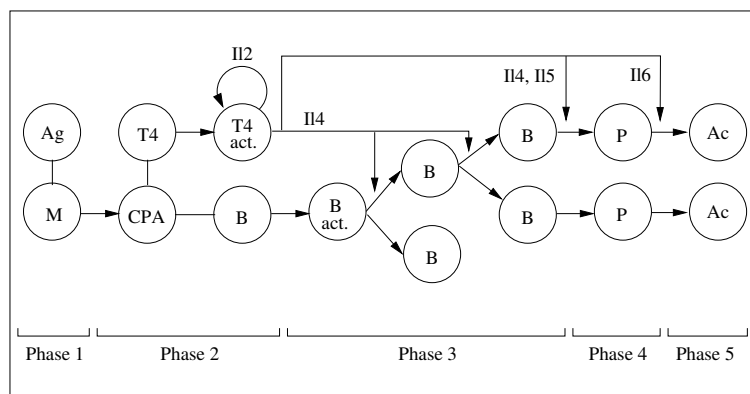


Figure 2.27: Modèle biologique coopératif de la réponse humorale [49].

A partir de ce modèle développé par les immunologistes, nous avons défini un modèle multi-agents équivalent. Celui-ci décrit le comportement des agents ainsi que leurs interactions. Un agent est une simplification d'une cellule ou d'une molécule du système immunitaire. La modélisation et les simulations ont été faites sur la plate-forme

de développement oRis qui permet une programmation orientée agent avancée [59, 60]. Cette plate-forme est décrite plus précisément dans la section 3.2 du chapitre 3.

La réponse humorale telle que nous l'avons modélisée compte neuf types d'agents. Ceux-ci représentent des cellules et des molécules depuis les macrophages qui phagocytent puis présentent les antigènes, jusqu'aux anticorps qui se fixent sur les agents antigènes, en passant par différentes familles d'interleukines. Le modèle que nous présentons utilise les comportements principaux et les interactions essentielles des cellules et des molécules impliquées dans cette réponse immunitaire (voir la figure 2.28). Nous avons modélisé l'environnement comme un espace à 2 dimensions dans lequel les agents se déplacent et interagissent. L'affinité entre antigène et anticorps a été modélisée à l'aide d'un calcul de distance de Hamming sur une chaîne de bits [113]. Un antigène et un anticorps s'attirent donc plus ou moins et, lorsqu'il y a contact entre les deux, ils sont détruits.

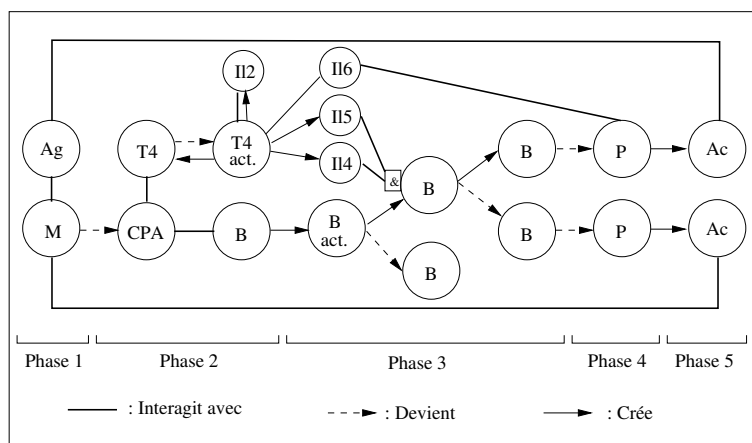


Figure 2.28: Modèle multi-agents coopératif de la réponse humorale.

Les résultats de simulation présentés figure 2.29 sont conformes aux attentes. Nous observons des ressemblances qualitatives (à un facteur d'échelle près) entre l'expérimentation *in vitro* et la simulation *in virtuo*. Dans les deux cas, nous observons la décroissance quasi linéaire des antigènes ainsi que la forte croissance des anticorps suivie d'une lente décroissance.

2.6.1.2 Modélisation des lymphocytes B-CD5

Les expériences *in vitro* effectuées par le Laboratoire d'Immunologie du CHU de Brest ont pour objectif d'étudier les lymphocytes B exprimant les récepteurs CD5 et CD72 [74]. Plus précisément, il s'agit de déterminer le rôle du récepteur CD5 dans la prolongation de l'activation de la cellule ou, au contraire, dans sa mort programmée (apoptose). Ces mécanismes sont mal connus mais le dysfonctionnement des signaux induits par les récepteurs au sein des lymphocytes semble être impliqué dans les Leucémies Lymphocytaires Chroniques [23, 132]. La simulation est utilisée ici afin

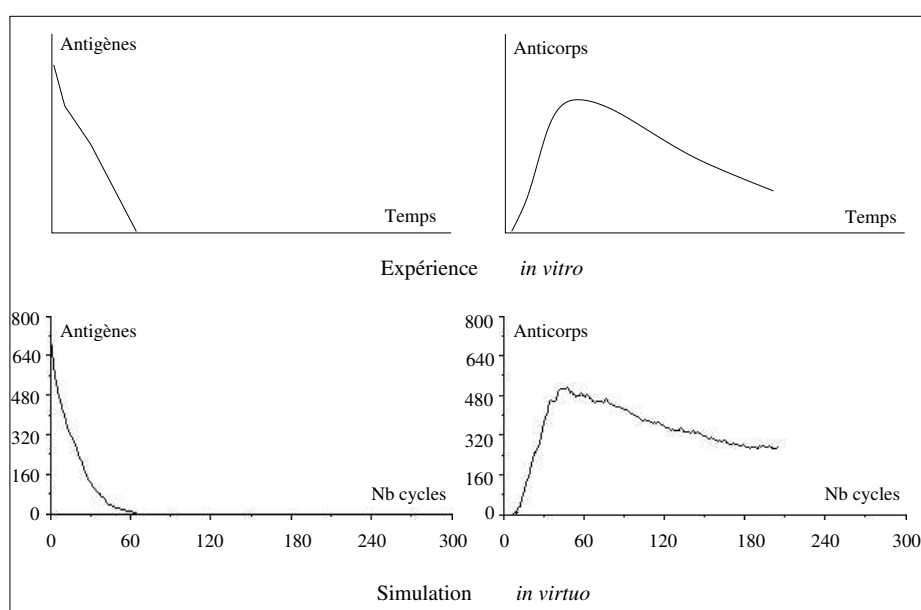


Figure 2.29: Comparaison entre l'expérimentation *in vitro* et *in virtuo* de l'évolution au cours du temps du nombre d'antigènes et du nombre d'anticorps lors d'une réponse humorale.

de tester différentes hypothèses sur le comportement d'une population de lymphocytes représentée par un certain nombre d'agents-cellules. Ces simulations ont deux objectifs. Le premier est de reproduire les expériences déjà effectuées *in vitro* pour valider le modèle. Le deuxième but est de préparer les futures expériences *in vitro* en simulant différents scénarii possibles pour n'en retenir que les plus pertinents.

La molécule CD5 est exprimée par tous les lymphocytes T mais aussi par une sous-population de lymphocytes B. Les lymphocytes B au repos ne prolifèrent pas en réponse à la liaison de la molécule CD5, alors que ces mêmes cellules préactivées par un anticorps IgM (Immunoglobuline de type M) et de l'interleukine 2 (Il2) se multiplient. Nous étudions spécifiquement les effets de la molécule d'anti-CD5 et de la molécule d'anti-IgM sur l'apoptose des lymphocytes B-CD5⁺ (le + indique la présence de la molécule CD5). La liaison du récepteur CD5 avec l'anti-CD5 ou celle du BCR (B Cell Receptor) avec l'IgM implique l'apoptose des lymphocytes.

L'apoptose commence plus tôt avec la liaison CD5 qu'avec la liaison IgM. Les cellules B-CD5⁺ sont plus sensibles à l'apoptose que les cellules B-CD5⁻ lorsque le BCR est impliqué. Dans les mêmes conditions, les récepteurs CD5 exprimés par les cellules T n'entraînent pas d'apoptose en cas de liaison. L'hypothèse formulée par les immunologistes est la suivante : les liaisons conjointes des récepteurs CD5 et BCR avec leur ligand influent sur le retard à l'apoptose.

La simulation porte donc sur le modèle donné par les immunologistes. Seuls les comportements, les récepteurs et les molécules impliquées dans les expériences *in vitro*

sont à reproduire. Cela limite grandement la complexité de la modélisation. L'agent lymphocyte B-CD5 ne comporte donc que 4 types de récepteurs : Le BCR, l'IL2, le CD5, et le CD72. Le BCR rend possible l'activation de l'agent lymphocyte. Le récepteur d'interleukine 2 permet la division cellulaire (mitose). Les récepteurs CD5 et CD72, supposés complémentaires, agissent sur l'activation du lymphocyte en influant sur l'apoptose. Si ces récepteurs sont liés, la durée de vie de la cellule augmente, sinon elle diminue (voir la figure 2.30).

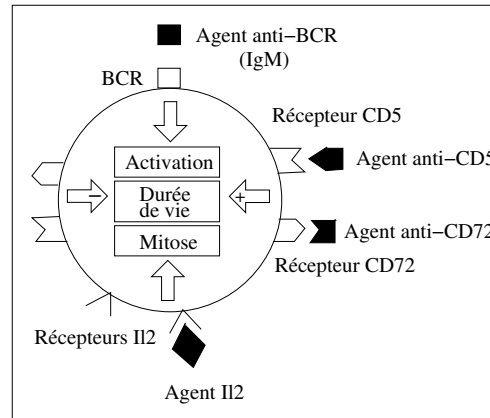


Figure 2.30: Modèle du lymphocyte B-CD5 et des agents molécules.

Le protocole expérimental réalisé par le Laboratoire d'immunologie du CHU de Brest consiste à placer dans un tube à essai, où sont présents des molécules d'interleukine et d'anti-BCR, des lymphocytes B-CD5. Les cellules s'activent et prolifèrent. Trois jours plus tard, au maximum de la prolifération, des anticorps anti-CD5 sont ajoutés ou non dans le tube. Un retard à l'apoptose est alors observé lors de l'ajout d'anti-CD5.

Pour simuler ce protocole expérimental, nous devons effectuer deux simulations *in virtuo* où nous plaçons aléatoirement 10 agents B-CD5, 100 agents IL2 et 100 agents anti-BCR à $t=0$. Chaque agent B-CD5 possède 10 récepteurs IL2, 10 récepteurs BCR et 10 récepteurs CD5 tous uniformément répartis sur la membrane cellulaire. Un récepteur de l'agent représente 10^4 récepteurs réels. Nous laissons ensuite évoluer le modèle et observons une prolifération des agents B-CD5 jusqu'à $t=3$ jours. A ce moment précis, nous injectons 100 agents anti-CD5 dans l'une des deux expériences *in virtuo*. Cette simulation montre clairement un maintien de la population des agents B-CD5 pendant une plus grande durée (2 jours) par rapport à l'expérience sans anti-CD5 [11, 12]. La figure 2.31 présente les résultats issus des deux simulations (avec et sans ajout d'anti-CD5).

La simulation donne ici des résultats quantitativement proches de la réalité. Le modèle donné par les immunologistes est donc une piste possible pour comprendre le phénomène réel. C'est ainsi que la simulation doit être perçue, c'est-à-dire un outil d'aide à la réflexion, permettant à la fois de tester des idées, d'omettre le moins possible de paramètres et de renforcer la valeur d'un modèle.

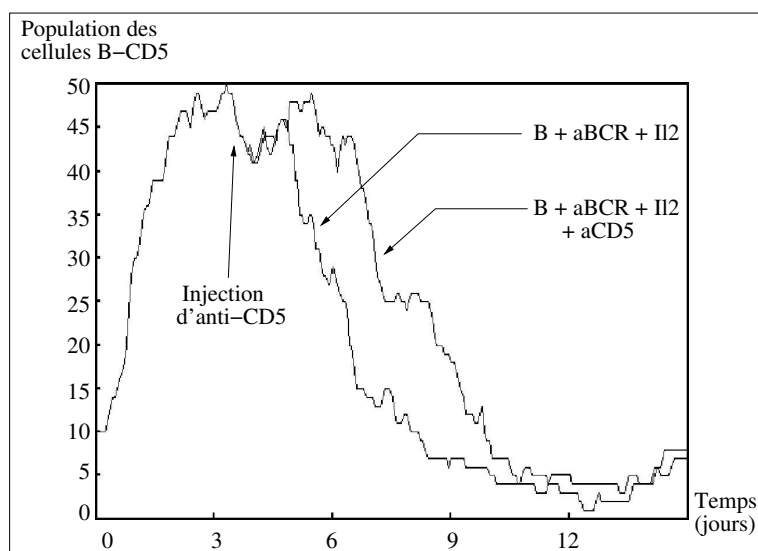


Figure 2.31: Evolution des populations des agents B-CD5 avec et sans introduction d'anti-CD5.

2.6.2 Applications en hématologie

L'hématologie étudie la physiologie et la pathologie du sang. Composé de cellules, le sang est un tissu fluide circulant dans les vaisseaux. Le sang transporte de nombreuses cellules et molécules ayant des rôles multiples comme les globules rouges transportant l'oxygène, des cellules du maintien de l'intégrité vasculaire, des molécules d'éléments nutritifs ou encore des composants du système immunitaire. L'hématologie se préoccupe aussi des organes qui créent ou qui éliminent les cellules et les molécules du sang. Un des premiers traités d'hématologie, écrit par Gabriel Andral en 1843, porte le titre d'"Essai d'hématologie pathologique" [4]. Depuis cette date, l'hématologie n'a cessé d'être présente sur le devant de la scène scientifique avec notamment la naissance de la pathologie moléculaire, la grande victoire de la transfusion sanguine, la recherche sur les leucémies, les greffes, l'hémostase avec les études sur l'hémophilie ou les thromboses.

Le sang circule en circuit fermé avec une pression relativement élevée. Des fuites peuvent apparaître, entraînant des hémorragies. Des caillots parfois se forment, bloquant ainsi la circulation. Le terme d'hémostase désigne les processus mis en œuvre par le système sanguin pour colmater les fuites et rétablir la circulation sanguine. L'hémostase se divise en trois étapes. L'hémostase primaire dure de 3 à 5 minutes pendant lesquelles des cellules, appelées plaquettes, se regroupent en un agrégat pour stopper l'hémorragie. La coagulation plasmatique, d'une durée de 5 à 10 minutes, consolide l'agrégat ou thrombus plaquettaire, grâce à la formation de molécules de fibrine. La fibrinolyse est la troisième étape de l'hémostase. En 48 à 72 heures, le caillot formé par les plaquettes et les molécules les cimentant est dégradé. Le sang circule à nouveau normalement.

Les deux premières étapes de l'hémostase sont regroupées sous le nom plus courant de coagulation. Cette coagulation, pour être efficace, doit se produire précisément à l'endroit où se trouve la brèche, avoir la bonne taille et être d'une durée suffisante pour que s'opère la réparation du vaisseau sanguin, mais elle ne doit pas être trop longue pour ne pas perturber la circulation.

Dans la suite de cette section, nous donnons tout d'abord plus de détails sur la coagulation. Puis, nous en présentons deux modélisations multi-agents : l'une basée sur l'approche agent-cellule et l'autre reposant sur la notion d'agent-réaction. L'intérêt de ces travaux est, d'une part, de simuler des maladies comme l'hémophilie ou la thrombophilie et, d'autre part, de tester l'effet de médicaments sur ces maladies.

La coagulation est le processus physiologique qui se met en place en réponse à une brèche vasculaire [78]. Il permet d'arrêter une hémorragie grâce à la formation rapide d'un caillot constitué de plaquettes agrégées entre elles et enserrées dans un réseau de fibrine au niveau de la brèche vasculaire.

Plus précisément, une brèche dans un vaisseau sanguin découvre un certain nombre de molécules et de cellules qui normalement sont cachées par l'endothélium, la couche supérieure de la veine. La paroi d'une veine est composée de cellules dites endothéliales. Les éléments se trouvant en dessous portent le nom de sous-endothélium. Les plaquettes circulant dans le sang sont attirées par les cellules et les molécules du sous-endothélium et viennent s'agglutiner et s'activer à leur contact. Comme activateur et attracteur des plaquettes, nous trouvons des molécules comme, par exemple, le facteur Willebrand ou le collagène. Une autre molécule, le facteur tissulaire (TF), appelé également facteur III, se trouve à la surface de cellules appelées fibroblastes. Ce facteur tissulaire, présent dans le sous-endothélium, est à l'origine d'une cascade d'événements qui amplifie de manière considérable la coagulation. En effet, les molécules de facteur VII circulant dans le sang sont activées par le facteur tissulaire. En tout, ce sont plus de douze types de facteurs qui, associés à des protéines procoagulantes ou anticoagulantes, entrent en jeu avec de multiples effets d'amplification et de régulation. Finalement, le fibrinogène (facteur I) circulant se voit activé pour devenir la molécule qui maintient fermement le thrombus plaquettaire, c'est-à-dire la fibrine (facteur Ia).

Signalons que le thrombus formé au cours de l'hémostase primaire ne permet qu'une hémostase imparfaite. Il doit être consolidé par de la fibrine pour être solide : le rôle de la coagulation plasmatique consiste à fournir cette fibrine. On distingue deux voies responsables de la génération de thrombine, enzyme clef permettant la formation du caillot : la voie extrinsèque, initiatrice de la coagulation, et la voie intrinsèque, génératrice de la quantité de thrombine nécessaire à la création de fibrine afin d'obturer la brèche par le caillot.

Les deux modèles que nous présentons plus loin prennent en compte les principaux phénomènes et entités connus dans la coagulation : 3 types de cellules (plaquettes, cellules endothéliales et fibroblastes) et 32 types de protéines sont impliqués dans 42 interactions. La figure 2.32 détaille ces interactions. Les différents facteurs plasmatiques

sont représentés à l'aide de chiffres romains. Le suffixe "a" indique qu'un facteur est présent sous sa forme activée.

Réactifs	Produits	Description	Réactifs	Produits	Description
Voie intrinsèque			AT3 + IXa	0	Inhibition
TF + VIIa	VII:TF	Coagulation	AT3 + Xa	0	Inhibition
VII + VIIa	VIIa + VIIa	Coagulation	AT3 + XIa	0	Inhibition
VII + VII:TF	VIIa + VII:TF	Coagulation	AT3 + IIa	0	Inhibition
IX + VII:TF	IXa + VII:TF	Coagulation	alpha2M + IIa	0	Inhibition
X + VII:TF	Xa + VII:TF	Coagulation	TM + IIa	IIi	Inhibition
TFPI + Xa	TFPI:Xa	Inhibition	AT3 + IXa	0	Inhibition
TFPI:Xa + VII:TF	0	Inhibition	AT3 + Xa	0	Inhibition
Voie extrinsèque			AT3 + XIa	0	Inhibition
VII + IXa	VIIa + IXa	Coagulation	AT3 + IIa	0	Inhibition
IXa + X	IXa + Xa	Coagulation	ProC + IIi	PCa + IIi	Inhibition
II + Xa	IIa + Xa	Coagulation	PCa + Va	0	Inhibition
XIa + IX	XIa + IXa	Coagulation	PCa + Va:Xa	Xa	Inhibition
XIa + XI	XIa + XIa	Coagulation	PCa + PS	PCa:S	Inhibition
IIa + XI	IIa + XIa	Rétro-activation	PCa + PCI	0	Inhibition
IIa + VIII	IIa + VIIIa	Rétro-activation	PCa:S + Va	0	Inhibition
IIa + V	IIa + Va	Rétro-activation	PCa:S + V	PCa:S:V	Inhibition
Xa + V	Xa + Va	Rétro-activation	PCa:S:V + VIIIa:IXa	IXa	Inhibition
Xa + VIII	Xa + VIIIa	Rétro-activation	PCa:S:V + VIIIa	0	Inhibition
Xa + Va	Va:Xa	Prothrombinase	Activation des plaquettes		
VIIIa + IXa	VIIIa:IXa	Ténase	P + IIa	Pa + IIa	Coagulation
VIIIa:IXa + X	VIIIa:IXa + Xa	Ténase	Formation de fibrine		
Va:Xa + II	Va:Xa + IIa	Prothrombinase	I + IIa	Ia + IIa	Coagulation

Le processus de coagulation s'initie à la surface des fibroblastes. C'est au niveau du facteur tissulaire (TF) à la surface des fibroblastes que va s'initier le processus de coagulation. La voie dite intrinsèque est mise en action : le facteur VII activé (VIIa) se lie au facteur tissulaire et permet l'activation des facteurs VII, IX et X. Les premières molécules de thrombine (II), facteur clef du phénomène, sont générées, ce qui permet de lancer l'action de la voie extrinsèque et la cascade proprement dite de la coagulation. La thrombine rétroactive les facteurs XI, V, VIII et surtout les plaquettes (P). A la surface des plaquettes ainsi activées, les complexes ténases et prothrombinases sont alors en mesure de se créer et d'apporter leur importante contribution à la formation de thrombine. La thrombine active aussi le fibrinogène en fibrine qui vient alors lier les plaquettes activées (Pa) entre elles présentes au niveau de la brèche. Le caillot plaquettaire se forme. Simultanément, les inhibiteurs agissent : le TFPI se lie au facteur X activé (Xa) pour inhiber la voie intrinsèque (facteur VIIa lié au facteur tissulaire), l'alpha2macroglobuline inhibe la thrombine, l'antithrombine3 inhibe la thrombine, les facteurs IX, X, XI activés (IIa, IXa, Xa, XIa). La protéine C est activée par le complexe thrombine-thrombomoduline (IIi : *thombin inhibitor*) se formant à la surface des cellules endothéliales. La protéine C activée (PCa) a deux chemins d'action : elle inhibe directement le facteur V activé (Va) et indirectement, liée à la protéine S et au facteur V, elle inhibe le facteur VIII activé (VIIIa). Le PCI (*protein C inhibitor*) empêche quant à lui l'action inhibitrice de la protéine C activée (PCa).

Figure 2.32: Les interactions mises en œuvre dans nos modèles de coagulation.

Face à une telle complexité, il est difficile pour l'esprit humain de déterminer précisément l'influence d'un facteur sur un autre. La puissance de calcul de l'ordinateur permet d'observer ces influences. De plus, l'étude des résultats d'une coagulation virtuelle permet de déterminer les causes les plus probables d'un dysfonctionnement et même de déterminer un éventuel traitement en repérant le levier biologique sur lequel jouer.

Etudions maintenant deux modélisations multi-agents de la coagulation : l'une basée sur l'approche agent-cellule et l'autre reposant sur la notion d'agent-réaction. La

validation de nos différents modèles repose sur plusieurs éléments :

- la similitude des courbes de génération de thrombine obtenues *in vitro* (voir la figure 2.33) et *in virtuo* via nos modèles,
- la cohérence avec les pathologies : diminution de thrombine générée dans l'hémophilie et augmentation dans la thrombophilie,
- le traitement de l'hémophilie par le NovoSeven qui est un médicament composé essentiellement de facteurs VII activés,
- le traitement de la thrombophilie par l'héparine qui exerce principalement une action anticoagulante en se liant à l'antithrombine et en catalysant l'inhibition de plusieurs facteurs activés de la coagulation sanguine : XIIa, XIa, IXa, Xa et thrombine.

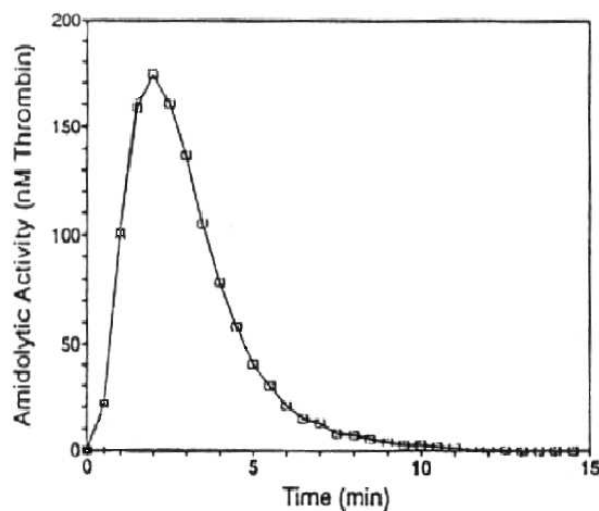


Figure 2.33: Courbe de génération de thrombine obtenue *in vitro* par sous-échantillonnage, d'après Hendrik Coenraad Hemker [62].

2.6.2.1 Modélisation de la coagulation : agent-cellule

Dans cette première modélisation de la coagulation, nous avons modélisé l'environnement par une surface de $200 \times 200 \mu m^2$ (voir la figure 2.34). La paroi de la veine est composée de deux niveaux. L'endothélium (cellules endothéliales) et le sous-endothélium (fibroblastes et facteurs de Willebrand). Nous avons ensuite introduit des agents-cellules ayant des comportements très simples reprenant les réactions décrites figure 2.32. Afin d'avoir des résultats quantitatifs, un agent de type protéine représente en fait plusieurs protéines. Le nombre total d'agents est d'environ 3000. Ces agents correspondent, soit à des cellules (endothéliales, fibroblastes, plaquettes), soit à des protéines de la coagulation. Des données plus précises sont disponibles dans [11].

Afin de valider notre modèle, nous avons effectué plusieurs simulations (voir la figure 2.35). La première est une simulation supposée reproduire une coagulation

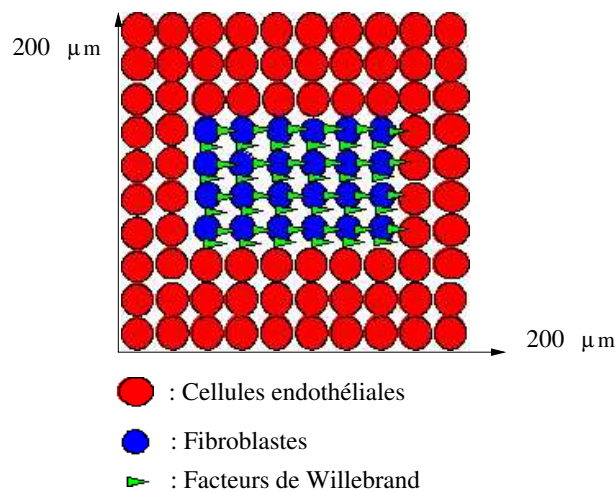
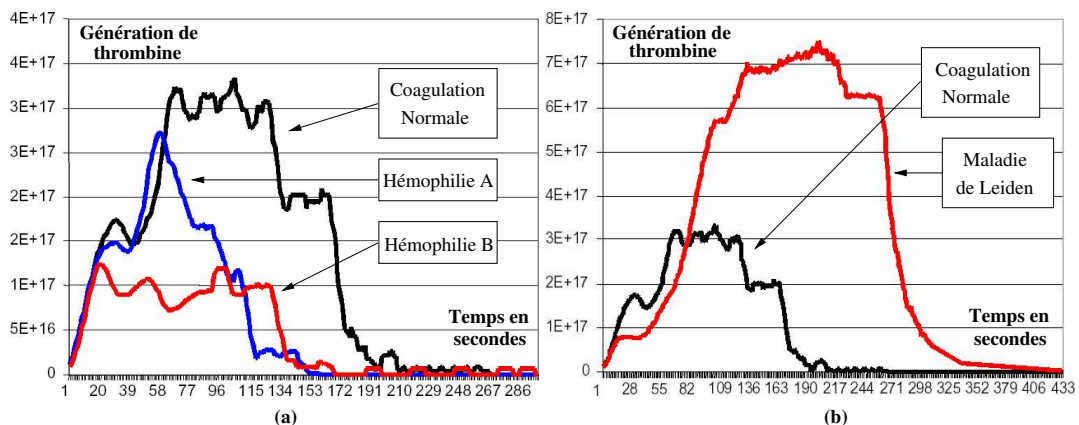


Figure 2.34: Vue de dessus de la paroi de la veine.

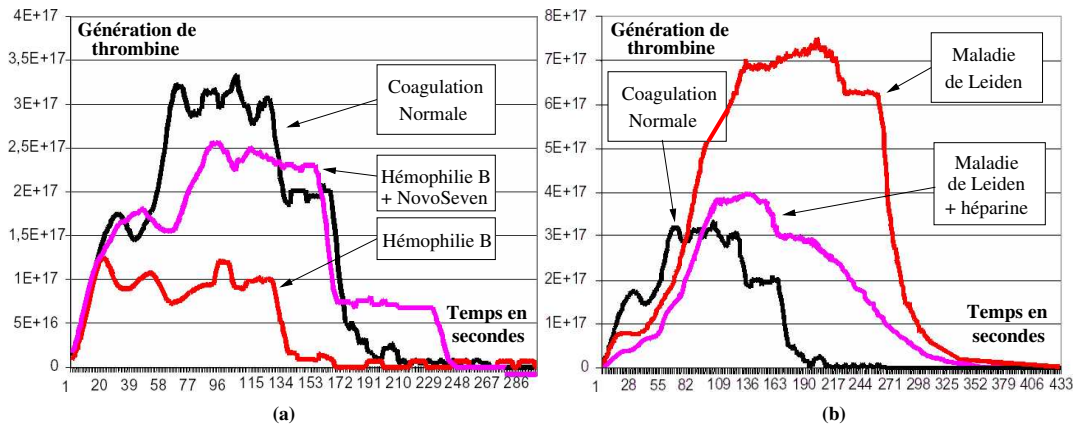
normale (voir la figure 2.35-a). La deuxième et la troisième s'attachent à simuler deux types d'hémophilie : l'hémophilie A et hémophilie B (voir la figure 2.35-a). L'hémophilie A, la plus fréquente (85 % des cas), est due à un déficit en facteur VIII. L'hémophilie B (15 % des cas) est due à un déficit en facteur IX. Enfin, la quatrième correspond à la simulation de la maladie génétique de Leiden qui, à cause d'une mutation du facteur V, favorise l'apparition de thromboses veineuses, c'est-à-dire la formation de caillots dans le sang (voir la figure 2.35-b). Cette mutation inhibe la désactivation du facteur V activé par la protéine C activée.



La figure a compare les résultats de simulation dans le cas d'un patient sain et dans le cas de patients atteints d'hémophilie A ou d'hémophilie B. Nous remarquons une sensible différence de coagulation entre les hémophiles et le patient sain : les hémophiles coagulent moins qu'une personne saine. La figure b compare les résultats de simulation dans le cas d'un patient sain et dans le cas d'un patient atteint de la maladie de Leiden. Notre simulation montre clairement un excès de génération de thrombine dans cette maladie par rapport à une coagulation normale. La simulation est donc en accord avec les connaissances en hématologie et confirme le fort risque de thrombose lié à cette déficience. Notons également que la courbe de génération de thrombine d'un patient sain présente une forte croissance suivie d'une forte décroissance. Notre modèle semble donc donner des résultats cohérents avec ceux obtenus par Hemker (voir la figure 2.33).

Figure 2.35: Diverses simulations de coagulation : patient sain, patients malades (hémophilie A, hémophilie B, maladie de Leiden).

Notre modèle permet d'ajouter à n'importe quel moment de la simulation n'importe quel type d'agent. Nous pouvons donc envisager l'introduction de molécules dans nos simulations afin de soigner nos patients virtuels (voir la figure 2.36). Nous nous proposons de tester l'effet de la prise de NovoSeven (facteur VIIa) sur un malade souffrant d'hémophilie B (voir la figure 2.36-a). Nous testons également un antithrombotique (l'héparine) sur un patient atteint de la maladie de Leiden (voir la figure 2.36-b).



La figure a compare les courbes de génération de thrombine dans les cas de la coagulation normale, de l'hémophilie B (absence de facteur IX) et de l'hémophilie B traitée avec du NovoSeven, c'est-à-dire en ajoutant des facteurs VII activés. Nous remarquons que la coagulation est bien meilleure avec ce médicament. La figure b compare les courbes de génération de thrombine dans les cas de la coagulation normale, de la maladie de Leiden et de la maladie de Leiden traitée avec de l'héparine. La courbe de génération de thrombine montre qu'un patient atteint par la maladie de Leiden coagule beaucoup moins lorsqu'il est sous héparine.

Figure 2.36: Diverses simulations de traitement : hémophilie B traitée avec du NovoSeven et maladie de Leiden traitée avec de l'héparine.

Ces expériences de coagulation virtuelle indiquent que notre modélisation agent-cellules donne des résultats conformes aux attentes. Les courbes obtenues sont quelque peu accidentées car le nombre d'agents impliqués dans les simulation est relativement faible ($\simeq 3000$). Un plus grand nombre d'agents amènerait certainement à un "lissage" du comportement collectif à partir de comportements individuels plus nombreux.

Etudions maintenant une deuxième modélisation de la coagulation basée sur la notion d'agent-réaction qui, à l'image des équations différentielles, permet de retrouver l'évolution d'une population d'entités au cours du temps.

2.6.2.2 Modélisation de la coagulation : agent-réaction

Basée sur les agents-réactions, cette deuxième modélisation permet de simuler la cinétique de la cascade de réactions de la coagulation (voir la figure 2.32). Rappelons que la cinétique des réactions peut être décrite au niveau moléculaire ou au niveau de la réaction [106].

Au niveau moléculaire, la modélisation de la cinétique par un système multi-agents consiste à modéliser une molécule par un agent. Dans le cas de la coagulation plasma-

tique, le nombre de molécules présentes étant très important, seul le comportement d'un petit nombre d'entités peut être reproduit (voir la section 2.6.2.1).

Au niveau de la réaction, la modélisation de la cinétique par un système multi-agents consiste en la représentation d'une réaction par un agent. Dans le cas de la coagulation plasmatique – comme dans tous les systèmes biologiques impliquant des cascades de réactions –, le nombre de réactions est largement plus faible que le nombre de molécules : une modélisation de ce type est très facilement envisageable compte tenu de la puissance des ordinateurs actuels. Qui plus est, dans le cas particulier de la coagulation plasmatique, les constantes cinétiques des vitesses de réaction sont disponibles dans la littérature [92, 155, 156]. Ainsi dans notre approche, une réaction biochimique sera représentée par un agent : l'individu est la réaction.

Afin de valider notre modèle, nous avons effectué plusieurs simulations utilisant des agents-réactions. Ces simulations reproduisent la coagulation d'un patient sain, de malades souffrant d'hémophilie A ou d'hémophilie B et d'un patient atteint de la maladie de Leiden (voir la figure 2.37). La figure 2.38 correspond aux résultats de simulation de traitement : un hémophile de type B prend du NovoSeven et un patient atteint de la maladie de Leiden prend de l'héparine.

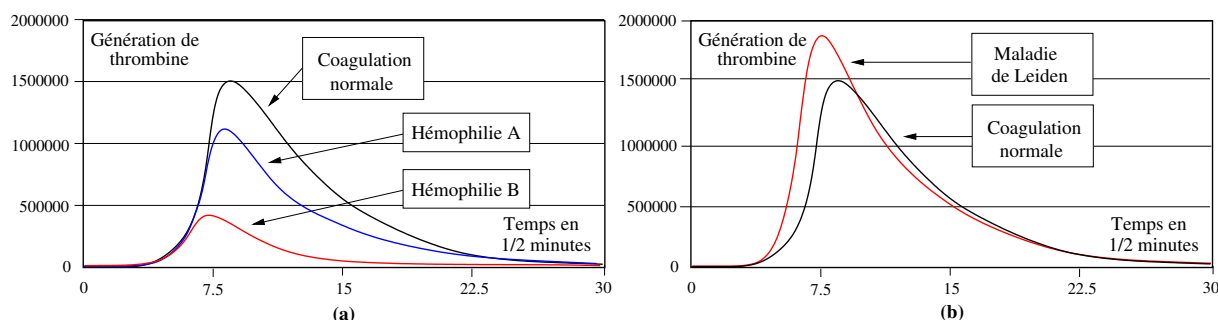
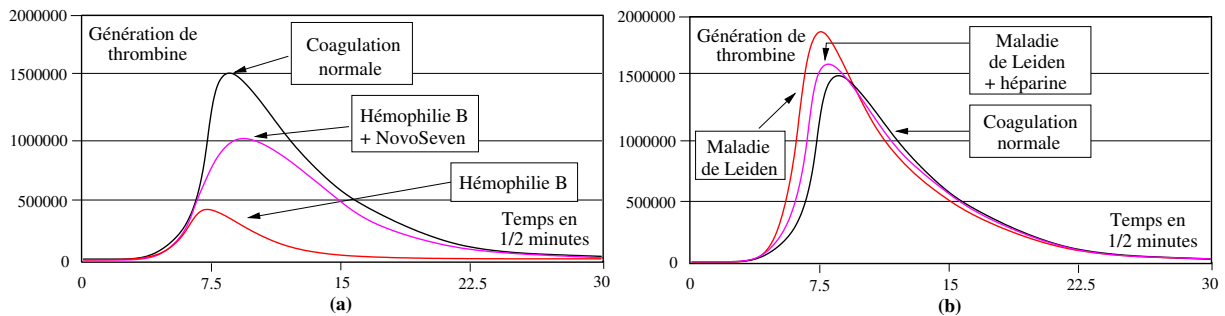


Figure a, nous pouvons observer qu'un hémophile génère moins de thrombine qu'une personne non malade. Figure b, nous voyons également qu'une personne atteinte de la maladie de Leiden génère plus de thrombine qu'une personne saine. En outre, ce malade virtuel a une génération de thrombine accélérée par rapport à une personne saine. D'anciens travaux ont montré que l'inhibition de la protéine C agissait à la fin du phénomène de coagulation, lorsqu'une grande quantité de thrombine avait été générée. Des travaux récents ont montré l'inverse, c'est-à-dire que la protéine C agissait dès le début de la coagulation. La figure b semble vérifier ces travaux récents. Dans le cas de ces deux maladies, notre modèle est cohérent avec les résultats biologiques connus.

Figure 2.37: Diverses simulations de coagulation : patient sain, patients malades (hémophilie A, hémophilie B et maladie de Leiden).

Notre modèle de la coagulation utilisant des agents-réactions semble donner des résultats cohérents avec les données de la médecine, mais il reste certainement perfectible. Les résultats obtenus *in virtuo* doivent être confrontés avec de véritables expérimentations *in vitro*. Rappelons que l'intérêt principal de notre approche par rapport à un modèle classique à base d'équations différentielles est qu'il est possible à tout moment de rajouter dans la simulation un traitement, et donc d'estimer le moment optimal de la prise d'un médicament.



Figures a et b, des résultats de traitements sont présentés. Là encore, les résultats sont cohérents : l'hémophilie B est soignée avec du NovoSeven (le patient virtuel coagule davantage) et la maladie de Leiden avec de l'héparine (le patient virtuel coagule moins et moins tôt).

Figure 2.38: Diverses simulations de coagulation : hémophilie B traitée avec du NovoSeven et maladie de Leiden traitée avec de l'héparine.

2.6.2.3 Intérêt de la modélisation et de la simulation pour l'hématologie

Ces expérimentations *in virtuo* de la coagulation montrent bien l'intérêt de la simulation pour l'hématologue : la complexité de la cascade de la coagulation est telle qu'il est difficile de prévoir les changements impliqués par une ou plusieurs modifications dans le système de l'hémostase. Il devient ainsi possible de tester *in virtuo* l'action d'un médicament et de déterminer le moment le plus adapté pour la prise de ce médicament. L'interaction entre le biologiste et le modèle simulé devient alors primordiale [143, 145].

2.6.3 Application en cancérologie : le cas du myélome multiple

Le myélome multiple ou maladie de Kahler est un cancer rare, responsable en France de 1% de mortalité par cancer, avec une incidence de 2 cas pour 100 000 habitants et par an. Il est lié à la prolifération monoclonale de plasmocytes malins. La majorité des données épidémiologiques relatent l'importance des facteurs environnementaux. Les éléments chimiques, notamment ceux présents dans les pesticides utilisés en agriculture, les dioxines et le benzène semblent être des éléments déclencheurs. Certains virus pourraient également être déclencheurs ou cofacteurs. De plus, certains auteurs ont mis en évidence des prédispositions génétiques.

Nous faisons tout d'abord le point sur différents modèles biologiques proposés pour le myélome. Puis, nous en proposons une modélisation informatique basée sur la notion d'agents-cellules. Le comportement de chaque agent est modélisé par un Graphe d'Influences Flou (GIF) qui, à l'image des voies d'activation des cellules, permet de décrire son comportement en fonction de stimuli externes (voir la section 2.3.4.1).

Notons que ce travail fait partie d'une thèse de biologie dont la soutenance est prévue fin 2004 [119] et qu'il est le fruit d'une collaboration avec l'équipe de INSERM U463 de Nantes dirigée par le Professeur François-Régis Bataille. Cette équipe s'occupe plus particulièrement de l'oncogénèse du myélome multiple et des lymphopathies B avec une caractérisation du clone tumoral et des interactions avec son environnement.

2.6.3.1 Un modèle biologique du myélome

Nous présentons ici un modèle cellulaire de croissance et de différenciation myélomateuse basé sur l'expression membranaire du CD45.

a) Le CD45 lors de la différenciation plasmocytaire normale

Le CD45 est une tyrosine phosphatase importante dans l'activation et le développement plasmocytaire. Lors de la différenciation plasmocytaire normale, il a été mis en évidence que la prolifération est associée à une forte expression de CD45 sur la surface de la cellule. La différenciation est, elle, corrélée à la diminution de CD45 allant parfois, mais rarement, jusqu'à la perte de CD45. L'évaluation *in vivo* de l'expression de CD45 plasmocytaire montre que les plasmocytes immatures présents dans les amygdales et le sang expriment fortement CD45 alors que l'étude phénotypique du CD45 plasmocytaire montre une hétérogénéité dans la moelle osseuse. En effet, dans la moelle osseuse les plasmocytes peuvent perdre l'expression de CD45. L'ensemble de ces données semble indiquer une association entre l'expression de CD45 et la prolifération cellulaire. Bataille distingue plusieurs types de cellules exprimant le CD45 qui, selon les cas, influent sur le pronostic de la maladie (voir la figure 2.39). Les cellules CD45 Bright expriment fortement CD45, les cellules CD45 Low expriment peu CD45 et les cellules CD45 Neg n'expriment pas CD45.

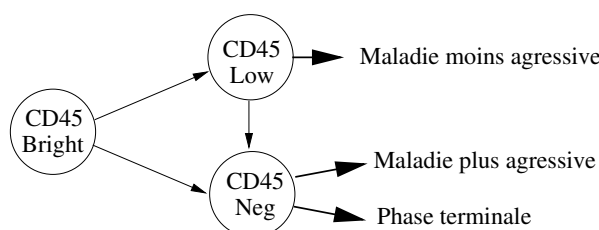


Figure 2.39: Evolution de la maladie en fonction de l'expression de CD45.

b) L'Il6, facteur de croissance des cellules myélomateuses

De plus, *in vitro*, l'Il6 cible principalement les cellules hautement prolifératives CD45 Bright pour supporter leur survie et leur croissance. *In vivo*, l'Il6 assure la survie des cellules CD45 Bright dans la circulation sanguine. Il a été montré aussi que l'Il6 a potentiellement une activité paracrine¹ de survie et/ou de croissance des

¹ Paracrine : terme employé pour désigner l'action de molécules sur les cellules environnantes.

cellules myélomateuses. De plus, il a été mis en évidence que l'Il6 stimule l'expression de CD45 [71]. La diminution du CD45 est parallèle à celle des récepteurs à Il6 jusqu'à ce que les cellules myélomateuses échappent à l'effet de l'Il6. Ces données sont renforcées par le fait que seules les cellules CD45 Bright prolifèrent en réponse à l'Il6. Ceci serait lié à l'activation par la phosphatase des SRC Kinases, essentiellement *Lyn* [72]. Les signaux suivant l'activation de *Lyn* restent à étudier.

c) La disparition de CD45 est associée à un *signalling* Insuline/IGF-1

Le CD45 est un inhibiteur du *signalling* Insuline/IGF-1 dans les cellules myélomateuses. Bien que l'expression de récepteurs à l'insuline soit équivalente dans les cellules myélomateuses immatures ou non, la phosphorylation de ces récepteurs² est multipliée par trois dans les cellules n'exprimant pas CD45 par rapport à celles l'exprimant. De même l'activité de la voie PI3K/AKT est triplée dans les cellules n'exprimant pas CD45. Le CD45 semble donc jouer le rôle d'un "interrupteur" qui va permettre à la cellule de choisir son "carburant" pour proliférer (voir la figure 2.40).

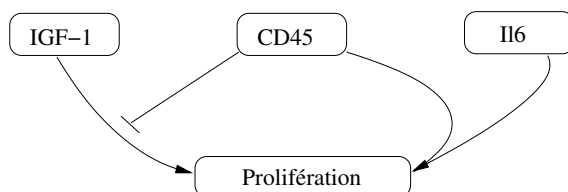


Figure 2.40: Rôle du CD45 dans les myélomes.

d) Le rôle du TNF dans les myélomes

De plus, il semblerait que $TNF\alpha$ ait un impact sur la croissance paracrine des cellules myélomateuses et sur leur survie dans leur micro-environnement [65, 98]. D'après [65], l'effet de $TNF\alpha$ serait indirect. Il stimulerait la sécrétion d'Il6 par les cellules stromales³ de la moelle osseuse et provoquerait l'expression de molécules d'adhésion (LFA-1, ICAM-1...) sur les cellules myélomateuses et les cellules du stroma médullaire.

2.6.3.2 Un modèle informatique du myélome

A partir de ces observations biologiques, nous proposons ici un modèle individu-centré qui s'intéresse tout d'abord au comportement de la cellule plutôt qu'à celui de la population. Pour donner un comportement aux cellules myélomateuses, nous proposons donc d'utiliser un modèle dérivé de celui proposé par Bataille.

² Phosphorylation de récepteurs : fixation de phosphore sur les récepteurs d'une cellule entraînant une activation ou désactivation de ces récepteurs et donc, une modification de la fonction de la cellule.

³ Cellules stromales : cellules composant le stroma (tissu conjonctif constituant la charpente d'une tumeur cancéreuse).

a) Voie de transmission d'Il6 et IGF-1

Les valeurs nécessaires pour la modélisation quantitative des voies de transmission sont difficilement accessibles dans les cas qui nous intéressent. Pour cela nous avons développé des interactions chimiques utilisant les Graphes d'Influences Flous (GIF), appelés aussi cartes cognitives floues [114], afin de pouvoir simuler le comportement de ces voies. Ces interactions nous permettent de donner un comportement à nos cellules en fonction de leur micro-environnement. Nous avons ainsi modélisé des cellules myélomateuses exprimant les récepteurs IGF-1, CD45 et Il6 (voir la figure 2.41).

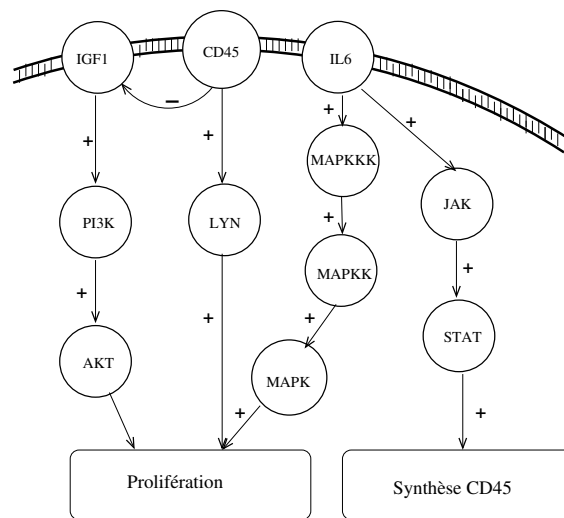


Figure 2.41: Modèle de cellule myélomateuse : GIF pour le *signalling* IGF-1/Il6.

Une cellule décide de son comportement en réponse aux récepteurs (capteurs) qu'elle possède. Elle peut exprimer de nouveaux récepteurs, sécréter des molécules ou effectuer un travail. Les GIF que nous utilisons possèdent des concepts perceptifs et des concepts moteurs. La décision de l'agent associée au GIF est obtenue par l'activation et la dynamique du GIF, les concepts perceptifs étant activés par *fuzzyfication* des récepteurs, les concepts moteurs activant des effecteurs par *défuzzyfication* (voir la figure 2.42).

b) Modélisation du comportement de CD45

Les GIF nous permettent de modéliser le comportement de cellules exprimant plus ou moins le CD45. Dans notre modélisation, la synthèse de CD45 étant liée aux signaux issus du *signalling* d'Il6, la sortie du GIF qui décrit le *signalling* des facteurs de croissance nous permet de contrôler la synthèse de CD45. Bataille a mis en évidence que la maturation des plasmocytes tumoraux est associée à une diminution de la détection de CD45. Nous avons donc fait l'hypothèse qu'ils se dimérisent, ce qui a pour conséquence de stopper l'activité des CD45. Pour cela nous avons créé deux agent-réactions autonomes agissant sur l'environnement des cellules, l'un qui synthétise les CD45 et l'autre qui dimérise les CD45 (voir la figure 2.43). L'environnement dans

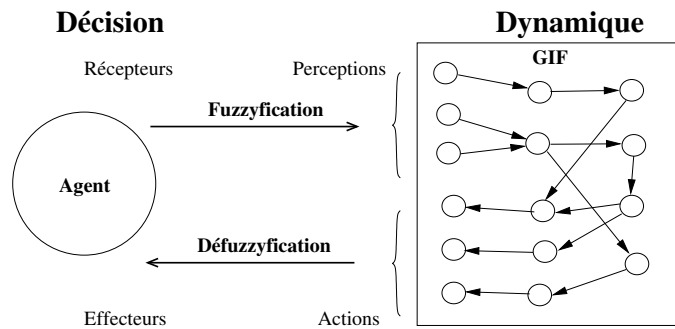


Figure 2.42: Connexion d'un agent à un GIF.

lequel sont plongées les cellules voit donc se modifier les concentrations de CD45 et de son dimère⁴.

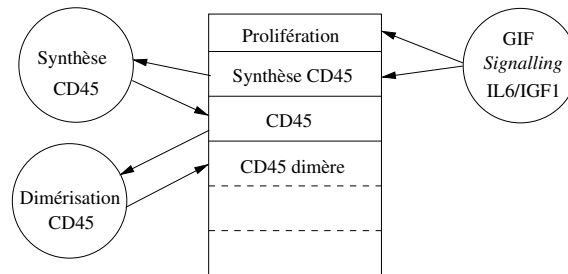


Figure 2.43: Modèle de cellule myélomateuse : dimérisation.

c) Evolution de la tumeur

La figure 2.44 montre l'évolution de la tumeur myélomateuse en fonction du modèle développé par Bataille. Elle met en jeu deux "carburants" différents, l'Il6 et l'IGF-1. Le premier agit, principalement, grâce à la voie des MAPK et JAK-STAT et stimule l'expression du phénotype⁵ CD45. Ce phénotype se comporte donc comme un "interrupteur" qui va permettre à la cellule de choisir son "carburant". Ainsi, quand la cellule s'éloigne des sources d'Il6 synthétisées au niveau des ostéoclastes, moins d'Il6 est alors disponible. La cellule va, de ce fait, utiliser l'IGF-1 pour stimuler sa croissance. De plus, la perte d'expression de CD45 observée à la surface des cellules, peut s'expliquer par une perturbation de la balance synthèse/dimérisation des CD45. Il a été récemment mis en évidence que la cinétique de cette homodimérisation est liée à l'isoforme⁶ de CD45 observée dans les myélomes. Ces isoformes pourraient donc être un facteur pronostique non négligeable en ce qui concerne la survie du patient, car la perte de CD45 est un des signes des phases les plus agressives de la maladie.

⁴ Dimère : molécule constituée à partir de deux molécules simples.

⁵ Phénotype : caractère ou élément produit par le génotype (génotype : patrimoine génétique d'un individu dépendant des gènes hérités de ses parents) en fonction de circonstances particulières.

⁶ Isoforme : une des formes de protéine qu'un gène peut produire.

La figure 2.45 présente l'évolution des différentes populations de CD45 : Bright qui prolifèrent peu, Low qui prolifèrent plus et Neg qui prolifèrent beaucoup plus et de façon exponentielle. Au début, seules les cellules exprimant CD45 Bright prolifèrent, c'est la phase initiale de la maladie. Puis, lorsque la maladie s'aggrave, celles exprimant CD45 Low commencent à proliférer, c'est la phase transitoire de la maladie. Enfin, les cellules exprimant CD45 Neg prolifèrent exponentiellement, c'est la phase terminale de la maladie.

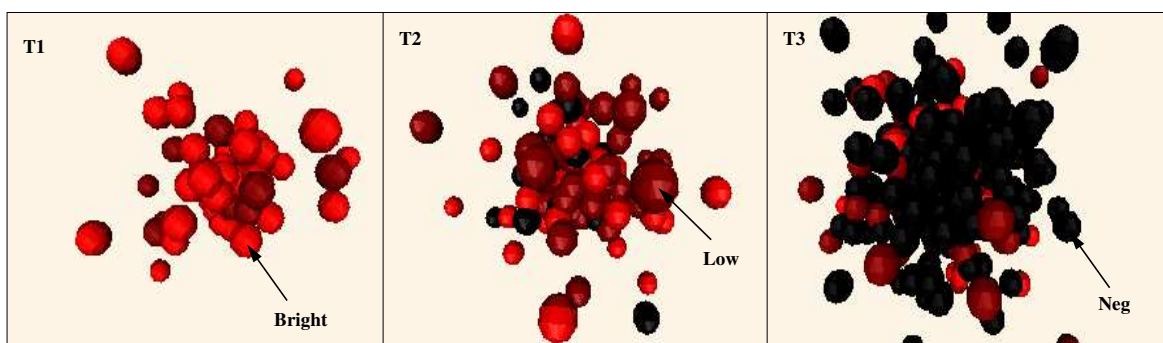


Figure 2.44: Evolution de la tumeur.

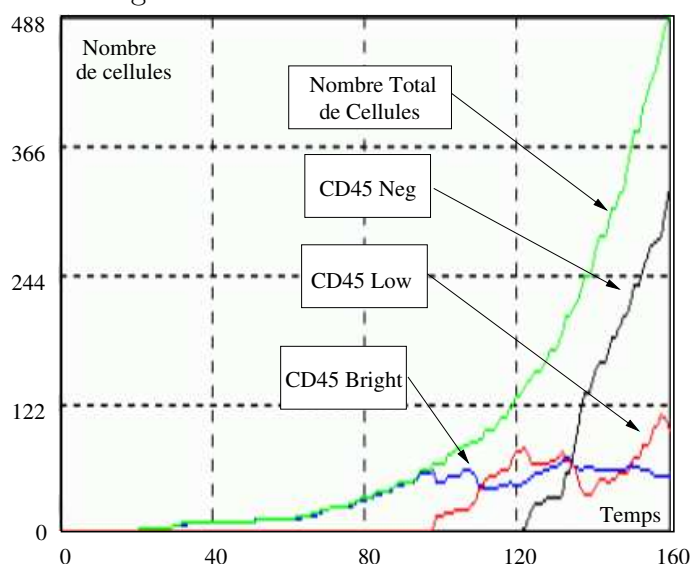


Figure 2.45: Evolution des populations de cellules exprimant plus ou moins CD45. Cette figure présente différentes courbes : le nombre total de cellules myélomateuses, le nombre de cellules exprimant fortement CD45 (CD45 Bright), le nombre de cellules exprimant faiblement CD45 (CD45 Low) et le nombre de cellules n'exprimant pas CD45 (CD45 Neg).

Les résultats obtenus *in virtuo* semblent cohérents avec les observations cliniques des cancérologues. L'expérimentation *in virtuo* a permis de pré-tester facilement des hypothèses qui doivent par la suite être vérifiées *in vivo* ou *in vitro*. Cette démarche de recherche peut être vue comme un outil de réflexion permettant de restreindre le nombre de voies à étudier et donc, probablement, de gagner un temps précieux dans le domaine de la cancérologie.

2.7 Conclusion sur la modélisation et la simulation de systèmes biologiques

Dans ce chapitre, nous avons vu que la modélisation et la simulation de phénomènes biologiques à l'aide de systèmes multi-agents sont intéressantes pour plusieurs raisons.

Tout d'abord, l'analogie entre un agent et une cellule ou une molécule se fait naturellement. Rappelons que, si le nombre d'entités est trop important, il est possible de ne plus considérer les cellules et les molécules comme éléments de base. Nous pouvons, en effet, ne plus prendre en compte les entités elles-mêmes, mais plutôt leurs concentrations dans le milieu, milieu qui peut être discrétisé et dont les subdivisions peuvent échanger des flux d'information ou de matière. Dans le cas où nous considérons les concentrations des entités, celles-ci interagissent grâce à des agents-réactions modélisant des réactions biochimiques.

Ensuite, un autre point très intéressant de l'utilisation des systèmes multi-agents pour la modélisation en biologie est leur modularité et leur extensibilité. Les systèmes multi-agents permettent en effet d'intégrer très simplement de nouvelles connaissances biologiques dans un modèle sans avoir à repenser entièrement ce modèle.

Pour finir, une autre raison qui nous semble peut-être encore plus fondamentale est l'interaction possible entre le biologiste et le modèle multi-agents en cours de simulation. Nous pouvons alors parler d'expérimentation *in virtuo*, c'est-à-dire que le biologiste peut, à tout moment, ajouter ou enlever une cellule ou une molécule, modifier le comportement d'une cellule en agissant, par exemple, sur le réseau de signalisation intracellulaire. Grâce à l'expérimentation *in virtuo*, le biologiste peut également imaginer des éléments de son modèle et les pré-tester à moindre coût.

Ainsi, la simulation peut intervenir dans l'étude préliminaire de fabrication d'un médicament. En effet, selon le type de maladie, il est possible de tester *in virtuo* divers médicaments virtuels. Le pharmacologue, désirant trouver à la fois le levier moléculaire sur lequel agir pour compenser une insuffisance due à une maladie et estimer la dose de médicament à injecter, pourra être fortement aidé par la simulation. Il lui sera également possible de déterminer le moment d'administration du médicament pour optimiser le traitement. Le chercheur en biologie peut, quant à lui, utiliser l'expérimentation *in virtuo* pour construire son modèle de façon incrémentale et l'évaluer sur ordinateur, tout en interagissant avec lui. Le biologiste a désormais à sa disposition un formidable outil de réflexion lui permettant de faire des hypothèses et de les pré-tester, réduisant ainsi le nombre de pistes à étudier lors de ces expérimentations *in vivo* ou *in vitro*.

Le chapitre suivant est consacré à un tout autre domaine de la biologie. Il s'agit de l'acquisition de données biologiques par traitement d'images. Nous utilisons pour cela une modélisation à base de systèmes multi-agents composés d'agents autonomes cherchant des éléments caractéristiques dans les images. Suite à nos travaux en immunologie et à la généralité des concepts mis en œuvre, nous avons eu l'idée d'utiliser certains de ces concepts pour réguler une population d'agents afin d'obtenir un arrêt automatique du traitement sans avoir à utiliser d'observateur global.

Chapitre 3

Système multi-agents immunitaire auto-régulé pour le traitement d'images biologiques

Dans ce chapitre, nous présentons un autre type d'utilisation des systèmes multi-agents en biologie. Plus précisément, nous détaillons une méthode de segmentation d'images biologiques utilisant un système multi-agents. Inspirée de la programmation par essaim [40], cette méthode utilise des agents aux comportements très simples leur permettant de prendre des décisions dans le but de détecter des éléments caractéristiques dans les images. Afin de spécifier le comportement de nos agents, nous utilisons le langage oRis qui permet de gérer et de décrire simplement le comportement d'entités autonomes. Nous analysons l'influence des différents paramètres de nos agents et présentons deux applications de traitement d'images biologiques.

Notons que l'intérêt de ce chapitre ne s'arrête pas à la simple utilisation de systèmes multi-agents pour le traitement d'images biologiques. En effet, nous abordons un problème purement informatique lié à l'arrêt automatique de systèmes multi-agents. Nous proposons une solution à ce problème en utilisant une métaphore biologique empruntée au système immunitaire. Plus exactement, nous proposons de tirer parti de phénomènes immunitaires pour auto-réguler une population d'agents. Cette régulation peut se faire en intégrant le comportement de cellules immunitaires à celui des agents.

Signalons enfin que ce chapitre correspond à la traduction "étendue" d'un article publié dans la revue *Pattern Recognition* sous le titre *An immune oriented multi-agent system for biological image processing* [126].

3.1 Introduction

Dans ce chapitre, nous décrivons un système distribué de traitement d'images reposant sur une plate-forme multi-agents. Notre système est composé d'agents réactifs situés.

La réactivité de nos agents étant un point crucial, leurs comportements doivent être très simples (perception-action). Malgré cette simplicité, les agents ont la capacité de s'adapter localement à leur environnement, c'est-à-dire, l'image. Un agent peut se déplacer et a sa propre position dans son environnement.

Plus précisément, un agent peut accéder à une petite partie de son environnement (son environnement local). Il peut également accéder à une mémoire partagée contenant les éléments de l'image déjà détectés (contours, régions,...). Cette notion de mémoire partagée est très importante parce qu'elle permet l'interaction entre les agents et la coordination de leurs actions.

Afin de modéliser notre système multi-agents dédié à la détection d'informations dans les images, nous avons utilisé le formalisme BRIC [42] (Basic Representation of Interactive Components). La figure 3.1-a présente notre système composé de trois parties essentielles : les agents, l'environnement (l'image et la mémoire partagée) et l'ordonnanceur qui coordonne l'ensemble.

Des capteurs permettent à chaque agent de percevoir des informations (voir la figure 3.1-b). En fonction des données renvoyées par ses capteurs, de sa mémoire et de son état interne, l'agent prend des décisions et les applique à l'aide de ses actionneurs (Aller vers, Détection d'éléments). Les états internes d'un agent sont décrits par une machine à états.

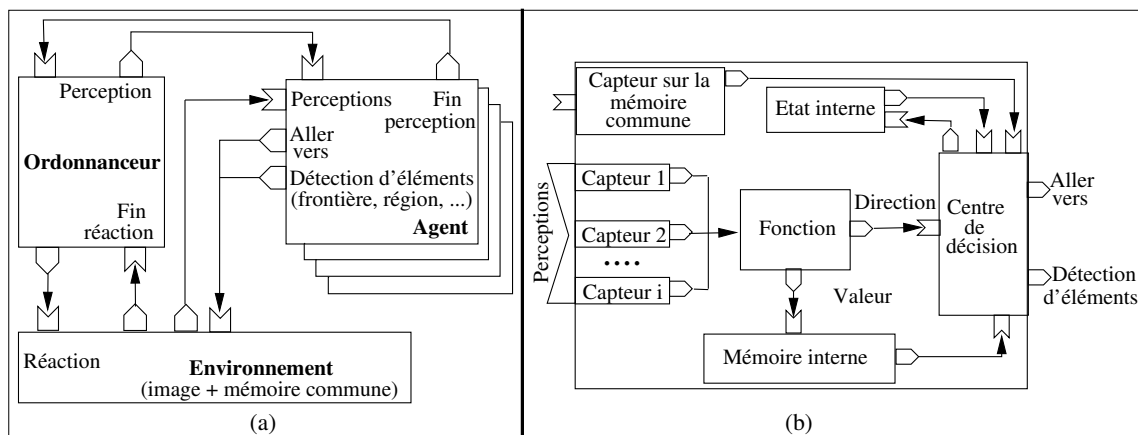


Figure 3.1: (a) Modèle BRIC de notre système multi-agents pour le traitement d'images, (b) Modèle BRIC de nos agents.

Dans notre système multi-agents, chaque agent a donc un comportement très simple qui lui permet de prendre une décision (trouver un contour, une région...) selon sa position dans l'image et l'information qui y est contenue. Ce comportement est très précisément et très simplement décrit en oRis [59, 60], un langage objet orienté agent qui est au centre de notre système distribué de traitement d'images.

Dans ce chapitre, nous présentons également une méthode de segmentation d'images utilisant un système multi-agents ainsi que deux applications biologiques réalisées avec oRis.

La première application vise à détecter des stries concentriques sur des “objets” biologiques (cernes des arbres, anneaux de croissance sur des otolithes de poissons [128],...). Dans cette application, nous utilisons des agents de deux types : des agents éclaircisseurs et des agents assombrisseurs. Ces agents suivent soit les stries claires (agents éclaircisseurs), soit les stries sombres (agents assombrisseurs). Ils ont également la possibilité d’agir sur les images. Leurs actions ont pour objectif le renforcement des contrastes permettant une détection fiable des stries même si celles-ci sont discontinues (voir la figure 3.2-a).

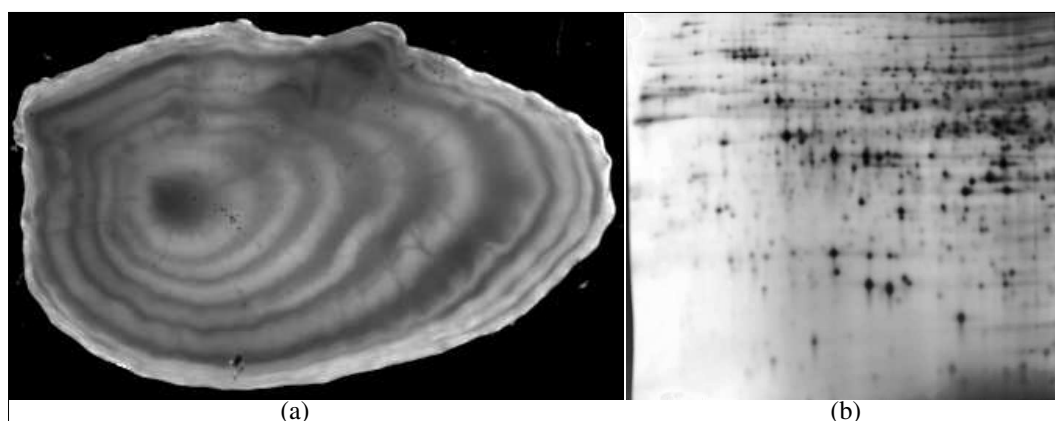


Figure 3.2: (a) Un otolithe de plie de 8 ans – (b) Un gel d’électrophorèse 2D.

La seconde application concerne le traitement d’images de gels d’électrophorèse 2D. Généralement employée en protéomique, la technique des gels d’électrophorèse 2D permet l’isolement des protéines en vue de leur caractérisation par spectroscopie de masse (voir la figure 3.2-b). Dans cette application, nous avons développé des agents capables de détecter des spots dans des images même si ces spots sont saturés ou très peu contrastés. Ces agents sont très similaires aux agents utilisés dans la première application.

Dans ces deux applications, nous essayons d’obtenir des informations image qui peuvent être assimilées à des contours en toit. Ce chapitre présente une approche pour détecter ce type de contours à l’aide d’un système multi-agents.

Les contours sont généralement considérés comme des discontinuités de niveaux de gris dans une image, et ont la plupart du temps le nom de contours en échelon (ou marche). Néanmoins, il existe d’autres types de contours dans les images, comme des contours en forme de toit (ou crête) ou de rampe. John Canny [25] a défini une méthode permettant d’évaluer si un détecteur de contours est efficace ou non. Pour être efficace, un détecteur doit donner une bonne localisation des contours et un nombre faible de pixels mal détectés (pixels détectés ne correspondant pas à un contour ou pixels de contour non détectés). Djemel Ziou [157] a adapté les filtres optimaux définis par Canny pour la détection de contours de type rampe ou toit et les a étendus en 2D. Il a appliqué son filtre sur des images synthétiques comportant des lignes noyées dans un bruit gaussien. Le rapport signal sur bruit était de 7 dB sur une moitié de l’image et de 10 dB sur l’autre moitié. Les résultats montrent une sur-détection importante des contours.

Robert Haralick [58] associe les crêtes ou les vallées aux passages par zéro de la dérivée première directionnelle. Le voisinage d'un pixel est alors utilisé pour estimer une surface continue dont on peut calculer analytiquement les dérivées directionnelles. Ce calcul est réalisé à l'aide d'un polynôme cubique ayant comme variables x et y (ligne et colonne). Des masques sur le voisinage du pixel courant sont employés pour calculer les coefficients du polynôme. Pour identifier une crête ou une vallée, un système de coordonnées est généré dont l'origine est centrée sur le pixel courant. Ensuite toutes les dérivées secondes directionnelles sont calculées à l'origine, pour déterminer la direction α qui donne un extremum de la dérivée seconde directionnelle. Un passage par zéro dans cette direction α de la dérivée première directionnelle est alors recherché. Cette méthode a besoin d'une quantité importante de calculs, de nombreux seuils et d'un choix arbitraire des coefficients du polynôme.

Raymond Smith [140] propose une méthode pour détecter les squelettes de lignes se trouvant dans des images de caractères manuscrits. Cette méthode, qui peut être utilisée pour détecter des arêtes dans des images, nécessite une première phase de binarisation. Cette étape est simple dans le cas de caractères manuscrits, mais elle peut être risquée dans le cas des images d'otolithes de poissons. En effet, un seuil trop bas atténuerait le contraste entre les derniers anneaux qui sont très clairs et un seuil trop haut effacerait les premiers anneaux qui sont plus foncés.

Basées sur des systèmes multi-agents, d'autres méthodes ont été récemment développées pour détecter des éléments caractéristiques dans les images [20, 35, 13, 97, 95, 96]. Bien que les agents utilisés réalisent chacun des actions très simples, ils arrivent, grâce à un partage de leurs résultats, à accomplir des traitements complexes. Jiming Liu [95] a développé un système multi-agents pour détecter des régions homogènes dans les images et l'a appliqué dans le cas d'images de scanner de cerveau. Dans son système, chaque agent peut effectuer des tests et des calculs (variance et moyenne) sur les pixels situés dans un voisinage circulaire autour de lui. Si l'agent estime que son voisinage satisfait les conditions pour être une région, le pixel central sera marqué et de nouveaux agents seront créés dans son voisinage. Cette méthode est bien adaptée pour le traitement d'images de scanner de cerveau comportant des régions bien régulières (tumeurs, parties saines, ...). Alain Boucher [22] a également utilisé un système multi-agents pour segmenter des images cytologiques. Il a défini un type d'agent qui peut être adapté à la recherche de quatre différentes structures présentes dans les images cytologiques : le noyau au cœur de la cellule, les zones blanches entourant la cellule dues à la diffraction de la lumière, les pseudopodes qui sont des extensions cytoplasmiques des cellules vers le milieu, et le fond de l'image. Dans ce système, la discrimination entre les différents éléments est essentiellement basée sur les caractéristiques des régions.

Dans ce chapitre, nous commencerons par décrire notre environnement multi-agents nommé oRis (section 3.2). Puis, nous présenterons deux applications de traitement d'images biologiques développées avec oRis (section 3.3). Nous analyserons ensuite l'influence des paramètres de nos agents (section 3.4).

Un des problèmes les plus importants des systèmes multi-agents est leur arrêt

automatique. Afin de résoudre ce problème, nous avons implémenté l'arrêt de notre système multi-agents de traitement d'images à l'aide d'une technique issue de l'immunologie : l'apoptose (section 3.5).

3.2 Langage et simulateur oRis

3.2.1 Concepts principaux d'oRis

L'environnement oRis propose à la fois un langage orienté agent permettant de décrire la structure d'un système multi-agents et un simulateur pour, dynamiquement, simuler et modifier ce système [59, 60].

Nous employons le terme de système multi-agents car chaque composant peut être doté d'un comportement autonome qui lui donne la possibilité d'interagir et d'évoluer parmi les autres composants du système (représentant alors son environnement).

Le langage oRis reprend à la fois les concepts de programmation structurée et de la programmation orientée objet (la syntaxe d'oRis est très proche de celle du C++) auxquels nous avons ajouté des fonctionnalités orientées agent et des propriétés dynamiques.

Dans la suite de cette section, nous détaillons les spécificités du langage oRis.

3.2.1.1 Programmation structurée et programmation orientée objet

Le langage oRis permet la programmation structurée d'une manière très semblable au langage C/C++. L'écriture d'un programme, en programmation structurée, consiste donc à définir des fonctions et des classes.

Le code principal du programme structuré est représenté par un bloc d'instructions intitulé 'execute'. Ce bloc est assez semblable à la fonction 'main()' du langage C/C++ dans le sens où il représente le point d'entrée du programme.

3.2.1.2 Programmation et fonctionnalités orientées agent

Le langage oRis ajoute à la programmation orientée objet la notion d'objet actif et permet ainsi d'accéder à la programmation orientée agent.

Le principal point apporté est l'autonomie des instances. Ceci signifie que celles-ci

peuvent être dotées d'un comportement autonome qu'elles exécutent perpétuellement dès lors qu'elles sont créées; ces instances sont alors qualifiées d'agents.

Ecrire une application orientée agent, c'est donc instancier des agents. Le bloc `'execute'` ne sert donc qu'à initialiser l'application en créant les instances; ensuite ce sont les instances qui font évoluer l'application et non un programme principal. Ainsi, la fin du bloc `'execute'` ne représente plus du tout la fin de l'application et c'est là une grande différence avec la fonction principale `'main()'` des langages C/C++.

Dans oRis, la distinction objet/agent se fait grâce à la méthode `'main()'` de l'instance. Une instance dotée de cette méthode est qualifiée d'agent sinon, c'est un objet, ce qui implique que l'on est obligé d'invoquer explicitement ses méthodes pour qu'elle fasse quelque chose.

En fait, oRis est plus qu'un langage, c'est aussi un simulateur chargé d'animer tous les agents. C'est pour cette raison que l'on parle usuellement de simulation pour désigner une application oRis. Le simulateur se charge d'appeler automatiquement et perpétuellement la méthode `'main()'` de chaque agent. En effet, celle-ci représente le point d'entrée du comportement de l'agent concerné.

La figure 3.3 présente un programme orienté agent très simple dans lequel figurent les différents points évoqués ainsi que l'affichage produit. Remarquons que le bloc `'execute'` ne comporte que des instanciations et que l'exécution proprement dite de l'application a lieu après la sortie de ce bloc et se poursuit indéfiniment.

3.2.1.3 Propriétés dynamiques d'oRis / granularité instance

Les agents ayant chacun leur autonomie et leur propre existence, il est parfois utile de spécifier des particularités de comportement au niveau d'une instance particulière et non pas de la classe toute entière. Ainsi oRis, grâce à sa granularité au niveau instance, permet d'obtenir un agent qui diffère des autres instances de sa classe. Ceci permet à un agent de redéfinir son propre comportement ou le comportement d'autres agents s'il estime que ce comportement est meilleur pour l'exécution d'une tâche donnée.

Nous entendons par propriétés dynamiques la possibilité de modifier ou de compléter le code d'une simulation en même temps que celle-ci s'exécute et ceci sans interrompre son déroulement. La totalité du langage est disponible lors de l'introduction de nouveau code. Il est ainsi possible à tout moment et sans arrêter la simulation de redéfinir des méthodes de classes existantes, de définir de nouvelles classes, de redéfinir des méthodes d'instances existantes, d'ajouter des méthodes à des instances, d'ajouter des attributs à des instances et d'exécuter des instructions quelconques. Le code introduit dynamiquement peut être composé par les agents eux mêmes. Il peut également être introduit à l'aide d'une fenêtre de dialogue permettant à un utilisateur d'agir sur le déroulement d'une simulation.

```

class Example          // Déclaration de la classe 'Example'
{
  int age;             // Un 'Example' contient un entier 'age'
  void main(void);    // Un 'Example' a une méthode 'main()'
};                    // ---> c'est un agent

void Example::main(void) // Définition de la méthode 'main()'
{
  println("The age of ", this , " is ", ++age);
}

execute                // Code à exécuter (initialisation)
{
  println("---- begin ----");
  for(int i=0; i<4; i++) new Example; // Instanciation
  println("---- end ----");          // de 4 'Example's
}

```

Le résultat de l'exécution de ce programme orienté agent est donné ci-après :

```

---- begin ----
---- end ----
The age of Example.2 is 1
The age of Example.3 is 1
The age of Example.4 is 1
The age of Example.1 is 1
The age of Example.1 is 2
The age of Example.3 is 2
The age of Example.2 is 2
The age of Example.4 is 2
...

```

Remarque : les agents sont choisis dans un ordre aléatoire.

Figure 3.3: Un très simple programme orienté agent en oRis.

Cette granularité de description au niveau instance permet vraiment d'envisager des agents dont le comportement de base change au cours du temps.

La figure 3.4 présente deux exemples de redéfinitions de méthode au niveau instance. Cet exemple suppose que la simulation décrite figure 3.3 soit toujours en fonctionnement. Signalons que la deuxième redéfinition est réalisée par un agent qui modifie sa propre méthode 'main()'.

Ce concept de granularité au niveau instance sera très utile par la suite lorsque les agents utilisés pour trouver des informations dans les images pourront se différencier les uns des autres (voir la section 3.5).

3.2.2 Communication entre agents

L'intérêt d'une architecture multi-agents réside dans la collaboration (directe ou non) entre les agents. De ce fait, ceux-ci disposent de plusieurs moyens de communication : synchrone (appels directs de méthodes), asynchrone (messages point à point – boîtes à messages –) et diffusion (émission d'un message dans l'univers des agents).

Nous détaillons ci-après ces différents moyens de communication entres agents.

```

void Example.2::main(void) // ==> Redéfinition de la méthode
{
    // 'main()' de l'instance Example.2
    println("First redefinition :");
    println("=> ", this , " : ", ++age , " cycles");

    string secondProgram = "";
    secondProgram = "void Example.2::main(void)";
    secondProgram += "{";
    secondProgram += " println(\"Second redefinition :\")";
    secondProgram += " println(\"==> \", this , \" : age=\", ++age);";
    secondProgram += "}";
    parse(secondProgram);
}

```

Après avoir introduit ce code dans la simulation, le résultat devient comme suit :

```

The age of Example.4 is 3
The age of Example.3 is 3
First redefinition :
=> Example.2 : 3 cycles
The age of Example.1 is 3
The age of Example.3 is 4
The age of Example.4 is 4
The age of Example.1 is 4

```

*La sortie écran
comme ici*

```

Second redefinition :
==> Example.2 : age=4
The age of Example.1 is 5
Second redefinition :
==> Example.2 : age=5
The age of Example.4 is 5
The age of Example.3 is 5
...

```

Figure 3.4: Deux exemples de redéfinition de méthode au niveau instance.

3.2.2.1 Communication synchrone

La communication synchrone correspond à de simples appels de méthodes comme en C++. Dans ce cas, l'exécution du traitement appelant est interrompue jusqu'à ce que le traitement appelé soit terminé. Elle se poursuit ensuite. Ce type de communication est tout à fait adapté à des objets passifs ou à des objets actifs de faible autonomie.

3.2.2.2 Communication asynchrone

Nous venons de voir que les actions élémentaires en oRis peuvent avoir lieu par de simples appels de méthodes. En revanche, une communication de plus haut niveau nécessite que les objets (ou agents) aient la possibilité de s'échanger des messages de manière asynchrone (avec gestion d'une boîte à messages) afin d'augmenter la souplesse et la fiabilité des demandes de services, des propositions, des négociations, etc.

En oRis, la communication asynchrone consiste à placer un message dans la boîte à messages d'un destinataire afin que celui-ci le traite lorsqu'il le souhaite et comme il l'entend. Ceci permet la mise au point d'agents ayant des comportements "évolués" ce qui élargit le champ d'application.

3.2.2.3 Communication par diffusion

Pour développer des agents ayant des comportements "évolués", il est nécessaire de disposer d'un moyen de propager et de capter de l'information dans un environnement.

Le mécanisme de diffusion semble bien adapté à ce genre d'exigence. En effet, il permet à l'émetteur d'envoyer des informations dans l'environnement sans se soucier des objets concernés, et, pareillement, il permet à ces mêmes objets d'être à l'écoute d'informations sans se soucier des circonstances de l'émission. Il suffit pour cela que les différents objets puissent émettre et recevoir le même type d'informations.

Ce point est très intéressant puisqu'il permet à un agent d'avertir les autres agents que, par exemple, une région précise de l'image n'a pas encore été traitée, ou qu'il vaut mieux employer un seuil donné pour le traitement de l'image.

3.2.2.4 Communication répartie

Lorsque le nombre d'agents est important, il peut être intéressant de répartir les agents sur plusieurs ordinateurs. Cet équilibrage de charge implique des communications entre agents distants à travers un réseau informatique [120, 127]. Nous avons implémenté cette communication entre agents distants à l'aide de mécanismes présents dans la norme HLA ou *High Level Architecture* [36]. Nous avons également proposé une adaptation de l'infrastructure HLA pour l'utilisation, à distance, des propriétés dynamiques d'oRis [120, 121].

3.2.3 Simulateur oRis

Bien qu'étant un langage, oRis est aussi un simulateur chargé d'animer les agents dont le comportement est décrit par le langage.

Une simulation multi-agents nécessite l'exécution parallèle de plusieurs agents. Nous devons alors nous assurer que le procédé d'activation de ces agents autonomes n'engendre pas de biais dans la simulation.

Le contrôle de l'ordonnancement des agents est donc un point crucial. L'expérience montre qu'en règle générale, peu de choses sont connues ou garanties par les services de multi-tâches fournis dans différents environnements de programmation. Ce manque d'information laisse toujours planer un certain doute concernant l'équitabilité entre les différentes tâches et les biais que cela pourrait introduire. Ici, nous présentons la procédure d'ordonnancement utilisée dans oRis afin que le programmeur sache exactement ce qu'il peut attendre lorsqu'il décide de développer des agents s'exécutant en parallèle.

3.2.3.1 Flots d'exécution

Le problème principal à aborder lorsque l'on parle de multi-tâches concerne la détermination de la nature des tâches que l'on souhaite exécuter en parallèle. Bien qu'elles soient gérées en interne de la même manière, le langage oRis fournit deux manières d'exprimer ces tâches que nous appellerons flots d'exécution.

En oRis, un agent a une méthode 'main()' qui représente le point d'entrée de son comportement. Dès lors qu'un agent est créé, sa méthode 'main()' est immédiatement prête à s'exécuter. Lorsque la fin de la méthode est atteinte, elle est automatiquement relancée. C'est donc un moyen simple d'implémenter le comportement autonome d'un agent. Une simulation multi-agents en oRis consiste à instancier de tels agents et à les laisser "vivre".

Un autre moyen pour initier un nouveau traitement parallèle consiste à dédoubler le flot d'exécution courant à l'aide de la primitive `start` (voir la figure 3.5). Ce procédé, qui génère plusieurs flots d'exécution à partir d'un seul, est principalement utilisé pour attribuer plusieurs activités à un même agent. Il semble en effet raisonnable qu'un agent puisse, par exemple, se déplacer tout en communiquant avec d'autres agents.

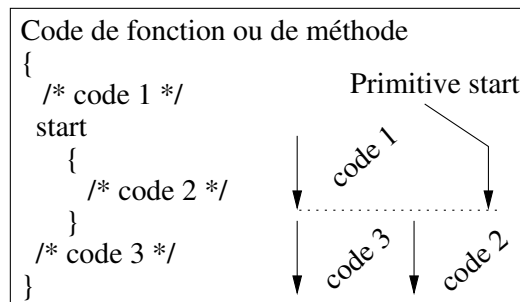


Figure 3.5: Dédoublage du flot d'exécution en oRis.

3.2.3.2 Ordonnanceur d'oRis

L'ordonnanceur d'oRis maintient plusieurs flots d'exécution. La figure 3.6 montre les structures de données utilisées pour contrôler ces processus parallèles. Chaque flot d'exécution est représenté par une structure de données qui, en plus de l'identifiant de l'activité, contient une pile de contextes et une pile de valeurs temporaires. La pile de contextes est utilisée pour gérer les appels de fonctions et de méthodes. Les modules exécutables sont représentés par une séquence de micro-instructions. Celles-ci étant atomiques, les commutations entre les différents flots ne peuvent avoir lieu qu'entre l'exécution de deux micro-instructions. La pile de valeurs temporaires est partagée par tous les contextes du flot d'exécution et permet notamment d'empiler les paramètres d'un module exécutable avant son appel, et de récupérer le résultat empilé lors de la terminaison du module appelé.

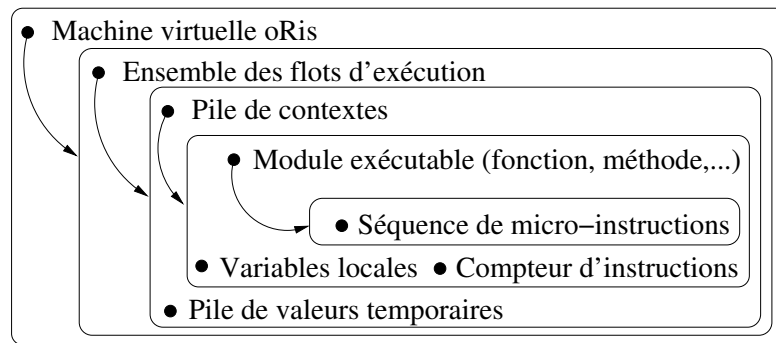


Figure 3.6: Structure de la machine virtuelle oRis.

Nous savons donc que différents flots d'exécution peuvent être interrompus entre deux de leurs micro-instructions. Nous allons maintenant expliquer dans quels cas un flot d'exécution peut s'interrompre au profit d'un autre.

Ceci mène à deux types de traitement multi-tâches (coopératif et préemptif) que le programmeur peut choisir librement dans son programme.

Une possibilité est de placer l'ordonnanceur dans le mode coopératif. Dans ces conditions, les activités ne peuvent pas changer spontanément. Un changement repose sur l'appel explicite de la fonction `'yield()'`, sur la fin d'une méthode `'main()'` ou sur l'attente d'une ressource. L'ordonnanceur peut alors interrompre le traitement courant afin de continuer l'exécution d'une autre activité.

Il est également possible de choisir un mode d'activation préemptif. Le programmeur doit alors indiquer à l'ordonnanceur le nombre de millisecondes après lequel un changement d'activité doit avoir lieu. Dans ces conditions, de longs comportements peuvent être écrits sans que le programmeur ait à placer de commutations explicites (`'yield()'`).

Afin de s'assurer que le temps est également partagé entre les différentes activités, nous introduisons la notion de cycle d'exécution qui apporte au système la propriété suivante : chaque flot d'exécution ne prend la main qu'une seule fois par cycle. De plus, si au cours d'un cycle, de nouvelles activités apparaissent, celles-ci ne commenceront à s'exécuter que lors du cycle suivant. Remarquons également que, pendant un cycle, nous proposons deux modes d'activation : un ordre fixe et un ordre aléatoire.

3.2.3.3 Couplage oRis/C++

Le simulateur oRis étant écrit en C++, il existe un couplage fort entre oRis et C++. Le programmeur peut très simplement faire le lien entre une classe oRis et une classe C++. Dans ce cas, l'appel à une méthode native oRis engendre un appel à la méthode associée en C++. Il est ainsi possible d'augmenter les performances du système en

réalisant des méthodes natives plus rapides que des méthodes équivalentes définies purement en oRis.

Toujours grâce au couplage entre oRis et C++, nous avons réalisé un ensemble de classes de manipulation et de visualisation d'images qui permettent de développer facilement des applications de traitement d'images.

3.3 Applications multi-agents de traitement d'images

Dans cette section, nous présentons deux applications développées avec oRis.

La première application vise à détecter des stries concentriques sur des “objets” biologiques (cernes des arbres, anneaux de croissance sur des otolithes de poissons [128], ...) même si ces stries sont discontinues.

La seconde application concerne le traitement d'images de gels d'électrophorèse 2D. Dans cette application, nous avons développé des agents capables de détecter des spots dans des images même si ces spots sont saturés ou très peu contrastés. Ces agents sont très similaires aux agents utilisés dans la première application.

3.3.1 Système multi-agents pour la détection de stries concentriques

Un des principaux problèmes rencontrés lors de la détection de stries concentriques sur des images biologiques (cernes des arbres, anneaux de croissance sur des otolithes de poissons, ...) est le manque de continuité de ces stries. Nous proposons une approche basée sur un système multi-agents pour résoudre ce problème. Chaque agent peut se déplacer dans son environnement qui est une image présentant une alternance d'anneaux concentriques clairs et sombres.

Dans cette section, nous présentons plus particulièrement une application concernant la détection d'anneaux de croissance sur des otolithes de poissons (voir la figure 3.2-a).

Les otolithes sont de minuscules concrétions calcaires situées dans la boîte crânienne des poissons, au niveau de l'oreille interne, et servant à leur audition et à leur équilibre. En raison de l'apparition alternée d'anneaux opaques (anneaux d'hiver) et d'anneaux translucides (anneaux d'été) pendant sa croissance accrétionnaire, le grossissement d'un otolithe à l'aide d'un microscope en lumière transmise fait apparaître une alternance de stries concentriques claires et sombres, dont le centre

est appelé *nucleus*. Ainsi le profil de telles images du *nucleus* vers le bord de l'otolithe présente une alternance de crêtes et de vallées, qui peuvent être assimilées à des contours en toit. La segmentation automatique de ce type d'images par des techniques traditionnelles peut être rendue difficile à cause de variations de niveau gris qui peuvent apparaître à l'intérieur des anneaux de croissance de l'otolithe, créant ainsi des textures (voir la figure 3.7).

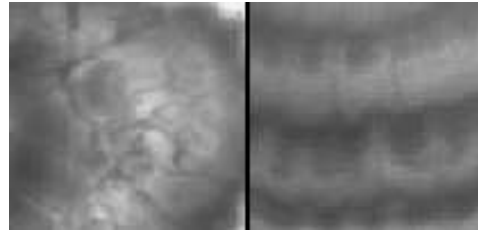


Figure 3.7: Exemples de textures présentes dans un anneau de croissance.

L'intérêt de l'identification des anneaux de croissance sur des images d'otolithes est l'acquisition de données sur l'âge et le développement de populations de poissons. De telles données sont nécessaires dans un grand nombre d'études biologiques et écologiques afin d'améliorer la gestion des stocks halieutiques. Jusqu'à présent, cette analyse était limitée à un simple comptage d'anneaux. Assurée par des opérateurs humains, cette analyse est une opération à la fois laborieuse et coûteuse en temps pour les biologistes et d'une fiabilité limitée puisque souvent liée à la subjectivité de l'observateur. L'opérateur humain base son observation essentiellement sur la perception de la continuité des anneaux de croissance. Cette tâche est délicate. Il est fréquent, en effet, de rencontrer des anomalies qui gênent l'interprétation, telles que des zones de resserrement local d'anneaux, des faux anneaux, des anneaux doubles ou interrompus, incomplets ou semi-opaques.

Notre système multi-agents dédié à la détection d'anneaux est composé d'un ensemble d'agents de deux types nommés agents éclaircisseurs et agents assombrisseurs. Ces agents suivent soit les anneaux clairs (agents éclaircisseurs), soit les anneaux sombres (agents assombrisseurs) et agissent sur l'image. Leurs actions ont pour objectif le renforcement des contrastes permettant une détection fiable des anneaux même si ceux-ci sont discontinus.

Afin de modéliser notre système multi-agents, nous utilisons le formalisme BRIC (voir la figure 3.8-a). A l'aide de capteurs, chaque agent peut obtenir des informations sur son environnement local (son voisinage dans l'image). Ces informations lui permettent, en fonction de son état interne, de prendre une décision (avancer, détecter un contour, ...). Le comportement complet d'un agent sera décrit plus loin dans cette section. Une modélisation BRIC d'un agent est présentée figure 3.8-b.

Un agent a trois actionneurs. Le premier lui permet de modifier son environnement en augmentant ou en diminuant l'intensité d'un pixel. Les mouvements de l'agent (rotation et translation) sont gérés grâce aux deux autres actionneurs.

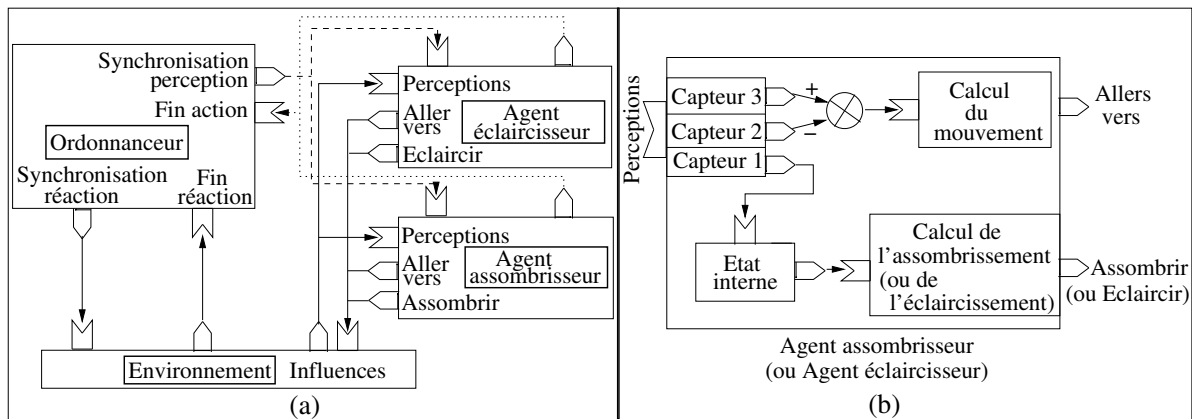


Figure 3.8: (a) Modèle BRIC de notre système multi-agents, (b) Modèle BRIC d'un agent.

Afin d'obtenir des informations sur son environnement, un agent dispose également de trois capteurs (voir la figure 3.9-a). Les capteurs de l'agent sont composés de capteurs unitaires retournant chacun la valeur d'un pixel. Plus précisément, les trois capteurs de l'agent sont:

- un capteur unitaire ($Capteur_1$) sur la mémoire commune permettant à l'agent de savoir s'il se trouve sur un anneau déjà détecté,
- deux capteurs carrés ($Capteur_2$ et $Capteur_3$) composés de capteurs unitaires sur l'image. Ces deux capteurs, situés devant l'agent et distants l'un de l'autre, retournent la moyenne de leurs capteurs unitaires.

Ces deux capteurs carrés servent à déterminer les mouvements de l'agent et par conséquent servent également à détecter les anneaux. Un agent assombrisseur essaye toujours de se diriger en déviant vers le capteur d'intensité minimale. Un agent éclaircisseur fait bien sûr l'inverse. Plus la différence Δ entre les deux capteurs est importante, plus l'agent dévie.

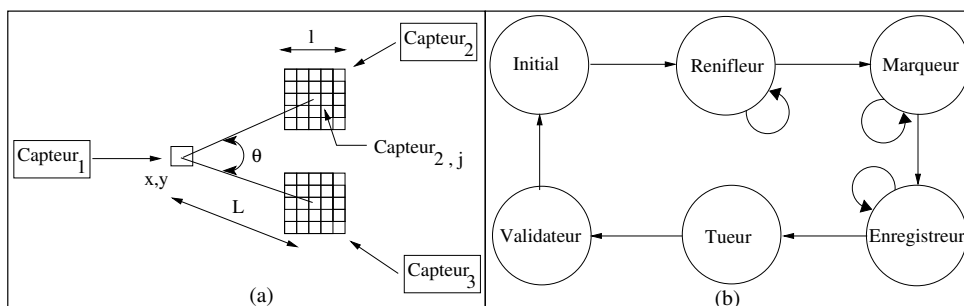


Figure 3.9: (a) Capteurs d'un agent, (b) Automate décrivant le comportement d'un agent.

Figure 3.9-a, nous pouvons observer :

- $(x, y) \in R^2$, position de l'agent dans l'environnement.
- $\theta \in [0, 2\pi]$, angle séparant les *Capteur*₂ et *Capteur*₃.
- $L \in N$, distance entre le *Capteur*₂ (*Capteur*₃) et le centre de l'agent.
- l , taille d'un côté d'un capteur carré.
- $i \in [2, 3] \cap N$, indice d'un capteur carré.
- $j \in [1, l^2] \cap N$, indice d'un capteur unitaire dans un capteur carré.
- $V_{i,j}$, valeur de niveau de gris renvoyée par *Capteur* _{i,j} , le capteur unitaire *capteur* _{j} du capteur carré *Capteur* _{i} .
- $V_i = \frac{\sum_{j=1}^{l^2} V_{i,j}}{l^2}$, valeur retournée par le capteur carré *Capteur* _{i} .
- $\Delta = V_2 - V_3$, différence entre les valeurs renvoyées par les deux capteurs carrés *Capteur*₂ et *Capteur*₃.

Section 3.4, nous étudierons l'influence de ces paramètres sur la qualité de la détection.

Les capteurs permettent à l'agent de recevoir des informations sur son environnement. En fonction des données reçues et de son état interne, un agent peut prendre des décisions et les appliquer via ses actionneurs. Les états internes d'un agent sont représentés par un automate (voir la figure 3.9-b). Les états et les transitions d'un agent sont décrits ci-après.

- Etat *Initialisation* : l'agent est placé aléatoirement sur l'image.
- Etat *Renifleur* : l'agent essaye de détecter un anneau (un anneau blanc dans le cas d'un agent éclaircisseur ou un anneau noir dans le cas d'un agent assombrisseur). L'agent passe dans l'état *Marqueur* s'il effectue un tour complet. Si l'agent atteint un âge maximum, il est tué et est remplacé aléatoirement sur l'image. Il retourne alors dans l'état *Renifleur*.
- Etat *Marqueur* : l'agent renforce le contraste de l'anneau sur lequel il se trouve. Si l'agent est un agent éclaircisseur, il augmente la valeur du pixel courant. Dans le cas d'un agent assombrisseur, celui-ci diminue la valeur du pixel courant. Dès que l'agent a effectué un tour complet, il passe dans l'état *Enregistreur*. Si l'agent atteint un âge maximum, il est tué et est remplacé aléatoirement sur l'image. Il retourne alors dans l'état *Renifleur*.
- Etat *Enregistreur* : l'agent mémorise son chemin jusqu'à avoir accompli un tour complet en étant revenu à son point de départ. L'agent passe alors dans l'état *Tueur*. Si l'agent atteint un âge maximum, il est tué et est remplacé aléatoirement sur l'image. Il retourne alors dans l'état *Renifleur*.
- Etat *Tueur* : l'agent mémorise son chemin dans la mémoire commune. De cette façon, l'agent indique aux autres agents l'emplacement de l'anneau qu'il vient de détecter. De plus, le chemin mémorisé est une sorte de poison pour les autres agents. Ceci évite aux différents agents de trouver plusieurs fois le même anneau. L'agent passe ensuite dans l'état *Validateur*.

- Etat *Valideur* : l'agent dessine sur l'image l'anneau qu'il vient de trouver. En fonction du type de l'agent, cet anneau est soit blanc (agent éclaircisseur), soit noir (agent assombrisseur). L'agent est finalement remplacé aléatoirement sur l'image et retourne dans l'état *Initialisation*.

Dans chaque état, si un agent se trouve sur du poison, il est tué et est remplacé aléatoirement sur l'image. Il retourne alors dans l'état *Renifleur*.

D'un coût de mise en œuvre relativement faible, ce système multi-agents n'en demeure pas moins efficace (voir la figure 3.10-a). Afin d'améliorer les résultats, nous proposons également d'utiliser des informations de "haut niveau" comme, par exemple, la forme du contour externe de l'otolithe [56, 57]. En évaluant son chemin par rapport à cette information de haut niveau, un agent peut savoir si son chemin est correct ou non. Ceci permet d'améliorer nettement les résultats (voir la figure 3.10-b). Figure 3.10, les agents utilisés ont les paramètres suivants : $L = 8$, $\theta = \pi/4$ et $l = 1$. L'influence de ces paramètres sur la qualité de la détection sera analysée plus loin dans ce chapitre. Avant cette analyse, étudions une autre application de traitement d'images biologiques. Cette application permet la détection de spots sur des images de gel d'électrophorèse 2D.

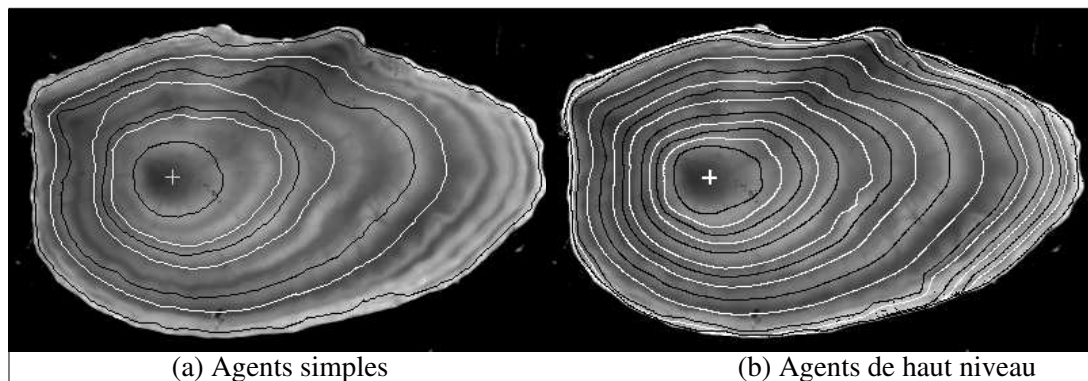


Figure 3.10: Exemples de résultats obtenus avec des agents simples et des agents de "haut niveau".

3.3.2 Système multi-agents pour la détection de spots

La technique des gels d'électrophorèse 2D est une méthode permettant, à partir d'un échantillon, la séparation et l'identification de protéines par un déplacement dans deux dimensions orientées perpendiculairement l'une par rapport à l'autre.

Généralement employée en protéomique, la technique des gels d'électrophorèse 2D permet l'isolement des protéines en vue de leur caractérisation par spectroscopie de masse. La séparation des protéines est tout d'abord réalisée dans une première

dimension par focalisation isoélectrique (IEF). La focalisation isoélectrique permet de séparer les protéines selon leur charge (pI). Puis, la séparation se fait dans une deuxième direction, perpendiculaire à la première, selon leur poids moléculaire. Cette deuxième séparation est réalisée avec la méthode SDS-PAGE (Sodium Dodecyl Sulphate - PolyAcrylamide Gel Electrophoresis). La figure 3.2-b présente une image d'électrophorèse 2D. Il existe un certain nombre de logiciels commerciaux permettant de détecter automatiquement ou semi-automatiquement des spots dans ce type d'image. Citons, à titre d'exemple, le logiciel Mélanie [5, 152].

Les images de gels 2D d'électrophorèse à traiter sont loin d'être parfaites. Les spots peuvent avoir des formes très différentes : circulaires, elliptiques, etc. De plus, les images contiennent des traînées ou des taches d'eau.

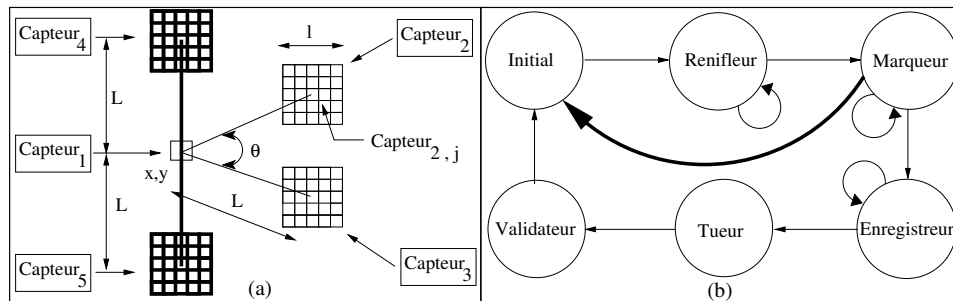


Figure 3.11: (a) Les capteurs d'un agent, (b) Automate décrivant le comportement d'un agent.

Les agents utilisés pour cette application sont très similaires à ceux utilisés pour la détection d'anneaux (voir la figure 3.11) car dans les deux cas, il faut détecter des formes arrondies. Un agent possède toujours deux capteurs frontaux ($Capteur_2$ et $Capteur_3$). De plus, il a maintenant deux capteurs latéraux ($Capteur_4$ et $Capteur_5$). Ces deux nouveaux capteurs ont pour but de lui indiquer son côté le plus sombre. Ceci lui évite de tourner autour d'une tache blanche – une tache d'eau –. Si tel est le cas, l'agent doit être ré-initialisé (transition de l'état *Marqueur* vers l'état *Initialisation*). Les agents ne peuvent tourner que dans un sens (sens positif ou sens négatif). Le sens dans lequel ils sont autorisés à se déplacer est déterminé aléatoirement à leur création. Cette façon de procéder leur permet de mieux résister au bruit, et notamment de résister aux formes rectilignes superposées aux spots – les traits –. La figure 3.12 présente une image de gel d'électrophorèse 2D ainsi que les spots détectés.

3.4 Influence des paramètres des agents

Dans cette section, nous nous proposons d'analyser les divers paramètres des agents et leur influence sur la qualité de la détection d'anneaux de croissance sur des images d'otolithes de poissons.

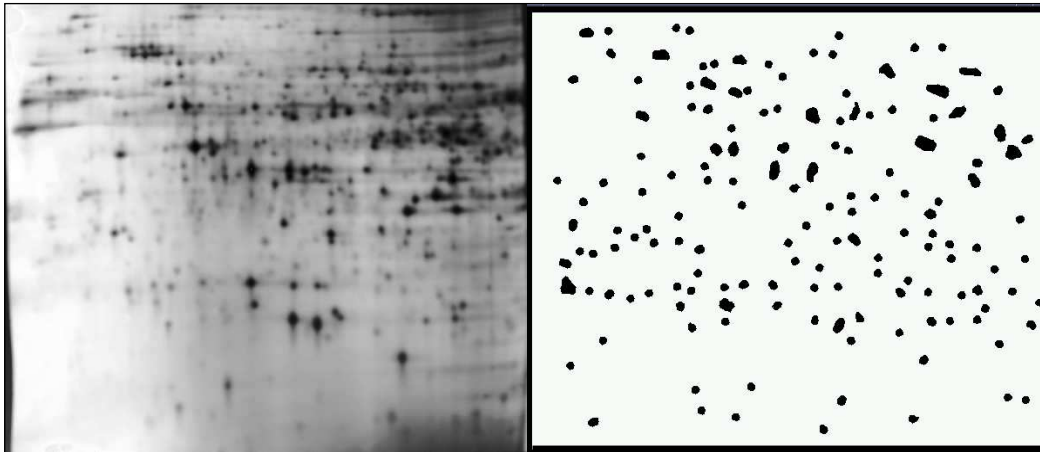


Figure 3.12: Une image et les spots détectés ; les spots sont remplis pour une meilleure visibilité.

Rappelons tout d'abord le comportement des agents et leurs paramètres.

Les agents sont placés aléatoirement sur l'image et sont guidés par leurs capteurs (voir la figure 3.9-a). Lorsque les agents atteignent le bord de l'image, ils sont replacés aléatoirement sur l'image. Un agent assombrireur essaye toujours de se déplacer dans la direction où les valeurs renvoyées par ses capteurs frontaux ($Capteur_2$ et $Capteur_3$) sont minimales. Un agent éclaircisseur fait l'inverse. Un agent modifie sa direction proportionnellement à la différence entre les valeurs retournées par ces deux capteurs.

A chaque étape, un agent calcule donc les valeurs renvoyées par ses capteurs $Capteur_2$ et $Capteur_3$. Dans la mesure où ces deux capteurs sont carrés et non orientés, il est possible de prétraiter l'image en appliquant, par exemple, un filtre moyenneur ou un filtre gaussien. Une fois ce prétraitement réalisé, nous pouvons utiliser des agents dont les capteurs frontaux sont réduits à des capteurs unitaires. Cette méthode peut s'avérer efficace si le nombre d'itérations (nombre total de pas effectués par les agents) est supérieur au nombre de pixels de l'image.

Dans la suite de cette section, nous étudions en détail les agents éclaircisseurs qui cherchent les maxima dans les images. Notons que le comportement des agents assombrieurs est symétrique et que l'étude serait similaire avec ces agents cherchant les minima dans les images. Notons également que nous utilisons des capteurs frontaux dont la taille est de un pixel ($l = 1$). Ceci est rendu possible grâce à un prétraitement de l'image qui peut être effectué séparément.

Les divers paramètres qu'il est donc possible d'ajuster afin de modifier le résultat de la détection des agents sont :

- la distance L entre la position centrale de l'agent et les capteurs $Capteur_2$ et $Capteur_3$, et
- l'angle θ séparant ces deux capteurs.

Afin de tester le comportement des agents en fonction des valeurs de leurs

paramètres, nous avons créé des images de synthèse contenant un contour en toit noyé dans un bruit gaussien. Les images de taille $128 * 128$ pixels comprennent un contour en toit situé en $x = 64$. La pente et la largeur du contour sont variables. La position du contour étant connue, il est aisé de comparer la détection du contour par les agents et la position réelle de ce contour.

Nous avons donc déterminé deux paramètres qu'il est nécessaire de faire varier. Ces deux paramètres ne sont pas complètement indépendants. En effet, si l'on désire régler la distance w entre les deux capteurs selon la largeur du contour, cette distance peut être obtenue avec différentes valeurs pour L et θ (voir la figure 3.13).

Cette distance w peut être très simplement calculée en utilisant l'équation (3.1). De plus, pour une valeur de θ donnée, si l'on souhaite une largeur w , on peut calculer la valeur de L à l'aide de l'équation (3.2).

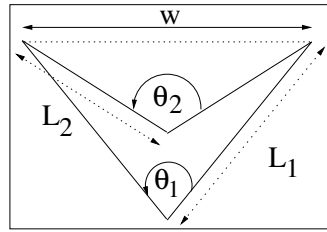


Figure 3.13: Deux combinaisons différentes de (L, θ) pour le même w .

$$w = 2.L_1.\sin\left(\frac{\theta_1}{2}\right) = 2.L_2.\sin\left(\frac{\theta_2}{2}\right) \quad (3.1)$$

$$L = \frac{w}{2.\sin\left(\frac{\theta}{2}\right)} \quad (3.2)$$

Dans la suite de l'étude sur l'influence des paramètres des agents, nous allons choisir une valeur fixe pour θ , le paramètre L pouvant être réglé ensuite.

3.4.1 Paramètre θ

Pour choisir le paramètre θ , nous avons envisagé deux cas extrêmes. Si l'angle θ est trop aigu, la distance entre les deux capteurs $Capteur_2$ et $Capteur_3$ peut s'annuler, ce qui ne doit pas arriver si on désire effectuer une comparaison entre les intensités des deux capteurs. En effet, si la différence d'intensité entre les deux capteurs est nulle, l'agent ne modifiera jamais sa direction. Par exemple, si on a $L = 2$ et $\theta = \pi/13$, le calcul de w donnera $w = 0.48$ qui est inférieur à 0.5; l'entier le plus proche étant 0, la distance entre les deux capteurs sera nulle. Par contre, si l'angle θ est trop obtus, comme π , la direction de l'agent sera imprécise, il effectuera une oscillation autour de la position du

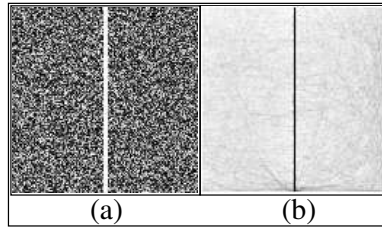


Figure 3.15: (a) Image d'un contour en toit (largeur 3 pixels, amplitude 60 niveaux de gris) noyé dans un bruit gaussien d'écart-type 2 (après une égalisation d'histogramme pour une meilleure visibilité) ; (b) Chemins des agents sur cette image avec $L = 2$ (niveaux de gris inversés).

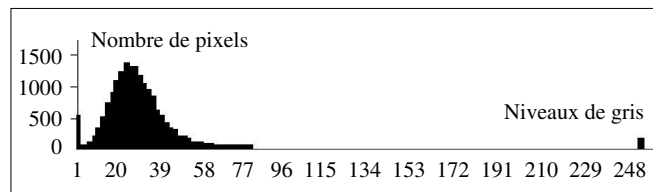


Figure 3.16: Histogramme de l'image des chemins des agents.

un seuillage automatique pour dissocier ces deux lobes [44]. L'image binaire obtenue peut alors être utilisée pour comparer le contour détecté avec le contour parfait en utilisant les critères de Canny (voir les équations (3.3), (3.4) et (3.5)). Nous avons ajouté un autre critère qui permet d'exprimer la qualité de la détection avant la binarisation des chemins des agents (voir l'équation (3.6)). Nous allons maintenant essayer d'optimiser ces critères en choisissant la meilleure valeur pour L lorsque $\theta = \pi/4$.

$$c_1 = 100. \frac{\text{nombre_de_pixels_détectés_n'étant_pas_des_points_contours}}{\text{nombre_de_pixels_non_contours}} \quad (3.3)$$

$$c_2 = 100. \frac{\text{nombre_de_pixels_contours_non_détectés}}{\text{nombre_de_pixels_contours}} \quad (3.4)$$

$$c_3 = \text{distance_moyenne_entre_la_position_du_contour_détecté} \\ \text{et_la_position_réelle_du_contour} \quad (3.5)$$

$$c_4 = \text{niveau_de_gris_moyen_des_pixels_contours} \\ - \text{niveau_de_gris_moyen_des_pixels_non_contours} \quad (3.6)$$

3.4.3 Paramètre L

Quand L augmente, la distance entre les deux capteurs de l'agent augmente également. Il est donc préférable de garder L faible pour obtenir une localisation plus précise des

capteurs des agents de part et d'autre du contour. Choisir $L = 1$ pourrait introduire des problèmes de chevauchement des deux capteurs, à cause de l'arrondi des coordonnées des agents à l'entier le plus proche. Avec $L = 2$, si le niveau de bruit est relativement faible, la détection peut être satisfaisante, pour des contours de largeur variable. Cependant si l'image est très bruitée, une valeur plus forte de L permet d'obtenir une meilleure détection. Afin d'étudier le paramètre L , nous avons créé une image de synthèse contenant un contour en toit que l'on est susceptible de trouver dans une image d'otolithe. La pente du toit est de 1.52 et correspond à la différence de niveau de gris entre deux pixels adjacents du toit. La largeur du toit est de 73 pixels et correspond à la distance entre les deux bases du toit. Nous avons ensuite ajouté à cette image un bruit gaussien d'écart-type 2 représentant le bruit caractéristique présent dans une image d'otolithe. Nous avons testé nos agents sur cette image en augmentant progressivement la valeur de L . En augmentant L , nous avons amélioré les critères c_1 , c_2 , c_3 et c_4 (voir le tableau 3.1). L'augmentation de L permet à l'agent d'avoir une vue plus générale du contour bruité, et lui permet donc d'améliorer sa détection (voir la figure 3.17).

L	c_1	c_2	c_3	c_4
2	1.27	11.36	0.2	115
3	0.76	9.09	0.18	141
4	0.21	36.36	0.25	101
5	0.29	18.18	0.12	121
6	0.72	0	0	207
7	1.46	0	0	186
8	1.39	0	0	181
9	1.68	0	0	201
10	0.52	0	0	193

Tableau 3.1: Critères obtenus avec différentes valeurs de L sur des images très bruitées.

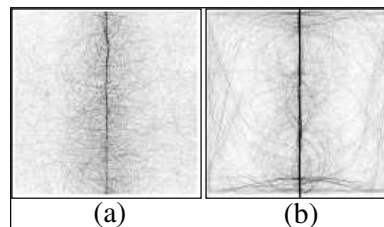


Figure 3.17: Chemins des agents. (a) $L = 2$; (b) $L = 6$.

Comme le montre le tableau 3.2, pour un rapport signal sur bruit correct (RSB : 10 fois le logarithme de la pente du toit divisé par l'écart-type du bruit), nous pouvons obtenir une bonne détection en gardant $L = 2$.

Pente	Largeur	Bruit	c_1	c_2	c_3	c_4
15	[1, 30]	Aucun	0.09	0	0	145
15	[1, 30]	$\sigma = 2$	0.00	0	0	153
15	[1, 27]	$\sigma = 4$	0.00	0	0	176
[1.5, 55]	3	Aucun	0.41	0	0	123
[3.5, 55]	3	$\sigma = 2$	0.01	0	0	212
1	1	Aucun	0	0	0	218
1	1	$\sigma = 1$	1.64	59.09	1.54	48
254	1	Aucun	0	0	0	106
254	1	$\sigma = 2$	0	0	0	110
254	1	$\sigma = 12$	0	0	0	146

Tableau 3.2: Caractéristiques (pente et largeur) d'un contour en toit et critères de détection obtenus avec $L = 2$ (valeurs maximales pour chaque ensemble) ; σ est l'écart-type du bruit gaussien ajouté aux images.

3.4.4 Robustesse de la méthode et application aux images d'otolithes

La méthode a également été évaluée sur une image test décrite par Ziou [157]. Cette image, de taille 150×150 pixels, contient des lignes horizontales et verticales de pente 50 (voir la figure 3.18-a). Un bruit d'écart-type 10 (RSB = 7dB) est ajouté dans le triangle supérieur droit de l'image, et un bruit d'écart-type 5 (RSB = 10dB) est ajouté dans le triangle inférieur gauche. Le résultat de la détection lorsque $L = 2$ est présenté figures 3.18-b et 3.18-c.

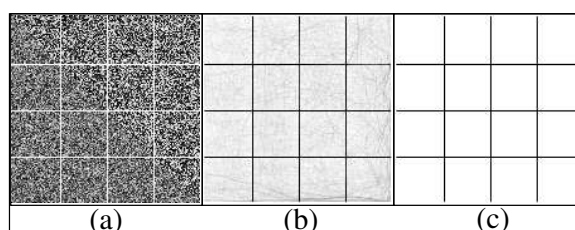


Figure 3.18: (a) Image bruitée définie par Ziou (après une égalisation d'histogramme) ; (b) Chemins des agents, $L = 2$, niveaux de gris inversés ; (c) Chemins des agents après un seuillage automatique.

Nous avons appliqué le même traitement sur des images de taille 512×512 pixels d'otolithes de poissons. Les résultats sont présentés figure 3.19. Ces résultats montrent que la meilleure localisation des stries est obtenue pour $L=8$. Rappelons que les autres paramètres des agents à prendre en compte sont l , fixé à 1, et θ , dont la valeur la plus adéquate est $\pi/4$.

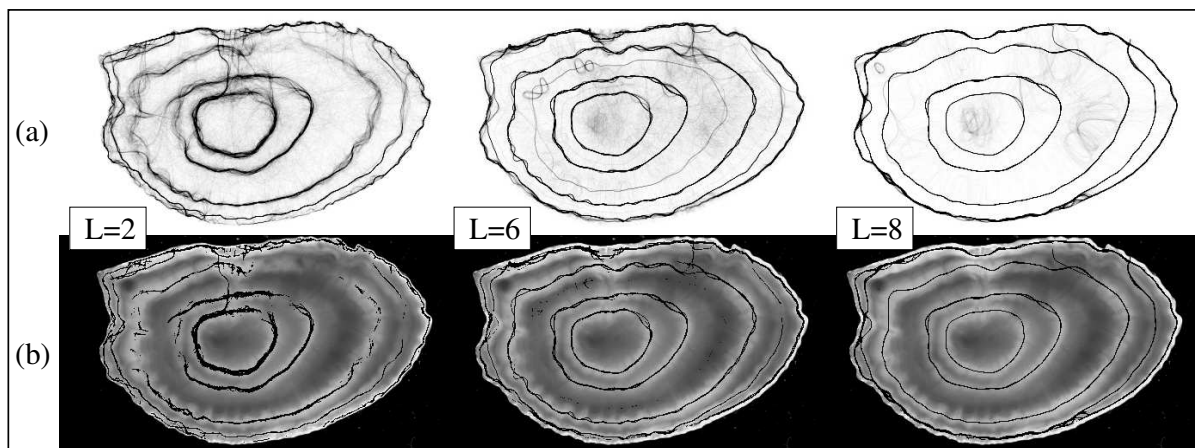


Figure 3.19: (a) Chemins des agents pour différentes valeurs de L ; (b) Chemins après seuillage.

La figure 3.20 montre un ensemble de résultats obtenus à l'aide de notre méthode de détection de stries. Les otolithes présentés correspondent à des poissons âgés de 1 à 8 ans. Plus un poisson est âgé, plus les stries externes de son otolithe sont difficiles à détecter car elles sont très fines et rapprochées.

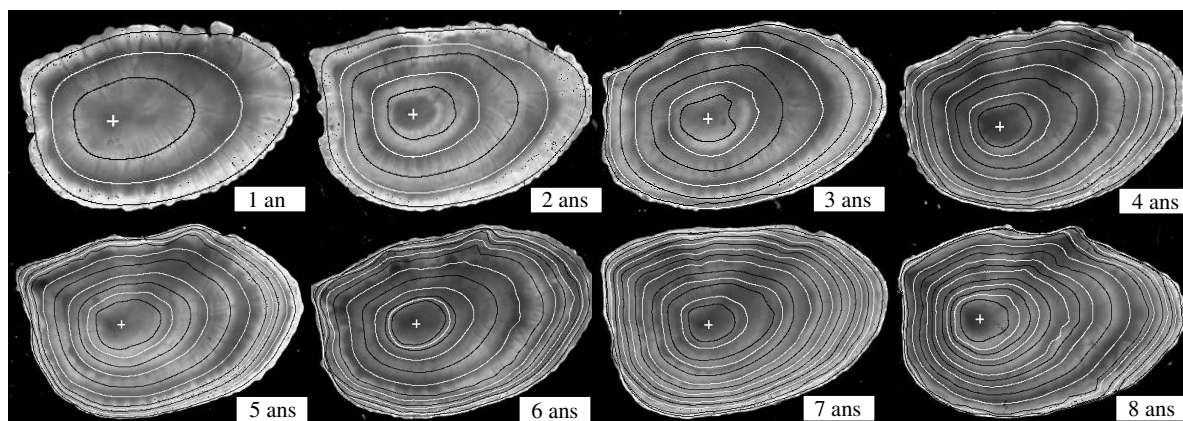


Figure 3.20: Résultats obtenus avec des otolithes de 1 à 8 ans.

Nous avons appliqué notre système multi-agents de détection de stries sur un échantillon test de 119 images d'otolithes de plies. Pour évaluer nos résultats, nous avons demandé à un expert humain d'estimer l'âge de chaque otolithe. Rappelons que cette estimation, étant liée à la subjectivité de l'expert, est d'une fiabilité limitée. Le nombre d'anneaux sombres et d'anneaux clairs sur l'image donne le nombre de saisons vécues par le poisson, et connaissant la saison où le poisson a été pêché, les experts peuvent estimer l'âge du poisson.

Le tableau 3.3 permet de comparer les résultats de notre méthode de détection de stries avec les méthodes développées précédemment. Notre méthode donne de meilleurs

résultats que ceux obtenus avec la méthode mono-dimensionnelle décrite dans [153]. Cette méthode se base sur une extraction de radiale du nucleus vers le bord de l'otolithe et une détection d'extrema sur les profils d'intensité obtenus. Malheureusement cette méthode ne prend pas en compte la continuité des structures recherchées et nécessite un pointage du nucleus par un opérateur humain. Notre méthode donne également de meilleurs résultats que la méthode de templates déformables – appelés aussi modèles prédéfinis déformables –. Cette méthode utilise la forme du contour externe de l'otolithe réduite à une échelle inférieure par homothétie centrée sur le nucleus [16, 148]. Cette méthode ne permet de donner qu'une approximation de la forme des anneaux de croissance. De plus, il est nécessaire d'effectuer un pointage manuel du nucleus. Une autre méthode appelée méthode par construction de graphes [128] donne des résultats similaires à la nôtre. Cette méthode par construction de graphes requiert une transformation polaire de l'image centrée sur le nucleus qui doit être donné interactivement. Les pics correspondant aux anneaux clairs et les vallées aux anneaux sombres sont obtenus par transformation morphologique. L'image est ensuite binarisée et labellisée afin d'obtenir des objets. Enfin, les objets proches sont reliés pour reconstruire les anneaux.

Groupes d'âge	1D	Templates	Graphes	Agents
1–3 ans	50%	100%	90%	87%
1–5 ans	50%	80%	80%	82%
5–8 ans		20%	60%	68%

Tableau 3.3: Pourcentage de bonne estimation de l'âge avec différentes méthodes.

Pour conclure, nous pouvons dire que notre méthode de détection de stries permet de percevoir la continuité des contours dans des images texturées, bruitées et peu contrastées. Les méthodes précédemment développées nécessitaient l'intervention d'un opérateur pour indiquer la position du nucleus. Un tel pointage n'est plus utile dans notre système. Notre méthode présente l'originalité d'utiliser des agents qui détectent des contours locaux dans les images, tout en percevant la continuité de ces contours lors de leurs déplacements.

3.5 Arrêt de notre système multi-agents par auto-régulation

Nous présentons ici l'utilisation de mécanismes immunitaires pour l'auto-régulation de systèmes multi-agents réactifs. Plus précisément, l'objet de cette section est de montrer comment il est possible de tirer parti de phénomènes immunitaires pour auto-réguler des populations d'agents. Notons que la régulation de systèmes multi-agents grâce à des principes immunologiques n'est encore que peu abordée [30].

Cette régulation peut se faire en intégrant le comportement de cellules immunitaires à celui des agents. Ces comportements peuvent aboutir par exemple à la mitose, à l'apoptose ou à la différenciation qui sont des mécanismes immunitaires indispensables au bon fonctionnement de la réponse immunitaire. Dans notre système auto-régulé, le travail à effectuer par les agents – le problème à résoudre – est vu comme un corps étranger, c'est-à-dire comme un antigène à éliminer.

Les agents représentent les cellules immunocompétentes ayant pour but l'élimination de l'antigène. Cette opération doit se faire avec une certaine efficacité, c'est-à-dire en effectuant le travail nécessaire (l'inverse de l'immunodéficience) sans pour autant trop en faire (l'inverse de l'allergie). Chaque agent hérite d'un ou plusieurs comportements de cellules. Ces comportements sont tirés des cellules immunitaires qui ont toutes des rôles bien définis.

Le premier rôle consiste à détecter l'antigène (le travail à effectuer), le second à donner l'alerte à grande échelle, le troisième à faire croître l'efficacité et la précision de la réponse et le quatrième à éliminer effectivement l'antigène. Nos agents reprennent ces différents rôles pour mimer une réponse immunitaire.

Voici, en trois points, les raisons de notre choix de la réponse immunitaire comme métaphore pour réguler les systèmes multi-agents :

- Le système immunitaire est composé d'entités autonomes capables de coopérer, dotées d'un comportement, de capteurs et de moyens d'action. Une cellule est donc proche du concept agent.
- Le système immunitaire est capable de différencier le "soi" du "non soi", c'est-à-dire de détecter un antigène (un travail à effectuer), parmi 10^{16} "patterns" ou formes différentes. Ce système est donc souple et adaptatif ce qui procure un avantage certain dans des environnements à forte variabilité (comme les images aériennes [102]). Pour les besoins de la simulation, le nombre de formes possibles peut être réduit [138].
- Le système immunitaire humain est quasi optimal dans la puissance de la réponse à mettre en œuvre pour éliminer l'antigène. Dans le cadre des systèmes multi-agents, ce point peut s'avérer intéressant puisqu'il permettrait une utilisation quasi optimale des ressources machines lors de traitements.

Nous commençons cette section par l'étude des concepts immunitaires que nous utilisons comme métaphore pour la régulation d'une population d'agents. Ensuite, nous présentons l'application de concepts immunitaires pour arrêter automatiquement notre système multi-agents de détection de stries. Finalement, nous concluons sur l'intérêt de cette approche immunitaire pour la conception de systèmes multi-agents.

3.5.1 Principes immunitaires pour la régulation de systèmes multi-agents

Nous abordons dans cette section l'utilisation d'un certain nombre de mécanismes immunitaires pour le développement de systèmes multi-agents auto-régulés.

Cette section décrit donc plusieurs types de phénomènes immunitaires impliqués dans l'auto-régulation. Nous voyons les sélections dites négatives et positives permettant d'éviter la présence de cellules (ou d'agents) inutiles ou perturbatrices dans le système. Puis, nous abordons les phénomènes d'activation, de différenciation, de prolifération et de mort cellulaire programmée (apoptose). Ces phénomènes sont à la base des mécanismes d'auto-régulation en cours de réponse immunitaire. La coopération entre les lymphocytes T et les lymphocytes B au sein du système immunitaire permet aussi une limitation des risques de dérive (sous-traitement de l'antigène ou effets de bords non désirés) tout en permettant une adaptation localisée dans le temps et l'espace. Ce dernier point ne sera pas décrit ici. Finalement, nous proposons une architecture globale de système multi-agents basée sur les principes immunitaires cités ci-dessus.

3.5.1.1 Sélections négative et positive pour la régulation en amont

Les sélections négative et positive sont des mécanismes clefs intervenant dans le système immunitaire. C'est grâce à ces sélections que le système immunitaire est capable de distinguer le "soi" du "non-soi".

Notre approche concerne l'optimisation du nombre d'entités dans les systèmes multi-agents à l'aide de mécanismes de régulation immunitaire. C'est pourquoi de notre point de vue, il ne s'agit pas d'apprentissage mais bien d'auto-régulation par la suppression des agents qui sont soit inutiles soit trop actifs. Dans les faits, cela consiste pour un agent qui s'estime inadapté à s'auto-détruire avant de devenir actif dans le système. La décision est prise localement au niveau de l'agent. Aucun contrôleur global n'est nécessaire. En pratique, chaque agent nouvellement créé est soumis à une sélection. Si sa réactivité est trop faible ou trop forte, il entre en apoptose. La difficulté réside dans la détermination des valeurs minimales et maximales de réactivité acceptables pour le problème à résoudre.

3.5.1.2 Mécanismes de base de la régulation pendant le traitement

Les principes d'activation, de différenciation, de prolifération et d'apoptose (ou mort cellulaire programmée) sont essentiels au bon déroulement des mécanismes

immunitaires (voir la section 2.6.1.1 du chapitre 2). C'est grâce à eux que le système immunitaire est capable d'accroître spécifiquement son potentiel envers un ou plusieurs antigènes. Plus précisément, le système immunitaire augmente tout d'abord le nombre de cellules dirigées contre un antigène. C'est la première phase de la réponse immunitaire. Puis, lors de la deuxième phase, une fois les antigènes éliminés, le nombre de cellules diminue (voir la figure 3.21).

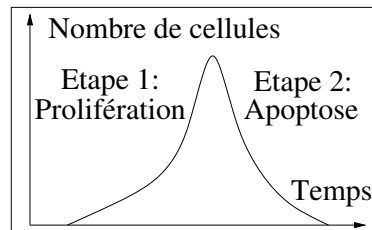


Figure 3.21: Les deux principales phases d'une réponse immunitaire.

a) Activation

L'activation est l'étape qui initie la réponse du système immunitaire contre une ou plusieurs substances antigéniques. Lors de l'activation d'une cellule, celle-ci change de morphologie et peut aussi changer de rôle. Nous observons donc à la fois une modification de la structure, de la forme cellulaire mais aussi du comportement.

Les changements structuraux permettent par exemple une amélioration de la mobilité, une sensibilité accrue aux messagers chimiques et/ou une variation de l'espérance de vie. Par exemple, un macrophage ayant phagocyté un agent pathogène devient capable de présenter cet antigène aux lymphocytes T4.

L'activation dépend de l'état interne de la cellule et de son environnement local. Nous limitons ici notre étude à l'activation des lymphocytes B et T.

Un lymphocyte (B ou T) doit, pour être activé, recevoir deux types de signaux. Le premier est un signal provenant de l'antigène et le deuxième est un signal appelé facteur de croissance (interleukine).

Le signal basé sur l'antigène peut utiliser deux voies. La première s'appelle endocytose. C'est-à-dire que l'antigène est détecté directement par le lymphocyte – exemple : antigène et lymphocyte B –. La deuxième forme est une signalisation via une Cellule Présentant l'Antigène (CPA) – exemple : CPA et lymphocyte T4 –. La première voie permet une action rapide contre l'antigène mais en même temps, laisse la voie libre à une réaction démesurée. La deuxième voie modère l'activation directe et sécurise la réponse immunitaire.

Pour l'auto-régulation, l'activation est essentielle. En effet, seuls les agents activés sont capables de proliférer. Inversement, un agent n'ayant reçu qu'une activation partielle voit son espérance de vie diminuer. Dans ce cas, cela signifie que l'agent n'est plus utile ou qu'il n'est pas adapté au problème.

Au cours du temps, les agents sont susceptibles de se trouver dans un des trois états suivants :

- L'agent est activé. Dans ce cas, il acquiert de nouvelles propriétés comme une réceptivité accrue à certains facteurs de croissance, un accroissement de sa durée de vie, de nouvelles capacités ou de nouveaux objectifs.
- L'agent est partiellement activé. Il réduit alors sa durée de vie. En effet, il est certainement devenu inutile car le problème dont il s'occupe est probablement résolu.
- L'agent est non activé. Il attend simplement de l'être.

b) Différenciation – Maturation

La différenciation permet aux cellules de se spécialiser au cours de leur évolution. Ces spécialisations permettent au système immunitaire d'augmenter son efficacité contre un antigène. Ce phénomène est un processus lié à la maturité de la cellule (son âge), aux signaux extérieurs ou au deux conjointement. La différenciation implique à la fois des changements morphologiques comme l'augmentation ou la diminution de la taille de la cellule, la création de certains récepteurs de surface mais aussi des modifications comportementales.

Nous traitons la différenciation des agents à l'aide des propriétés dynamiques de notre langage oRis qui permet une granularité au niveau de l'instance. Le comportement de l'agent qui se différencie effectue un saut qualitatif. Cela se répercute sur le système tout entier qui subit donc une augmentation qualitative de sa capacité à résoudre un problème donné.

c) Prolifération

La prolifération est le mécanisme par lequel le système immunitaire augmente sa capacité d'élimination de l'antigène. Cette augmentation se divise en deux parties distinctes. La première correspond à une augmentation quantitative du nombre de cellules combattant le corps étranger. La deuxième est une augmentation qualitative, c'est-à-dire que les cellules éliminent de mieux en mieux l'antigène. En effet, les cellules qui prolifèrent voient leurs récepteurs muter. Cela implique une augmentation de l'affinité entre les cellules et l'antigène. Ce phénomène est appelé mitose avec hypermutation somatique.

En termes informatiques, la prolifération correspond à la création de nouveaux agents. Un agent nouvellement créé hérite des caractéristiques de son créateur. Cet héritage n'est pas rigide mais offre à la descendance la possibilité de muter, de changer, d'évoluer. Cette adaptabilité permet d'optimiser le système en cours de traitement.

d) Apoptose

L'apoptose correspond à la mort cellulaire programmée, c'est-à-dire que la cellule s'auto-détruit. Ce mécanisme est mis en œuvre lorsqu'une cellule n'est pas adaptée

pour l'élimination de l'antigène. Il permet d'éliminer les cellules qui soit ne servent à rien, soit peuvent être néfastes car trop actives.

Les agents capables de reproduction et d'apoptose sont fortement soumis aux stimuli qu'ils perçoivent. Un agent entre en apoptose en fonction de son état interne et lorsque son activation n'est que partielle (voir la figure 3.22).

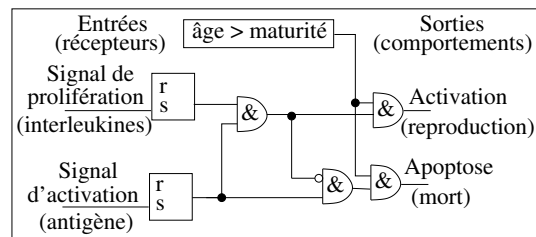


Figure 3.22: Schéma logique du déclenchement d'un comportement de type activation ou apoptose. Le seuil de maturité est ajusté au cours du temps en fonction de l'utilité de l'agent dans le système.

3.5.2 Structure générale de notre système multi-agents immunitaire

Nous pouvons regrouper les différents mécanismes immunitaires décrits précédemment dans un système reproduisant une partie du système immunitaire (voir la figure 3.23).

Nous avons tout d'abord, durant la phase 1, la génération des agents. Ensuite, phase 2, les agents subissent deux sélections. La sélection positive concerne les agents capables de réagir envers un problème donné. La sélection négative s'applique sur les agents ayant une réactivité trop grande pour le problème à traiter. Cette deuxième sélection permet au système d'éviter de traiter les informations de façon erronée. Cela correspond pour le système immunitaire à la destruction de toutes les cellules impliquées dans les maladies auto-immunes. Les phases 3, 4, 5, 6, 7 et 8 concernent la régulation de notre système.

Dans la section suivante, nous appliquons cette notion de système multi-agents immunitaires au domaine du traitement d'images biologiques, et plus précisément à la détection de stries.

3.5.3 Application au traitement d'images d'otolithes

Il nous a donc semblé intéressant d'étendre ce concept de régulation aux systèmes multi-agents qui n'ont *a priori* rien à voir avec l'immunologie.

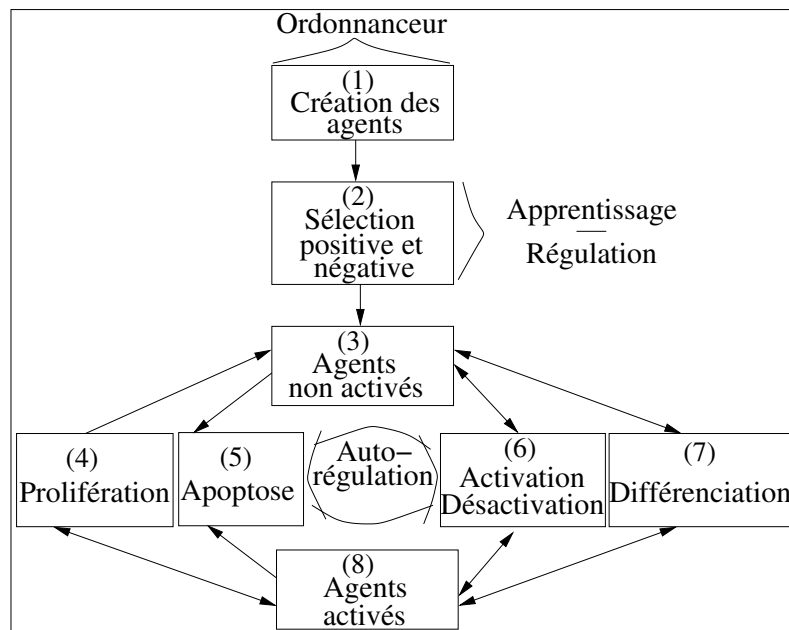


Figure 3.23: Système de régulation immunitaire pour systèmes multi-agents.

Pour cela, il fallait étendre la notion d'antigènes à la notion de travail à effectuer et la notion d'interleukine au concept de stimulant. Ainsi, un agent effectuant un travail quelconque peut hériter des propriétés des lymphocytes B pour être un acteur à part entière de la régulation du système auquel il appartient. Nous avons donc fait hériter nos agents de traitement d'images des propriétés des agents lymphocytes B. Ils sont donc maintenant capables de se reproduire et de se détruire.

Avec ce nouveau modèle, un traitement pour détecter des stries concentriques commence par la création de deux agents (un agent éclaircisseur et un agent assombrisseur) placés au hasard sur l'image à traiter (l'environnement). Des stimulants sont également déposés, eux aussi de façon aléatoire, dans l'environnement. Lorsqu'une strie est détectée par un agent, cet agent génère des stimulants placés aléatoirement sur l'image.

Grâce aux stimulants, les agents vont se reproduire et accomplir le traitement. Lorsque les stimulants vont disparaître, les agents vont mourir. La figure 3.24 présente l'évolution du nombre d'agents en fonction du temps.

En ce qui concerne la qualité de détection d'anneaux sur des images d'otolithes, les résultats obtenus avec ce système régulé sont très semblables à ceux obtenus avec le même système sans régulation. Toutefois, comme nous le verrons dans la section suivante, le système sans régulation peut fournir des résultats très instables lorsque le nombre d'agents est trop faible ou au contraire trop important.

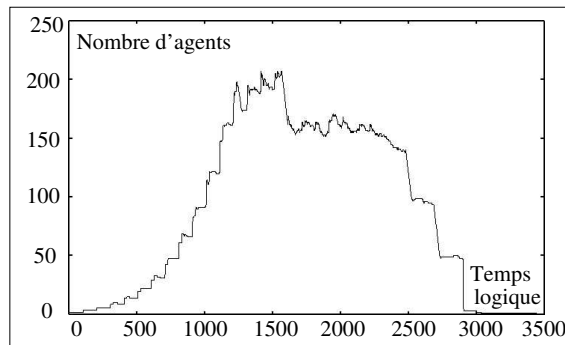


Figure 3.24: Evolution du nombre d'agents en fonction du temps.

3.5.4 Intérêt de la régulation

Cette approche donne des résultats intéressants dans la mesure où le système est capable de s'arrêter tout seul sans contrôleur global. Notons que le nombre maximum d'agents dépend du nombre de stimulants placés initialement sur l'image. L'idée pour ne pas dépendre de ce nombre initial de stimulants consisterait à placer des agents spécialisés dans la détection du travail à effectuer comme le font les macrophages et les lymphocytes T4 lors d'une réponse immunitaire. Ils placeraient des stimulants sur les stries encore non détectées, ce qui stimulerait les agents travaillant directement sur l'image.

Analysons maintenant l'impact de la régulation immunitaire sur la qualité du traitement. Pour cela, nous avons effectué plusieurs traitements avec régulation en changeant le nombre maximum d'agents. Les résultats obtenus sont comparés avec ceux d'un traitement sans régulation avec un nombre d'agents fixe (voir la figure 3.25). Nous observons que le système régulé est particulièrement efficace à partir d'un nombre suffisant d'agents. Par contre, sans régulation, le système devient très instable et peu performant lorsque le nombre d'agents est trop important. Un système multi-agents régulé via des principes immunitaires permet donc d'atténuer l'influence du nombre d'agents : si ce nombre est trop faible les agents vont avoir tendance à se dupliquer, alors que, s'il est trop important, un certain nombre d'agents vont entrer en apoptose. De cette manière, le système s'adapte de lui-même à son environnement.

Afin de confirmer l'intérêt de la régulation pour des simulations impliquant de grands nombres d'agents, nous évaluons les temps de traitements nécessaires à la détection de l'ensemble des anneaux de croissance présents sur une image d'otolithe. Dans le cas du système régulé, la figure 3.26 montre un temps de traitement quasi constant quel que soit le nombre d'agents. Sans régulation, le temps de traitement est plus faible lorsque peu d'agents sont utilisés. Par contre, dès que le nombre d'agents devient important, le système avec régulation est bien plus performant. Notons également que même si dans certains cas le temps de traitement est légèrement plus important avec régulation, le nombre de cycles du simulateur – son horloge interne –

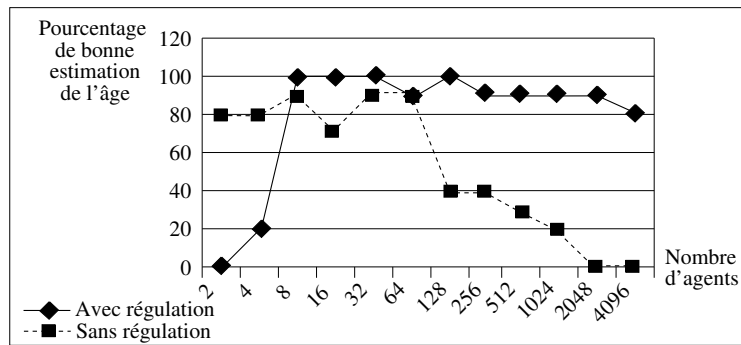


Figure 3.25: Comparaison de la qualité du traitement avec et sans régulation.

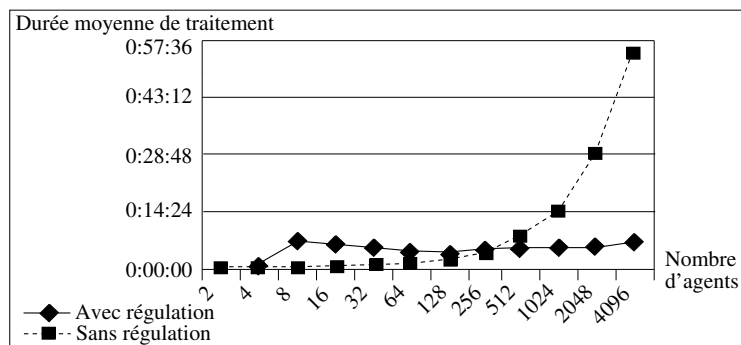


Figure 3.26: Durée du traitement avec et sans régulation.

reste toutefois inférieur.

La métaphore du système immunitaire offre donc aux informaticiens des principes de régulation qui peuvent être inclus dans des systèmes multi-agents pour l'optimisation de populations d'agents [11, 14].

3.6 Conclusion sur le traitement d'images par agents auto-régulés

Dans ce chapitre, nous avons présenté un système distribué de traitement d'images biologiques reposant sur le concept d'agents réactifs. Ceci signifie que dans notre système, chaque agent a un comportement très simple lui permettant de prendre une décision (trouver un contour, une région, ...) selon sa position dans l'image et l'information qui y est contenue.

Notre système utilise le langage oRis qui permet de décrire simplement et assez finement le comportement d'agents. Ce langage multi-agents est interprété et dynamique. Pour décrire le langage oRis, nous pouvons dire tout d'abord que c'est un langage objet classique avec l'utilisation de classes regroupant des méthodes

et des attributs. Sa syntaxe est proche de celle de C++ et permet l'héritage multiple. Mais oRis est également un langage orienté agent : tout objet disposant d'une méthode 'main()' devient automatiquement un agent. Cette méthode est cycliquement exécutée par l'ordonnanceur du système et correspond au comportement de l'agent. De plus, en oRis, un agent peut très simplement communiquer avec les autres grâce à différents mécanismes : appel de méthode, envoi de message et diffusion.

Nous avons également décrit un système multi-agents dédié à la détection de contours en toit dans des images biologiques. Le système proposé est composé d'agents dont la tâche individuelle est la détection d'extrema locaux dans des images en niveaux de gris. Les agents ont à leur disposition des capteurs leur permettant d'obtenir des informations image. En calculant la moyenne des niveaux de gris de chaque capteur placé autour de lui, un agent peut, en fonction de son comportement, décider de tourner dans telle ou telle direction. Nous avons analysé l'influence des divers paramètres intervenant dans leurs comportements. Les agents avec lesquels nous avons travaillé utilisent uniquement des informations de bas niveau et peuvent donc facilement être adaptés à différents types d'images (otolithe de poisson, gel d'électrophorèse 2D, ...).

Un des problèmes les plus importants des systèmes multi-agents est leur arrêt automatique. Afin d'essayer d'apporter un élément de réponse à ce problème, nous avons implémenté l'arrêt de notre système multi-agents par auto-régulation en utilisant des mécanismes immunitaires. Nous pouvons alors parler de système multi-agents immunitaire.

Chapitre 4

Conclusion et perspectives

Dans ce document, nous avons abordé plusieurs aspects de l'utilisation des systèmes multi-agents en biologie.

Nous avons tout d'abord montré le bien-fondé de l'approche multi-agents pour la modélisation et la simulation de phénomènes biologiques. Rappelons que les systèmes biologiques et les systèmes multi-agents ont un certain nombre de caractéristiques communes comme la robustesse, l'émergence, l'auto-organisation et l'adaptabilité, caractéristiques qui justifient d'autant plus notre approche. L'intérêt majeur de l'utilisation des systèmes multi-agents est, sans nul doute, la possibilité d'interagir avec le modèle en cours d'exécution. Le biologiste a désormais à sa disposition un véritable laboratoire d'expérimentation *in virtuo* où il peut, à tout moment, comme dans des conditions expérimentales *in vivo* ou *in vitro*, intervenir sur le déroulement de son expérience numérique en ajoutant – ou en enlevant – une cellule ou une molécule, en modifiant le comportement d'une ou de plusieurs cellules, etc.

Un autre aspect de notre contribution est l'utilisation de systèmes multi-agents pour l'acquisition de données biologiques par traitement d'images. Nous avons ainsi présenté deux applications dans des domaines aussi variés que la détection de stries concentriques sur des images d'otolithes et la détermination de spots sur des images de gels d'électrophorèse 2D. Les agents utilisés disposent de capteurs sur les images. Ils ont ainsi accès à des informations sur leur environnement local. Grâce à des comportements très simples, les agents sont capables de percevoir des contours locaux dans les images, tout en retrouvant la continuité de ces contours lors de leurs déplacements. D'autres types d'application sont envisageables puisque les agents utilisés ne travaillent qu'avec des informations de bas niveau. Ils peuvent donc être facilement adaptés.

Nous avons aussi abordé un problème qui, *a priori*, ne concerne que l'informatique et plus précisément l'arrêt automatique de systèmes multi-agents. Avec comme exemple le traitement d'images par système multi-agents, nous avons vu comment transposer des concepts immunitaires afin de réguler une population d'agents. Les agents héritent de

caractéristiques leur permettant de s'adapter à leur environnement local par mutation lors de la prolifération. Ils subissent également les sélections négatives et positives afin de ne garder que des agents adaptés au problème. Les résultats obtenus par cette approche révèlent une meilleure stabilité de la qualité du traitement ainsi qu'un nombre plus faible de cycles de simulation. L'utilisation de métaphores issues notamment de la régulation de phénomènes biologiques constitue, de notre point de vue, une avancée majeure dans le domaine des systèmes multi-agents. Cette voie de recherche doit être approfondie, nous sommes en effet certains que les phénomènes biologiques recèlent de nombreux mécanismes d'optimisation et de régulation non encore explorés par les informaticiens. Citons, par exemple, la balance TH1-TH2 intervenant dans l'allergie comme facteur déclencheur ou régulateur de la réaction immunologique.

Nous avons vu, au cours de ce document, que la modélisation de systèmes biologiques complexes est aujourd'hui, et plus que jamais, en plein essor. Même si notre méthodologie de modélisation multi-agents a montré tout son intérêt dans ce domaine, plusieurs points méritent d'être approfondis afin d'apporter une meilleure compréhension des mécanismes biologiques et de permettre le développement de méthodes informatiques utilisables à grande échelle par des médecins et des chercheurs en biologie.

Par exemple, nous pensons que la modélisation de systèmes biologiques ne peut être que multi-modèles. En effet, à la lumière de nos résultats, nos modèles semblent qualitatifs – ou explicatifs – plutôt que quantitatifs. La littérature en biologie ne fournissant en général pas suffisamment de données pour construire des modèles mathématiques précis, il faut savoir utiliser les données autant que possible si celles-ci existent, mais il faut aussi savoir pallier les manques grâce à différents outils de l'intelligence artificielle qu'il nous faut intégrer au sein d'un même modèle. Il serait également intéressant d'apporter une preuve formelle à nos différents modèles multi-agents de phénomènes biologiques. Nous pensons notamment évaluer l'impact, sur les résultats de simulation, du tirage aléatoire effectué lors de l'activation des agents. Cet aspect a d'ores et déjà commencé et se poursuivra avec l'aide d'un maître de conférences en mathématiques appliquées recruté récemment au sein de l'équipe *In Virtuo*, composante du Laboratoire d'Ingénierie Informatique de l'Ecole Nationale d'Ingénieurs de Brest. Un autre point à développer est certainement l'utilisation d'un ensemble d'ordinateurs en réseau afin de répartir la charge de calcul et donc, de permettre des simulations impliquant un grand nombre de cellules nécessaires à l'émergence de certaines fonctions biologiques complexes. Il se posera alors des problèmes classiques de distribution comme la synchronisation, la répartition et la cohérence. A l'heure actuelle, la notion de calcul utilisant une grille d'ordinateurs (*grid computing*) semble être une solution prometteuse.

Pour finir, notons que nos travaux devront se faire avec toute l'attention de l'informaticien qui doit toujours garder à l'esprit que la rencontre d'un nouveau concept ou d'une nouvelle propriété biologique peut éventuellement être la clef ouvrant la voie vers de nouvelles solutions à des problèmes informatiques.

Références Bibliographiques

- [1] J.-F. Abgrall, P. Ballet, S. Kerdélo, M. Nicolas, V. Rodin, and J. Tisseau. L'hématologue et la coagulation virtuelle. *Thromboses, sous la direction de C. Leroyer et al., éditions Margaux Orange*, pages 287–299, 2004.
- [2] G.A. Agha, K.M. Carley, J. Cuenca, E.H. Durfee, E. Ellis, L. Gasser, M.P. Georgeff, M.N. Huhns, T. Ishida, N. Jamali, S. Ossowski, H.V.D. Parunak, A.S. Rao, T.W. Sandholm, S. Sen, M.P. Singh, L.M. Stephens, G. Tel, J. Wainer, G. Weiss, M.J. Wooldridge, and M. Yokoo. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Ed. G. Weiss, Cambridge, MA, USA, 1999.
- [3] J. Aguilar. A survey about fuzzy cognitive maps papers (invited paper). *International Journal of Computational Cognition*, 3(2):27–33, June 2005.
- [4] G. Andral. *Essai d'hématologie pathologique*. Fortin, Masson et cie, Paris, 1843.
- [5] R. Appel, D. Hochstrasser, M. Funk, R. Vargas, C. Pellegrini, A. Muller, and J.-R. Scherrer. The melanie project - from a biopsy to automatic protein map interpretation by computer. *Electrophoresis*, 12:722–735, 1991.
- [6] Assemblée nationale - 2ème lecture. Projet de loi relatif à la bioéthique. 11 décembre 2003.
- [7] H. Atlan. *L'organisation biologique et la théorie de l'information*. Editions Hermann, 1992.
- [8] H. Atlan. La cellule vivante : un paradigme des systèmes naturels complexes. *Médecine/Sciences*, 18:764–766, 2002.
- [9] T.K. Attwood and D.J. Parry-Smith. *Introduction to Bioinformatics*. Prentice-Hall [Longman Higher Education], 1999.
- [10] R. Axelrod. *Structure of Decision*. Princeton University Press, Princeton, 1976.

- [11] P. Ballet. *Intérêts mutuels des systèmes multi-agents et de l'immunologie. Application à l'immunologie, l'hématologie et au traitement d'images*. PhD thesis, Thèse de doctorat en Informatique de l'Université de Bretagne Occidentale, Brest (France), 28 Janvier 2000.
- [12] P. Ballet, J.O. Pers, V. Rodin, and J. Tisseau. A multi-agent system to simulate an apoptosis model of B-CD5 cells. In *SMC'98, IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1–5, San Diego, CA (USA), 11-14 October 1998.
- [13] P. Ballet, V. Rodin, and J. Tisseau. Multiagent boundary detection system: A way to parallel image processing. In *Parallel and Distributed Methods for Image Processing I. SPIE's Optical Sciences, Engineering And Instrumentation'97*, volume 3166, pages 316–323, San Diego, CA (USA), 27 July – 1 August 1997.
- [14] P. Ballet, V. Rodin, and J. Tisseau. Immune mechanisms to regulate multi-agent systems. In *GECCO'00, Genetic and Evolutionary Computation Conference*, pages 33–35, Las Vegas (USA), 8 July 2000.
- [15] B.A. Barshop, R.F. Wrenn, and C. Frieden. Analysis of numerical methods for computer simulation of kinetic processes: development of KinSim – a flexible, portable system. *Analytic Biochemistry*, 130(1):134–145, 1 April 1983.
- [16] A. Benzinou. *Vers un système de vision par ordinateur pour la reconnaissance de structures sur des pièces calcifiées (otolithes et statolithes). Application à la détermination de l'âge et de paramètres de croissance de poissons et de céphalopodes*. PhD thesis, Thèse de Docteur en Électronique, Option Traitement du Signal et de l'Image, de l'Université de Bretagne Occidentale, Brest (France), 27 Janvier 2000.
- [17] U. Bhalla and R. Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283:381–387, 1999.
- [18] N. Blüthgen. *Dynamical models of signal transduction and the influence of feedback loop*. PhD thesis, Humboldt University Berlin, Berlin (Germany), 2001.
- [19] M.S. Boguski. Bioinformatics – a new era. *Trends Guide to Bioinformatics, Elsevier Trends Journal Supplement*, pages 1–3, 1998.
- [20] O. Boissier and Y. Demazeau. ASIC: An Architecture for Social and Individual Control and its Application to Computer Vision. In *Proceedings of MAAMAW Conference*, pages 135–149, Odense (Danemark), 2–4 August 1994.
- [21] E. Bonabeau, G. Theraulaz, J.L. Deneubourg, S. Aron, and S. Camazine. Self-organisation in social insects. *Trends in Ecology and Evolution (Elsevier)*, 12(5):188–193, May 1997.

- [22] A. Boucher and C. Garbay. A Multi-agent System to Segment Living Cells. In *Proceedings of the 13th IAPR International conference on Pattern Recognition (ICPR'96)*, IEE Computer Society Press, volume 3, pages 558–562, Vienna (Austria), 1996.
- [23] L. Boumsell, H. Coppin, D. Pham, B. Raynal, J. Lemerle, J. Dausset, and A. Bernard. An antigen shared by a human T cell subset and B cell chronic lymphocytic leukemic cells. Distribution on normal and malignant lymphoid cells. *Journal of Experimental Medicine*, 152:229–234, 1980.
- [24] F. Bourdon. *Systèmes d'information ouverts : sémantique interactionnelle des connaissances et systèmes multi-agents*. PhD thesis, Habilitation à Diriger des Recherches, Université de Caen, Caen (France), Janvier 1998.
- [25] J.F. Canny. A computationnal approach to edge detection. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.
- [26] C.G.L. Cao. Augmented reality displays for endoscopic orientation and navigation. In *CHI 2000 Conference on Human Factors in Computing Systems*, pages 91–92, The Hague, NL, 2000.
- [27] F. Celada and P.E. Seiden. A computer model of cellular interactions in the immune system. *Immunology Today*, 13(2):56–62, 1992.
- [28] D.L. Chao, M.P. Davenport, S. Forrest, and A.S. Perelson. Modelling the impact of antigen kinetics on T-cell activation and response. *Immunology and Cell Biology*, 82(1):55–61, February 2004.
- [29] J-M. Claverie, S. Audic, and C. Abergel. La bioinformatique: une discipline stratégique pour l'analyse et la valorisation des génomes. <http://igs-server.cnrs-mrs.fr/jcnrs.html> , 2000.
- [30] D. Dasgupta. *Artificial Immune Systems and Their Applications*. Springer-Verlag, New-York (USA), 1999.
- [31] L.N. De Castro and F.J. Von Zuben. Artificial Immune Systems: Part II - A Survey of Applications. Technical Report DCA-RT 02/00, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, state of São Paulo, Brazil, February 2000.
- [32] J. de Rosnay. *Le microscope : vers une vision globale*. Editions Point, 1975.
- [33] J.-P. Demailly. *Analyse Numérique Et Equations Différentielles*, chapter 5, pages 115–142. Presses Universitaires De Grenoble, 1996.
- [34] Y. Demazeau. From interactions to collective behaviour in agent-based systems. In *First European conference on cognitive science*, pages 117–132, Saint Malo, France, April 1995.

- [35] Y. Demazeau, O. Boissier, and J.L. Koning. Using Interaction Protocols to Control Vision System. In *Proceedings of IEEE International Conference on System, Man and Cybernetics*, San Antonio (USA), 1994.
- [36] Department of Defense. High Level Architecture Interface Specification – Version 1.3. *IEEE P1516.1, Standard for Modeling and Simulation (M&S)*, Avril 1998.
- [37] G. Desmeulles. Simulation cellulaire *in virtuo*. Perspective multi-agents à travers la réification des interactions. Technical report, DEA Informatique, IFSIC, Rennes (France), 2003.
- [38] G. Desmeulles. *Modèle informatique de l’allergie et application dans les systèmes immunitaires artificiels*. PhD thesis, Thèse de doctorat en Informatique de l’Université de Bretagne Occidentale, Brest (France), Soutenance prévue en 2006.
- [39] G. Desmeulles, D. Dupré, V. Rodin, and L. Misery. Mise au point d’un modèle de recherche *in virtuo* de la réaction allergique : résultats préliminaires. *Annales de dermatologie et de vénéréologie*, accepté, à paraître, 2005.
- [40] M. Dorigo, G. Di Caro, and M. Sampels, editors. *Ant Algorithms, Third International Workshop, ANTS 2002, Brussels, Belgium, September 12-14, 2002, Proceedings*, volume 2463 of *Lecture Notes in Computer Science*. Springer, 2002.
- [41] W.M. Feldman. *Biomathematics*. Griffin, London, GB, 1923.
- [42] J. Ferber. *Les systèmes multi-agents. Vers une intelligence collective*. InterEditions, Paris, 1995.
- [43] J. Ficini, N. Lumbroso-Bader, and J.-C. Depezay. *Structure De La Matière – Cinétique Chimique*, chapter 4, pages 149–189. Herman, 1978.
- [44] W.D. Fisher. On grouping for maximum homogeneity. *JASA*, 53:789–798, 1958.
- [45] E.L. Foo and C.G. Heden. Bioinformatics and the development of biotechnology. In *Microbial Technology in the Developing Countries*, Eds. E.J. DaSilva, Y.R. Dommergues, R.J. Nyns and C. Ratledge, *Oxford University Press*, pages 394–417, New York (USA), 1987.
- [46] J. Franklin. *The Future of The Medical Journal (S.P. Lock Ed.)*. British Medical Journal Press, 1991.
- [47] A. Funahashi, N. Tanimura, M. Morohashi, and H. Kitano. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BioSilico*, 1(5):159–162, November 2003.
- [48] C. Garbay. Les sciences du traitement de l’information comme pivot de l’interdisciplinarité. *Information-Interaction-Intelligence (I³)*, Cepaduès–Edition, 2(2), 2002.

- [49] N. Genetet. *Immunologie (4 Ed.)*. Editions Médicales Internationales, Paris, mai 2002.
- [50] D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.
- [51] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
- [52] D.F. Gillies, A. Haritsis, and C.B. Williams. Computer simulation for teaching endoscopic procedures. *Endoscopy*, 2:544–548, 1992.
- [53] N. Goodman. Biological data becomes computer literate: new advances in bioinformatics. *Current Opinion in Biotechnology, Elsevier*, 13 (1):68–71, 2002.
- [54] P.J.E. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 95(12):6750–6755, June 1998.
- [55] R. Greco-Perotto, F. Assimacopoulos-Jeannet, and B. Jeanrenaud. Insulin modifies the properties of glucose transporters in rat brown adipose tissue. *Biochemical Journal*, 247(1):63–8, 1 October 1987.
- [56] A. Guillaud. *Un système multi-agents pour la reconnaissance de formes sur des pièces calcifiées. Aide à l'estimation de l'âge de poissons et de céphalopodes*. PhD thesis, Thèse de Docteur en Électronique, Option Traitement du Signal et de l'Image, de l'Université de Bretagne Occidentale, Brest (France), 7 Décembre 2000.
- [57] A. Guillaud, H. Troadec, A. Benzinou, J. Le Bihan, and V. Rodin. A multiagent system for edge detection and continuity perception on fish otolith images. *EURASIP Journal on Applied Signal Processing*, 2002(7):746–753, July 2002.
- [58] R.M. Haralick. Ridges and Valleys on Digital Images. *Computer Vision, Graphics & Image Processing*, 22:28–38, 1983.
- [59] F. Harrouet. *oRis: s'immerger par le langage pour le prototypage d'univers virtuels à base d'entités autonomes*. PhD thesis, Thèse de doctorat en Informatique de l'Université de Bretagne Occidentale, Brest (France), 8 décembre 2000.
- [60] F. Harrouet, J. Tisseau, P. Reignier, and P. Chevaillier. oRis: un environnement de simulation interactive multi-agents. *RSTI-TSI*, 21(4):499–524, 2002.
- [61] J.S. Hege and L.J. Cole. A mathematical model relating circulating antibody and antibody forming cells. *Journal of Immunology*, 97:34–40, 1966.

- [62] H.C. Hemker and S. Beguin. Thrombin generation in plasma: its assessment via the endogenous thrombin potential. *Journal of Thrombosis and Haemostasis*, 74(1):134–8, July 1995.
- [63] V. Henri. Théorie générale de l'action de quelques diastases. *Comptes Rendus de l'Académie Des Sciences*, 135:916–919, 1902.
- [64] C. Hewitt. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323–364, June 1977.
- [65] T. Hideshima, D. Chauhan, R. Schlossman, P. Richardson, and K.C. Anderson. The role of tumor necrosis factor alpha in the pathophysiology of human multiple myeloma: therapeutic applications. *Oncogene*, 20:4519–4527, 2001.
- [66] R. Hightower, S. Forrest, and A.S. Perelson. *The Baldwin effect in the immune system: learning by somatic hypermutation*, pages 159–167. Belew, R.K and Mitchell, M. (Eds.), *Adaptive Individuals in Evolving Populations*, Addison-Wesley, Reading, MA (USA), 1996.
- [67] R. Hofestadt and S. Thelen. Quantitative modeling of biochemical networks. *In Silico Biology*, 1:39–53, 1998.
- [68] C.-Y.F. Huang and J.E. Ferrel Jr. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 93:10078–10083, September 1996.
- [69] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J.C. Doyle, H. Kitano, and the rest of the SBML Forum: A.P. Arkin, B.J. Bornstein, D. Bray, A. Cornish-Bowden, A.A. Cuellar, S. Dronov, E.D. Gilles, M. Ginkel, V. Gor, I.I. Goryanin, W.J. Hedley, T.C. Hodgman, J.-H. Hofmeyr, P.J. Hunter, N.S. Juty, J.L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L.M. Loew, D. Lucio, P. Mendes, E. Minch, E.D. Mjolsness, Y. Nakayama, M.R. Nelson, P.F. Nielsen, T. Sakurada, J.C. Schaff, B.E. Shapiro, T.S. Shimizu, H.D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.
- [70] L. Hunter and R.H. Lathrop. Computer science and biology: An unlikely pair. *IEEE Intelligent systems*, pages 8–10, March/April 2002.
- [71] H. Ishikawa, M.S. Mahmoud, R. Fujii, S. Abroun, and M.M. Kawano. Proliferation of immature myeloma cells by interleukine-6 is associated with CD45 expression in human multiple myeloma. *Leukemia and Lymphoma*, 39(1-2):51–55, 2000.
- [72] N. Ishikawa, H. Tsuyama, S. Abroun, S. Liu, O. Taniguchi, and M.M. Kawano. Requirement of src kinase activity associated with CD45 fo myeloma cell proliferation by interleukine-6. *Blood*, 99(6):2172–2178, 2002.

- [73] A. Jacquart and A. Kahn. *L'avenir n'est pas écrit*. Bayard, Août 2001.
- [74] C. Jamin, R. Le Corre, P.M. Lydyard, and P. Youinou. Anti-CD5 extends the proliferative response of human CD5⁺ B cells activated with anti-IgM and interleukine 2. *European Journal of Immunology*, 26:57–62, 1996.
- [75] M.R. Jean (collectif). Emergence et SMA. In *5èmes Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents (JFIADSMA '97)*, Hermès, Eds. J. Quinqueton, M.-C. Thomas, B. Trousse, pages 323–342, La Colle-sur-Loup, France, 2–4 avril 1997.
- [76] N.K. Jerne. Towards a network theory of the immune system. *Annales d'Immunologie de l'Institut Pasteur, série C*, 125:373–389, 1974.
- [77] N.K. Jerne. Idiotypic networks and other preconceived ideas. *Immunological Reviews*, 79:5–24, 1984.
- [78] F. Jobin. *L'hémostase*. Les Presses De l'Université Laval – Editions Maloine, Québec (Québec), Canada, 1995.
- [79] E. Jolivet. *Introduction aux modèles mathématiques en biologie*, chapter 5, pages 70–88. Masson, 1983.
- [80] C.M. Jonker, J.L. Snoep, J. Treur, H.V. Westerhoff, and W.C.A. Wijngaards. Putting intentions into cell biochemistry: an artificial intelligence perspective. *Journal of Theoretical Biology*, 214:105–134, 2002.
- [81] B.N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascade. *European Journal of Biochemistry*, 267(6):1583–1588, March 2000.
- [82] H. Kitano. *Foundations of Systems Biology (Edited by Hiraoki Kitano)*. MIT Press, Cambridge, MA (USA), 2001.
- [83] H. Kitano. System biology : a brief overview. *Science*, 295:1662–1664, March 2002.
- [84] H. Kitano. A graphical notation for biochemical networks. *BioSilico*, 1(5):169–176, November 2003.
- [85] S.H. Kleinstein and P.E. Seiden. Simulating the immune system. *Computing in Science and Engineering*, pages 69–77, July/August 2000.
- [86] I. Koch, S. Schuster, and M. Heiner. Simulation and analysis of metabolic networks by time-dependant petri nets. In *Computer Science and Biology - GCB'99 Proceeding of German Conference of Bioinformatic*, pages 208–210, Hannover, Germany, October 4–6 1999.

- [87] B. Kosko. Fuzzy cognitive maps. *International Journal Man-Machine Studies*, 24:65–75, 1986.
- [88] D.C. Krakauer and J.B. Plotkin. Redundancy, antiredundancy, and the robustness of genomes. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 99(3):1405–1409, February 2002.
- [89] J.U. Kreft, G. Booth, and J.W.T. Wimpenny. Bacsim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, 144:3275–3287, 1998.
- [90] K.M. Kyoda, M. Muraki, and H. Kitano. Construction of a generalized simulator for multi-cellular organisms and its application to SMAD signal transduction. In *Pacific Symposium on Biocomputing*, volume 5, pages 314–325, Hawaii (USA), 4–9 January 2000.
- [91] N. Le Novère and T.S. Shimizu. StochSim: modelling of stochastic biomolecular processes. *Bioinformatics*, 17(6):575–576, 2001.
- [92] R.J. Leipold, T.A. Bozarth, A.L. Racanelli, and I.B. Dicker. Mathematical model of serine protease inhibition in the tissue factor pathway to thrombin. *The Journal Of Biological Chemistry*, 270(43):25383–25387, October 1995.
- [93] C. Levinthal. Molecular model-building by computer. *Scientific American*, pages 42–52, June 1966.
- [94] H.A. Lim. Bioinformatics. <http://www.d-trends.com/Bioinformatics/>, 1987.
- [95] J. Liu. Reactive Agents for Adaptative Image Analysis. In *Proceedings of Second International conference on Autonomous Agents*, volume 3, pages 441–448, Minneapolis/St. Paul, MN (USA), 1998.
- [96] J. Liu and Y.Y. Tang. Adaptative Image Segmentation With Distributed Behavior Based Agents. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 6:544–551, 1999.
- [97] M. Lückenhaus. A Multi-Agent System for Parallelizing Image Analysis Tasks. In *International conference on Intelligent Autonomous Systems (IAS)*, pages 579–586, Sapporo (Japan), June 1–4 1998.
- [98] D.J. MacEwan. TNF receptor subtype signalling : Differences and cellular consequences. *Cellular Signalling*, 14:477–492, 2002.
- [99] P. Maes. Modeling adaptive autonomous agents. *Artificial Life Journal, MIT Press, Ed C. Langton*, 1(1/2):135–162, Fall’93/Winter’94, 1994.
- [100] P. Marcenac. Modeling multiagent systems as self-organized critical systems. In *31th Hawaii International Conference on System Sciences (HICSS’31), IEEE Computer Society Press, Ed. D.R. Dolk*, volume 5, pages 86–95, Kohala Coast, HI, USA, 6–9 January 1998.

- [101] Groupe MARCIA. Auto-organisations : émergence de structures. In *Journée du PRC GDR IA "Les systmes multi-agents"*, Editions Hermès, 2 février 1996.
- [102] D.F. McCoy. Artificial immune systems and aerial image segmentation. In *Proceedings of IEEE conference on Systems Man and Cybernetics*, pages 867–872, Orlando, Florida (USA), 1997.
- [103] G.J. Mendel. Versuche über Pflanzen-Hybriden, (Experiments in Plant Hybridisation). In *Verhandlungen des naturforschenden Vereines in Brünn*, (*Proceedings of the Natural History Society of Brünn*), 1866.
- [104] P. Mendes. Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Computer Applications in Biosciences*, 9(5):563–571, October 1993.
- [105] P. Mendes. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends in Biochemical Sciences (Elsevier)*, 22(9):361–363, September 1997.
- [106] P. Mendes and D.B. Kell. Computer simulation of biochemical kinetics. In *BioThermokinetics Of The Living Cell*, Eds. H.V. Westerhoff, J.L. Snoep, F.E. Sluse, J.E. Wijker and B.N. Kholodenko, *BioThermoKinetics Press*, pages 254–257, Amsterdam, 1996.
- [107] L. Michaelis and M.L. Menten. Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369, 1913.
- [108] C. J. Morton-Firth. *Stochastic simulation of cell signalling pathways*. PhD thesis, University of Cambridge, Cambridge (UK), September 1998.
- [109] C.J. Morton-Firth and D. Bray. Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology*, 192:117–128, 1998.
- [110] F. Mourgues, F. Devernay, and E. Coste-Manière. 3D reconstruction of the operating field for image overlay in 3D-endoscopic surgery. In *Proceedings of International Symposium on Augmented Reality*, October 2001.
- [111] National Center for Biotechnology Information. What is bioinformatics? <http://www.ncbi.nlm.nih.gov/Education/>, 2003.
- [112] National Institutes of Health. NIH working definition of bioinformatics and computational biology. <http://www.bisti.nih.gov/CompuBioDef.pdf>, 17 July 2000.
- [113] M. Oprea and S. Forrest. Simulated evolution of antibody gene libraries under pathogen selection. In *SMC'98, IEEE International Conference on Systems, Man and Cybernetics*, San Diego, CA (USA), 11-14 October 1998.

- [114] M. Parentoën, P. Reignier, and J. Tisseau. Put fuzzy cognitive maps to work in virtual worlds. In *10th International Conference on Fuzzy Systems, Fuzz-IEEE'01*, volume 1, pages 252–255, Melbourne (Australia), december 2–5 2001.
- [115] H.V.D. Parunak, R. Savit, and R.L. Riolo. Agent-based modeling vs. equation-based modeling : A case study and users' guide. In Springer Verlag, editor, *Proceedings of Multi-agent systems and Agent-based Simulation, MABS'98*, pages 10–25, Paris (France), July 4–6 1998.
- [116] A.S. Perelson and G. Weisbuch. Immunology for physicists. *Reviews in Modern Physics*, 69(4):1219–1267, 1997.
- [117] S. Picault and A. Collinot. La socialité : étude, enjeux et applications dans les systèmes multi-agents. In *6èmes Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents (JFIADSMA'98)*, Hermès, Eds. J.-P. Barthès, V. Chevrier, C. Brassac, pages 327–340, Pont-à-Mousson, France, 18–20 novembre 1998.
- [118] E. Promayon, J.L. Martiel, and P. Tracqui. *Physical-object-oriented 3D Simulations of Cell Deformations and Migration*, pages 125–140. Polymer and Cell Dynamics, Birkhäuser, Eds. W. Alt, M. Chaplain, M. Griebel and J. Lenz, Bonn (Germany), September 2002.
- [119] G. Querrec. *Modèle informatique du myélome et application en recherche médicale*. PhD thesis, Thèse de doctorat en Biologie de l'Université de Bretagne Occidentale, Brest (France), Soutenance prévue en 2005.
- [120] V. Raullet. *Prototypage interactif et collaboratif. Vers une architecture de communication pour une interactivité coopérante dynamique dans les environnements virtuels distribués*. PhD thesis, Thèse de doctorat en Informatique de l'Université de Bretagne Occidentale, Brest (France), 18 Juin 2003.
- [121] V. Raullet, A. Nédélec, and V. Rodin. A new API to the HLA declaration and object management services for runtime modifications. In *European Simulation Interoperability Workshop 2003 (03E-SIW-092)*, pages 600–607, Stockholm (Sweden), 16–19 June 2003. SISO.
- [122] V.N. Reddy, M.N. Liebman, and M.L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, 26(1):9–24, January 1996.
- [123] A. Ringvold, E. Anderssen, and I. Kjønniksen. Impact of the environment on the mammalian corneal epithelium. *Investigative Ophthalmology and Visual Science*, 44(1):10–15, January 2003.
- [124] V. Rodin. *Stéréovision Axiale: calibrage, appariement et reconstruction tridimensionnelle – Application en cœliochirurgie*. PhD thesis, Thèse de Docteur en Informatique de l'INP de Toulouse, Toulouse (France), 3 Octobre 1993.

- [125] V. Rodin, A. Ayache, and Dr N. Berreni. Aspects of computer vision in surgical endoscopy. In *Image Processing. SPIE's Medical Image VII*, volume 1898, pages 284–295, Newport Beach, CA (USA), 14–18 February 1993.
- [126] V. Rodin, A. Benzinou, A. Guillaud, P. Ballet, F. Harrouet, J. Tisseau, and J. Le Bihan. An immune oriented multi-agent system for biological image processing. *Pattern Recognition, Elsevier*, 37(4):631–645, April 2004.
- [127] V. Rodin and A. Nédélec. oRis: An agents communication language for distributed virtual environment. In *RO-MAN'99, 8th IEEE International Workshop on Robot and Human Interaction*, pages 41–46, Pisa (Italy), 27–29 September 1999.
- [128] V. Rodin, H. Troadec, H. De Pontual, A. Benzinou, J. Tisseau, and J. Le Bihan. Growth ring detection on fish otoliths by a graph construction. In *ICIP'96, IEEE International Conference on Image Processing*, volume 2, pages 685–688, Lausanne (Switzerland), 16–19 September 1996.
- [129] M. Roux-Rouquié. *Systemic modeling of biological functions : Application to the development of the GENINTER software dedicated to the compilation of interrelationships between genes and/or gene products*, pages 33–53. University of Geneva, 2000.
- [130] M. Roux-Rouquié, N. Caritey, L. Gaubert, and C. Rosenthal-Sabroux. Using the Unified Modelling Language (UML) to guide the systemic description of biological processes and systems. *BioSystems, Elsevier*, 75(1–3):3–14, July 2004.
- [131] M. Roux-Rouquié and J.L. Le Moigne. The systemic paradigm and its relevance to the modelling biological functions. *Comptes Rendus Biologies, Special Issue: Model driven Acquisition, Elsevier*, 325(4):419–430, April 2002.
- [132] I. Royston, J.A. Majda, S.M. Baird, B.L. Meserve, and J.C. Griffiths. Human T cell antigens defined by monoclonal antibodies: the 65,000- dalton antigen of T cells (T65) is also found on chronic lymphocytic leukemia cells bearing surface immunoglobulin . *Journal of Immunology*, 125:725–731, 1980.
- [133] K. Sachs, D. Gifford, T. Jaakkola, P. Sorger, and D.A. Lauffenburger. Bayesian network approach to cell signaling pathway Modeling. *Science's STKE*, http://stke.sciencemag.org/cgi/content/full/OC_sigtrans;2002/148/pe38, 148:pe38, September 2002.
- [134] J. Schaff and L.M. Loew. The virtual cell. In *Pacific Symposium on Biocomputing*, volume 4, pages 228–239, Hawaii (USA), 4–9 January 1999.
- [135] M. Schillo, H.J. Bürckert, K. Fischer, and M. Klusch. Towards a definition of robustness for market-style open multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents*, ACM Press, pages 75–76, Montreal, Quebec, Canada, 2001.

- [136] I.H. Segel. *Enzyme Kinetics Behavior And Analysis Of Rapid Equilibrium And Steady-State Enzyme Systems*, chapter 1, pages 1–17. Wiley Classics Library, 1993.
- [137] P.E. Seiden and F. Celada. A model for simulating cognate recognition and response in the immune system. *Journal of Theoretical Biology*, 158:329–357, 1992.
- [138] D.J. Smith, D.H. Ackley, S. Forrest, and A.S. Perelson. Using lazy evaluation to simulate realistic-size repertoires in models of the immune system. *Bulletin of Mathematical Biology*, 60:647–658, 1997.
- [139] D.J. Smith, S. Forrest, D.H. Ackley, and A.S. Perelson. Using lazy evaluation to simulate realistic-size repertoires in models of the immune system. *Bulletin of Mathematical Biology*, 60:647–658, 1998.
- [140] R.W. Smith. Computer Processing on Line Images: a Survey. *Pattern Recognition*, 20(1):7–15, 1987.
- [141] R.R. Sokal and P.H.A. Sneath. *Principles of numerical taxonomy*. W.H. Freeman & Co., San Francisco (USA), 1963.
- [142] K. Takahashi, N. Ishikawa, Y. Sadamoto, S. Sasamoto, H. Ohta, A. Shiozawa, F. Miyoshi, Y. Naito, Y. Nakayama, and M. Tomita. E-Cell 2: multi-platform E-Cell simulation system. *Bioinformatics*, 19(13):1727–1729, 2003.
- [143] K. Takahashi, K. Yugi, K. Hashimoto, Y. Yamada, C.J.F. Pickett, and M. Tomita. Computational challenges in cell simulation: A software engineering approach. *IEEE Intelligent systems*, 17(5):64–71, September/October 2002.
- [144] C. Tastemain. L'étude du "protéome" humain, nouveau Graal des biologistes. *Le Monde*, 25 janvier 2002.
- [145] J. Tisseau. *Réalité Virtuelle – autonomie in virtuo –*. PhD thesis, Habilitation à Diriger des Recherches, spécialité Informatique, Université de Rennes I, Rennes (France), 6 décembre 2001.
- [146] E.C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 42(55):189–208, 1948.
- [147] 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 3rd. E-Cell: software environment for whole cell simulation. *Bioinformatics*, 15(1):72–84, 1999.
- [148] H. Troadec, A. Benzinou, V. Rodin, and J. Le Bihan. Use of deformable template for two-dimensional growth ring detection of otoliths by digital image processing: Application to plaice (*pleuronectes platessa*) otoliths. *Fisheries Research, Elsevier*, 46(1–3):155–163, May 2000.

- [149] J.J. Tyson and B. Novak. Regulation of the eukariotic cell cycle: Molecular antagonism, hysteresis and irreversible transitions. *Journal of Theoretical Biology*, 210(2):249–263, 21 May 2001.
- [150] J. Vohradsky. Neural model of the genetic network. *Journal of Biological Chemistry*, 276(39):36168–36173, June 2001.
- [151] J.D. Watson and F.H.C. Crick. Molecular structure of Nucleic Acids. A structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738, 25 April 1953.
- [152] D. Weber, T. Pun, C. Roch, C. Rosu, and C. Pellegrini. Mélanie: Medical Electrophoresis Analysis, Interactive Expert System. In *SGAICO - Swiss Group of AI and Cognitive Sciences Annual Conference on Artificial Intelligence in Manufacturing, Assembly, and Robotics*, Bern, October 1988.
- [153] H.C. Welleman and F. Storbeck. *Automatic ageing of plaice (Pleuronectes plasseta) otoliths by means of image analysis*, pages 271–282. Secor, Dean and Campana (Eds.), Recent Developments in Fish Otolith Research, University of South Carolina Press, Columbia (USA), 1995.
- [154] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review (Cambridge University Press)*, 10(2):115–152, 1995.
- [155] V.I. Zarnitsina, A.V. Pokhilko, and F.I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. I. The model description. *Thrombosis Research*, 84(4):225–236, November 1996.
- [156] V.I. Zarnitsina, A.V. Pokhilko, and F.I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. II. The results. *Thrombosis Research*, 84(5):333–344, December 1996.
- [157] D. Ziou. Line Detection Using an Optimal IIR Filter. *Pattern Recognition*, 24(6):465–478, 1991.