



LICENCE INFORMATIQUE FONDAMENTALE

Rapport de stage

Auteur :
Fabrice LÉCUYER

Encadrant :
Olivier ROUX

**Introduction de données quantitatives
dans la modélisation qualitative
des régulations biologiques**

Juin-Juillet 2016

Remerciements

Tout d’abord, je tiens à remercier mon maître de stage, Olivier ROUX, qui a proposé ce stage et m’a guidé tout au long de ces six semaines. Il m’a fait découvrir le grand thème des réseaux de régulations biologiques et m’a orienté sur certains domaines auxquels l’équipe MeForBio s’intéresse particulièrement. D’un point de vue plus personnel, il m’a aussi soutenu lors de mes recherches de logement et d’activités à Nantes.

J’adresse également mes remerciements aux autres personnes de l’IRCCyN (*Institut de Recherche en Communications et Cybernétique de Nantes*) que j’ai pu rencontrer, et en particulier à l’équipe MeForBio qui m’a très bien accueilli, tant sur le plan professionnel que relationnel. Assister aux conférences et réunions d’équipe a été très formateur et m’a permis d’appréhender en profondeur les diverses activités des chercheurs.

Du côté de l’ENS de Lyon, je remercie le corps enseignant qui nous a énormément facilité la recherche et le choix des stages, ainsi que le personnel administratif qui nous a suivi au travers des fastidieuses démarches conventionnelles.

Enfin, je remercie ma famille pour le soutien logistique et moral tout au long de l’année, et mes amis qui ont rendu ce séjour à Nantes très agréable.

Sigles et acronymes

- IRCCyN : institut de recherche en communications et cybernétique de Nantes
- BRN : biological regulatory network, réseau de régulation biologique
- IG^[déf.2] : interaction graph, graphe d’interaction
- IGT^[déf.3] : interaction graph with thresholds, graphe d’interaction à seuils
- PH^[déf.5] : process hitting, réseau à frappes de processus

Table des matières

Remerciements	1
Liste des acronymes	1
Introduction	3
1 Réseaux de régulation biologique discrets	4
1.1 Discrétisation	4
1.2 Interactions géniques	5
1.3 Frappes de processus	7
2 Modélisation des systèmes hybrides	10
2.1 Systèmes hybrides	10
2.1.1 Définition	10
2.1.2 Exemple	11
2.2 Logiciel SpaceEx	12
2.2.1 Création de modèle	12
2.2.2 Configuration	13
2.2.3 Simulation exhaustive	13
2.3 Simulation et analyse de cas	14
2.3.1 Cas de la mucoviscidose	14
2.3.2 Programme de simplification	16
2.3.3 Cas de l’horloge circadienne	17
Conclusion	18
Références	19

Introduction

Les systèmes biologiques complexes sont le résultat d'interactions entre de très nombreuses entités différentes, qui se régulent mutuellement. Ainsi, la modélisation de ces systèmes entraîne des problèmes d'explosion combinatoire rendant les calculs coûteux, car le nombre d'états du système entier croît exponentiellement avec le nombre d'entités.

Le contournement de ce problème est un des grands thèmes sur lesquels travaille l'équipe MeForBio (Méthodes formelles pour la Bioinformatique). L'idée est de construire des modèles plus simples afin de leur appliquer des méthodes de preuve formelle, de simulation non-déterministe, ou encore de prédiction.

L'équipe s'intéresse notamment aux processus biologiques dus à l'expression de gènes qui s'inhibent et s'activent les uns les autres. Pour représenter ces réseaux de régulation biologique, il existe différents types de graphes comme les réseaux à frappe de processus et les graphes d'interaction. Ces modèles ont constitué le cœur de l'étude bibliographique et seront présentés dans la première partie.

L'utilisation de telles abstractions est très utile pour comprendre le fonctionnement ou dysfonctionnement de certains systèmes biologiques. Mais un des enjeux cruciaux est de réussir à introduire la temporalité dans ces modèles discrets : les gènes s'activant et s'inhibant progressivement sous l'action de leur environnement, il est souvent nécessaire de prendre en compte l'évolution continue des différentes variables. On aboutit donc à des modèles hybrides tenant compte à la fois des niveaux discrets et des grandeurs continues.

L'objet du stage était de modéliser ces systèmes hybrides et d'apporter des preuves d'accessibilité ou d'inaccessibilité de certains états. Dans ce but, l'équipe souhaitait utiliser SpaceEx, un logiciel de vérification des systèmes hybrides. Le travail sur ce logiciel sera décrit dans la deuxième partie, tant au niveau théorique avec la construction des modèles et la paramétrisation des modélisations, qu'au niveau pratique avec la réalisation de programmes en simplifiant l'utilisation.

1 Réseaux de régulation biologique discrets

L'étude de systèmes aussi complexes que les réseaux de gènes nécessite de faire certaines approximations. Ces abstractions permettent alors de créer des modèles mathématiques plus simples sur lesquels il est envisageable de faire des calculs et des preuves. Cette partie présente l'étude bibliographique réalisée sur les modèles discrets de représentation des réseaux de régulation biologiques.

1.1 Discrétisation

Pour analyser efficacement les interactions entre les protéines codées par différents gènes, la première idée est de discrétiser les valeurs prises par les composants. Par exemple, au lieu de considérer la valeur précise d'une concentration en protéine, on peut définir quelques niveaux discrets correspondant à l'activation ou l'inhibition de cette protéine. Comme le montre la figure 1, les transitions continues entre niveaux, souvent sigmoïdales, deviennent alors des paliers.

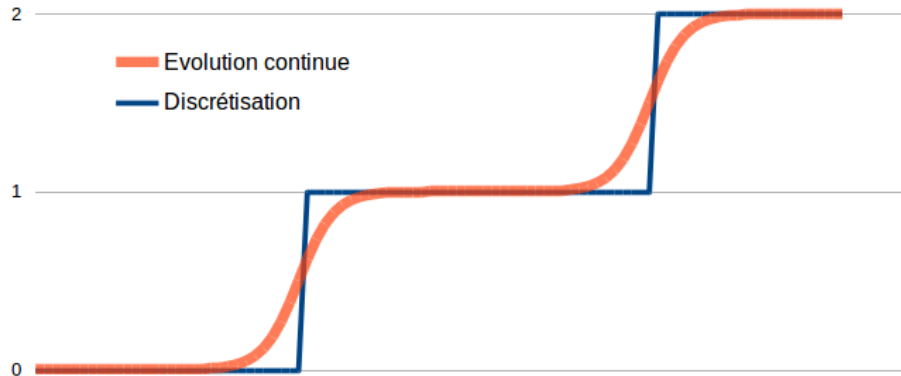


FIGURE 1 – Discrétisation des grandeurs continues

Cette simplification n'échappe certes pas à l'explosion combinatoire : pour un nombre n donné de gènes ayant chacun x niveaux d'expression, le système entier peut se trouver dans x^n états différents. Néanmoins, donner à chaque composant un niveau d'expression entier permet de se soustraire aux difficultés de représentation des nombres réels. D'autre part, les observations biologiques expérimentales ne distinguent nettement que quelques niveaux d'expression : on peut généralement se limiter à $x = 3$ niveaux (inactivé, faiblement activé, fortement activé). Le cas de deux niveaux (inactivé et activé)

se rapporte à de la logique binaire et a été étudié en profondeur^[18] ; il est donc intéressant de s'intéresser au cas de plusieurs paliers.

1.2 Interactions géniques

La représentation la plus simple des interactions entre n gènes consiste à inscrire dans un graphe orienté les influences positives (activation) et négatives (inhibition) qu'a chaque gène sur chaque autre. Plus rigoureusement, on définit ainsi un graphe d'interaction^[10] :

Définition 1 (Interaction Graph) *Un graphe d'interaction (IG) est un triplet (V, E_+, E_-) où*

- V est l'ensemble fini des n gènes
- $E_+ \subset V^2$ est l'ensemble des interactions positives
- $E_- \subset V^2$ est l'ensemble des interactions négatives

L'exemple de la figure 2 montre trois gènes fictifs et leurs influences. Ce graphe se lit en disant *le gène A active le B, B inhibe A, C est auto-activé etc.*

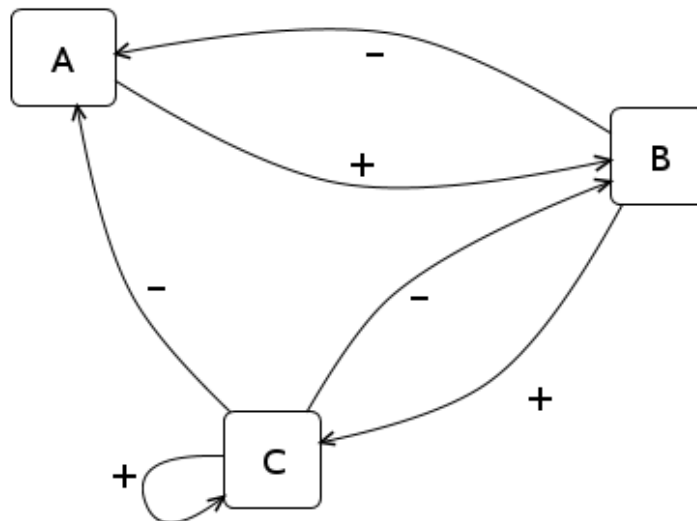


FIGURE 2 – Graphe d'interaction entre 3 gènes

D'un point de vue biologique, on remarque expérimentalement qu'un gène ne peut en activer d'autres que si sa concentration dépasse un certain seuil^[17]. Au-delà, l'activation est déclenchée ; en deçà, on considère qu'il y a inhibition.

Pour représenter ces seuils, il faut raffiner la définition des graphes d'interaction :

Définition 2 (Interaction Graph with Thresholds) *Un graphe d'interaction à seuils (IGT) est un graphe d'interaction (V, E_+, E_-) pondéré par une fonction $\omega : E \rightarrow \mathbb{N}^+$ (où $E = E_+ \cup E_-$). Pour $e = (u, v) \in E_+$ (resp. E_-), $\omega(e)$ représente le niveau discret à partir duquel le gène u active (resp. inhibe) le gène v .*

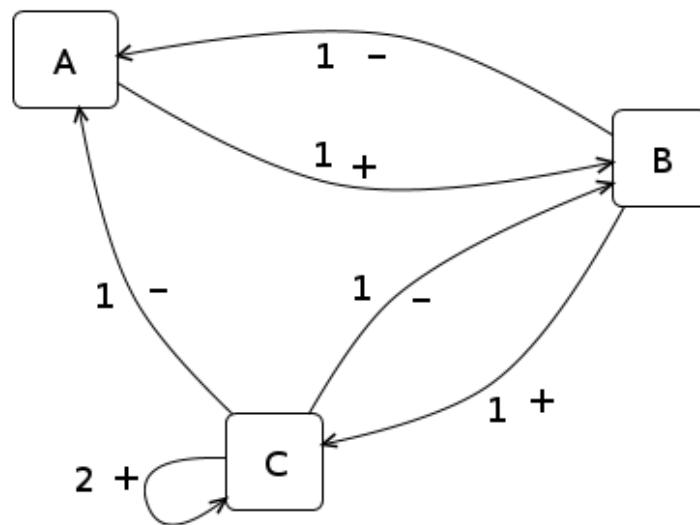


FIGURE 3 – Graphe d'interaction à seuils entre 3 gènes

Ces modèles permettent de renseigner les tendances qualitatives d'évolution. L'aspect quantitatif requiert un autre type d'information appelé paramètres logiques :

Définition 3 (Logical Parameter) *Pour un gène $v \in V$, on définit le paramètre logique $K_{v,Q}$ comme le niveau d'expression vers lequel tend le gène v lorsque l'ensemble des gènes qui l'activent est $Q \subset V$ (appelé ensemble des ressources de v).*

Le fonctionnement du système est intégralement défini par le graphe d'interaction et les paramètres logiques. Cependant, il y a souvent plusieurs paramètres possibles pour chaque gène et chaque ensemble de ressources. Pour choisir des paramètres cohérents^[3], l'équipe MeForBio se sert d'un logiciel appelé *SMBioNet*. Comme le montre le tableau 4, cet outil

Paramètre	Ressources		Valeurs	
	$Q =$	Possibles	Générées	Sélectionnées
K_A	$\{\}$	0	0	0
	$\{B\}$	0,1	0	0
	$\{C\}$	0,1	0,1	0
	$\{B,C\}$	1	1	1
K_B	$\{\}$	0	0	0
	$\{A\}$	0,1	0,1	0
	$\{C\}$	0,1	0,1	0
	$\{A,C\}$	1	1	1
K_C	$\{\}$	0	0	0
	$\{B\}$	0,1,2	2	2
	$\{C\}$	2	2	2
	$\{B,C\}$	2	2	2

FIGURE 4 – Tableau des paramètres logiques pour les 3 gènes

trouve tous les paramètres respectant les règles logiques (plus un gène a de ressources, plus il va s'activer) ; il détermine ensuite les paramètres satisfaisant des contraintes biologiques données (observation d'un cycle stable par exemple) ; enfin, il choisit une série de paramètres que l'on peut étudier.

L'ajout d'une telle table de paramètres logiques décrit entièrement le comportement du réseau de gènes : pour chaque état du système entier, on connaît son *état focal*, ou l'état vers lequel il va tendre. Néanmoins, ce tableau rend beaucoup plus compliquée l'assimilation visuelle du réseau biologique. En effet, le graphe d'interaction est facile à lire et donne immédiatement une idée du fonctionnement ; mais les paramètres logiques sont difficiles à interpréter rapidement. Pour cette raison, un autre formalisme a été mis en place.

1.3 Frappes de processus

Pour simplifier la représentation des réseaux de régulation génétiques (BRN) sans perdre de généralité, les chercheurs de l'équipe MeForBio ont introduit les *réseaux à frappe de processus*^[15]. L'idée est que chaque nœud du graphe représente un niveau d'expression d'un gène, et non un gène. On symbolise alors les influences par des flèches qui viennent *frapper* un niveau pour le faire *sauter* à un autre.

Définition 4 (Process Hitting) *Un réseau à frappe de processus (PH) est un triplet (Σ, L, \mathcal{H}) avec*

- Σ est un ensemble de catégories (ici, des gènes)
- $L = \prod_{a \in \Sigma} L_a$ est l'ensemble des états
- $\forall a \in \Sigma, L_a = \{a_1, \dots, a_{l_a}\}$ est l'ensemble des l_a processus (ou niveaux) propres à la catégorie a
- \mathcal{H} est un ensemble d'actions de forme $a_i \rightarrow b_j \uparrow b_k$ (lire a_i frappe b_j pour le faire sauter à b_k), où $a, b \in \Sigma, a_i \in L_a, b_j \neq b_k \in L_b$, et si $a = b$ alors $a_i = b_j$

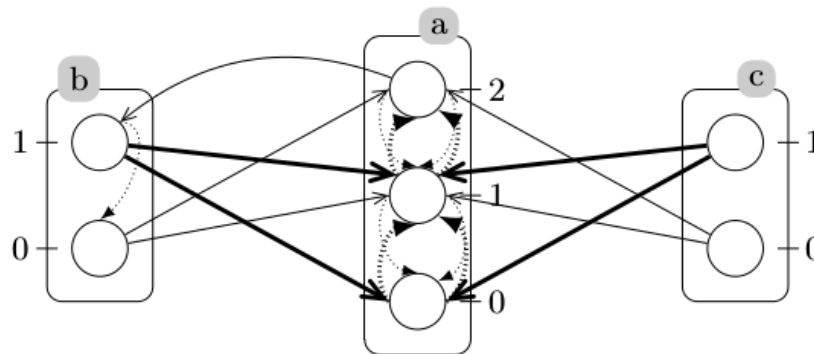


FIGURE 5 – Process hitting arbitraire de 3 gènes (a, b, c) ayant 2 ou 3 niveaux

Comme le montre la figure 5, on représente les catégories par des boîtes dans lesquelles des cercles symbolisent les processus. Les actions sont marquées par deux flèches : une pleine pour la frappe, une pointillée pour le saut. Ce nouveau modèle permet non seulement de comprendre visuellement le fonctionnement des interactions ; mais il permet aussi de savoir comment notre système peut évoluer à partir d'un état initial.

Un des résultats importants de l'équipe concerne l'inférence de graphes d'interaction à partir d'un réseau à frappes de processus^[7]. Ils ont en effet défini un ensemble des PH bien formés, desquels on peut déduire l'IGT associé et sa paramétrisation. Ces deux formats sont donc capables de représenter les mêmes réseaux de régulation biologique, mais le process hitting se démarque par sa simplicité : il n'est pas nécessaire d'y associer une liste des paramètres logiques possibles (liste qui s'agrandit très vite lorsque le réseau se complexifie). Par ailleurs, il est possible de rendre compte très simplement de la coopération entre plusieurs gènes (figure 6) par l'intermédiaire de multiplexes, de nouvelles catégories intermédiaires symbolisant l'action jointe de différents processus.

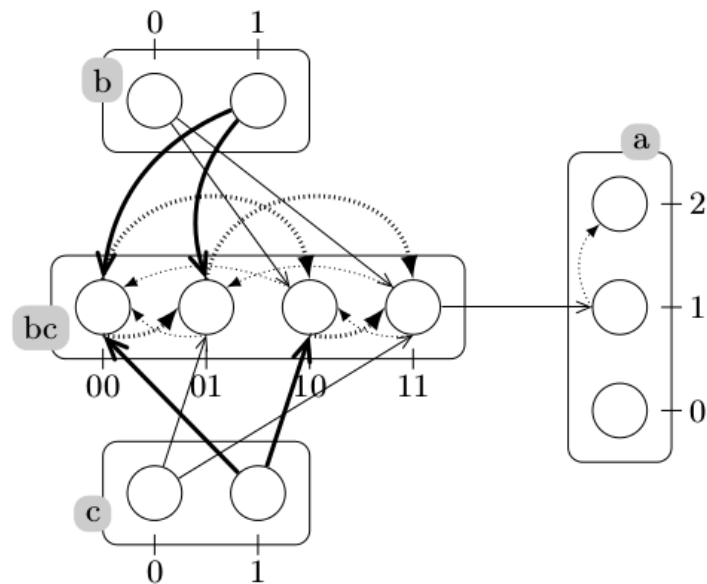


FIGURE 6 – Coopération des gènes b et c pour influencer a

Les modèles présentés ci-dessus ont l'avantage d'être simples et de permettre une étude formelle poussée. En particulier, les outils de vérification de modèles^[13] (model checking) sont capables de décider de l'accessibilité d'états donnés du système : à partir d'un état initial et en connaissance du BRN, la vérification sert à garantir théoriquement l'ensemble des états qui pourront être atteints. Ceci est une réponse utile pour les biologistes, car elle leur permet de comprendre comment peuvent être atteints un état malade ou un cycle stable^[14].

Toutefois, les niveaux d'expression discrets sont une lourde approximation : en réalité, l'évolution des concentrations est continue. Cela empêche par exemple que deux gènes changent de niveau au même instant précis. Pour parvenir à un modèle plus réaliste, il est donc nécessaire d'introduire des transitions entre niveaux, et donc une notion de temps. Naît alors un croisement entre discret et continu : un modèle hybride.

2 Modélisation des systèmes hybrides

Pour se rapprocher de la réalité biologique, les modèles doivent prendre en compte des variables continues ; mais pour rester abordables au niveau calculatoire, il faut pouvoir travailler par paliers. Le résultat, appelé système hybride^[4], sera défini en 2.1. Comme l'étude théorique de tels systèmes est beaucoup plus compliquée que celle des modèles discrets, il peut être plus intéressant de simuler leur comportement. Le logiciel SpaceEx utilisé à ces fins sera présenté en 2.2. Enfin, le travail de modélisation réalisé sera décrit en 2.2.2.

2.1 Systèmes hybrides

2.1.1 Définition

Intuitivement, un système hybride peut être vu comme un graphe dans lequel on se déplace de nœud en nœud en un mouvement discret, avec dans chaque nœud des équations différentielles régissant un mouvement continu. Par exemple, un modèle météorologique très simple pourrait avoir deux états discrets (jour et nuit) entre lesquels on alterne. La température (variable continue) serait croissante le jour et décroissante la nuit.

Plus formellement, on peut considérer la définition^[16] suivante :

Définition 5 (Système hybride) *Un système hybride est un 5-uplet $(L, \mathcal{V}, E, A, I)$ avec*

- L : un ensemble de nœuds appelés **emplacements**
- \mathcal{V} : un ensemble de **variables** réelles. Une **valuation** consiste en l'attribution d'une valeur à chaque variable. L'ensemble des valuations est noté V . L'ensemble des **états du système** est $\Sigma = L \times V$ (un emplacement et une valuation)
- $E \subseteq L \times \mathcal{P}(\mathcal{V}^2) \times L$: l'ensemble des **transitions** avec un élément typique $e = (\ell, a, \mu, \ell')$, transition entre les emplacements ℓ et ℓ' labelisé par a avec une **relation de transition** $\mu \subseteq \mathcal{V}^2$.
Dans l'état $(\ell, v) \in \Sigma$, la transition e est activée si et seulement si $\exists v', (v, v') \in \mu$. Alors l'état $(\ell', v') \in \Sigma$ est le **successeur** de (ℓ, v)
- A : la fonction associant à chaque emplacement ℓ un ensemble $A(\ell)$ d'**activités** (évolutions continues)
- I : la fonction associant à chaque emplacement ℓ un **invariant** $I(\ell) \subseteq V$ (un ensemble de valuations possibles)

2.1.2 Exemple

Pour mieux comprendre la définition très abstraite ci-dessus^[2], voici un exemple de thermostat modélisé par un système hybride. Le thermostat capte en permanence la température ambiante x et a pour objectif de la garder entre m et M . Quand le chauffage est éteint (état ℓ_0 de la figure 7), la température diminue en exponentielle décroissante vers zéro (on peut considérer que le zéro est la température extérieure); lorsqu'il est allumé (état ℓ_1), la température augmente vers h (paramètre lié à la puissance du chauffage). Par ailleurs, si le thermostat détecte une température inférieure à m (resp. supérieure à M), il va allumer (resp. éteindre) le chauffage.

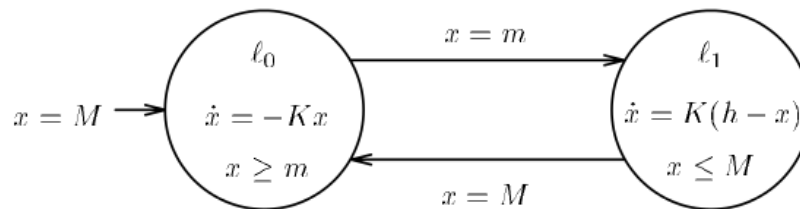


FIGURE 7 – Thermostat modélisé par un système hybride à 2 états

Avec un modèle aussi basique, le résultat est intuitif : la température oscille entre m et M , plus ou moins rapidement selon les paramètres. Mais pour des systèmes plus compliqués ayant de nombreux états, il est rare d'avoir un résultat prévisible sans outil de modélisation. C'est pourquoi des logiciels tels que SpaceEx existent.

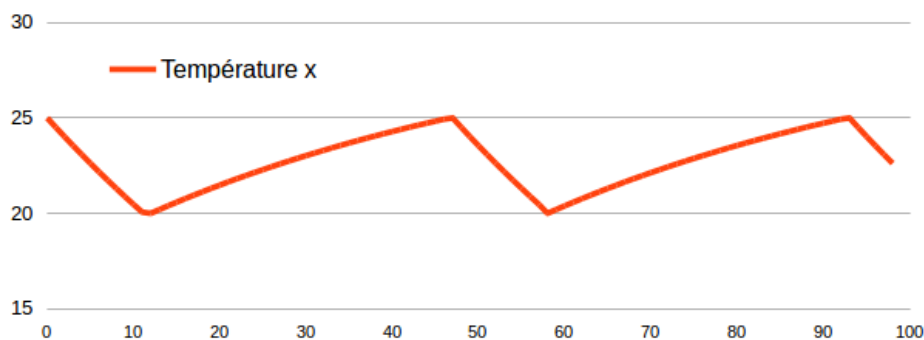


FIGURE 8 – Evolution de la température ($m = 20, M = 25, h = 30, K = \frac{1}{50}$)

2.2 Logiciel SpaceEx

SpaceEx est un outil dédié à la vérification des systèmes hybrides^[8] créé en 2010 par des membres du laboratoire Verimag, de l'Université Grenoble Alpes. Il propose une interface utilisateur pour définir le système et simuler son exécution.

2.2.1 Création de modèle

Une partie du logiciel est un éditeur de modèles de systèmes hybride. Elle se présente sous forme d'une application Java (voir figure 9) dans laquelle on place les différents composants du système : on dessine un automate où les états sont les emplacements du système hybride et les arêtes sont les transitions.

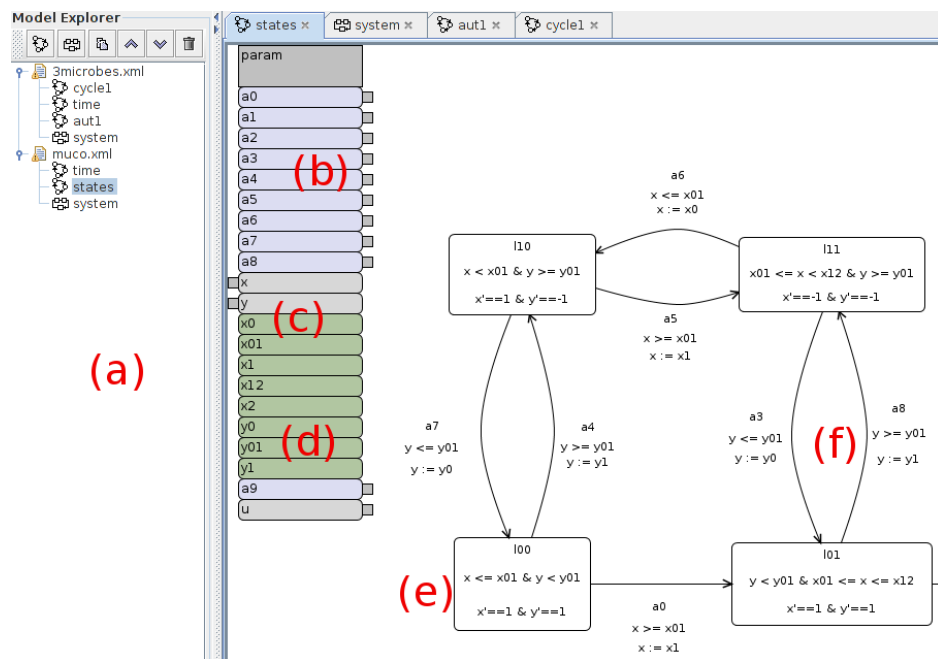


FIGURE 9 – Editeur de modèles de SpaceEx. (a) liste des projets et de leurs différents fichiers, (b) labels des transitions, (c) variables, (d) constantes internes, (e) emplacement avec équation d'invariance et équations différentielles de flot, (f) transitions avec garde et assignation

A chaque emplacement sont associés un invariant, qui est une condition à respecter dépendant des variables du modèle (par exemple, pour le thermostat, l'invariant lorsque le chauffage est allumé est $x \leq M$), et un flot, qui est

une liste d'équations différentielles auxquelles seront soumises les variables sur cet emplacement. Les transitions sont étiquetées par un label permettant de synchroniser plusieurs automates, une garde qui indique sous quelles conditions la transition peut être empruntée, et un assignement, qui permet de réaffecter certaines variables au franchissement de la transition.

A partir de cette schématisation, l'éditeur de modèle va créer un fichier *XML* reprenant toute l'architecture de l'automate. On obtient donc une description précise de notre système hybride, que l'on peut ensuite donner en entrée au simulateur. L'outil a l'avantage d'être très visuel, mais son utilisation devient assez fastidieuse pour de gros systèmes, d'où l'intérêt du programme présenté en partie 2.3.2.

2.2.2 Configuration

Une fois le modèle de système hybride créé, il faut le configurer, c'est-à-dire lui donner les conditions initiales en attribuant une valeur à chaque variable externe. Un grand avantage de SpaceEx est que l'on peut donner un intervalle de valeurs lors de cette paramétrisation. Par exemple, dans le cas du thermostat, on peut simuler l'évolution des températures avec une condition initiale $22 \leq x \leq 23$: le logiciel calculera alors la courbe d'évolution pour tout cet intervalle. Encore plus intéressant, on peut élargir les valeurs des constantes. Pour le thermostat, ce pourrait être $30 \leq h \leq 50$ (qui donnerait une simulation avec tout un panel de puissance de chauffage).

En plus des attributions de valeurs, la paramétrisation sert aussi à indiquer l'état initial de l'automate, le nombre d'exécutions, la précision des calculs, les variables à retourner, etc. Le tout est finalement stocké dans un fichier texte *.cfg*, que le programme 2.3.2 génère aussi.

2.2.3 Simulation exhaustive

Alors que de nombreux résultats d'accessibilité existent pour les modèles discrets présentés en partie 1, les systèmes hybrides sont beaucoup plus difficiles à étudier théoriquement car ils incorporent des mouvements continus. Les concepteurs de SpaceEx ont donc mis de côté la prédiction exacte pour se tourner vers une simulation exhaustive mais non minimale. Pour n'oublier aucun cas particulier des évolutions continues, le logiciel calcule à chaque étape une enveloppe convexe des états accessibles (figure 10), et se base dessus pour simuler l'étape suivante. Cette méthode permet de réduire significativement la complexité des calculs : au lieu d'opérer sur chaque état atteint, les calculs

prennent en compte toute l'enveloppe d'un coup. Ainsi, la vérification des systèmes hybrides consistant à déterminer si un état est accessible ou non peut être "passée à l'échelle" : SpaceEx est capable de résoudre ces problèmes pour de très grands systèmes^[9].

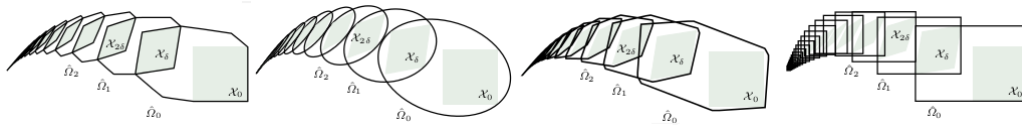


FIGURE 10 – Différents types d'enveloppes convexes utilisés dans SpaceEx (polyèdres, ellipsoïdes, zonotopes, fonctions de support)

2.3 Simulation et analyse de cas

L'équipe MeForBio avait déjà utilisé des outils de simulation et de vérification tels que HyTech^[11]. Mais les équations différentielles autorisées correspondaient seulement à des variations linéaires. Dans le but d'obtenir une meilleure approximation du monde vivant, l'équipe souhaitait donc que je découvre SpaceEx pour déterminer si ce logiciel pouvait devenir leur nouvel outil de travail.

2.3.1 Cas de la mucoviscidose

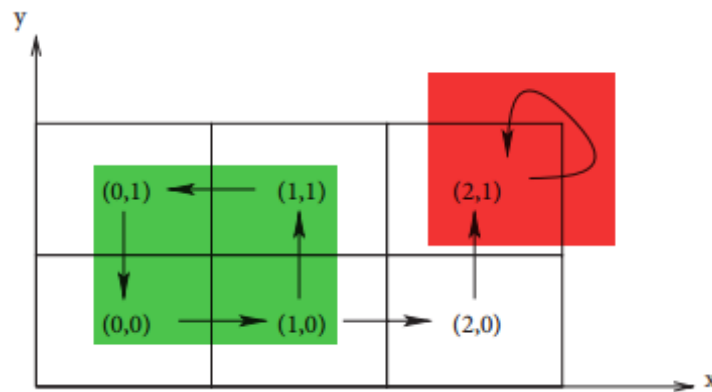


FIGURE 11 – Schéma de l'interaction des gènes x et y dans le cas de la mucoviscidose. Le cycle stable (vert) correspond à un fonctionnement normal, la maladie est due à l'attracteur (rouge).

Après avoir étudié quelques cas artificiels comme celui de la figure 2, je me suis intéressé à un cas bien connu : la mucoviscidose. Dans un modèle simplifié, cette maladie est due à deux gènes (notés ici x et y) ayant respectivement trois et deux niveaux d'activation. Chez un individu sain, un cycle se répète indéfiniment entre les états $(0, 0)$, $(1, 0)$, $(1, 1)$ et $(0, 1)$; en revanche, la maladie se déclare lorsque l'état $(2, 0)$ est atteint, car il mène irrémédiablement vers l'attracteur $(2, 1)$, sans sortie possible.

Cette maladie ayant déjà été étudiée dans le cas de systèmes hybrides temporisés^[1], j'ai pu trouver des valeurs utilisables pour les délais d'activation et d'inhibition. Dans un premier temps, j'ai utilisé des équations différentielles extrêmement simples, pour avoir des évolutions linéaires entre les niveaux d'activation. La figure 12 (gauche) montre le noyau d'invariance : pour toute condition initiale dans la zone verte, les états futurs y seront aussi, donc l'individu est sain et reste dans le cycle.

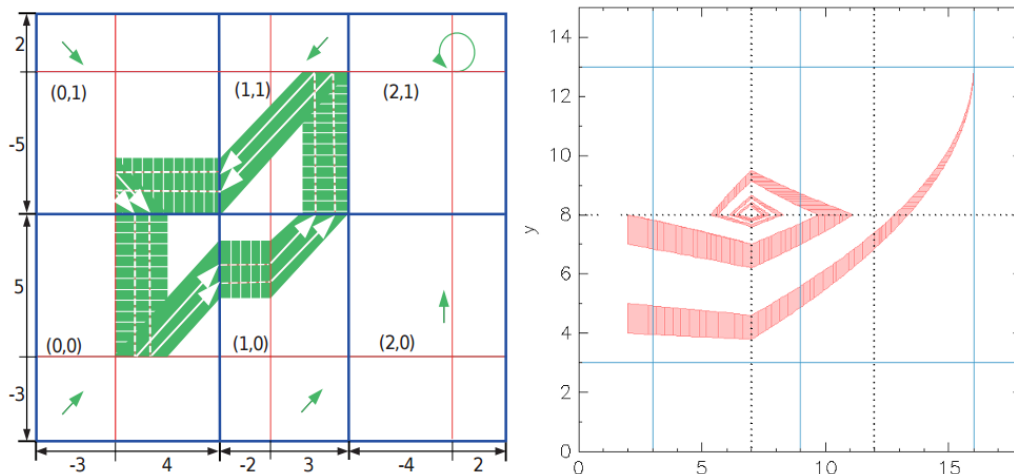


FIGURE 12 – Modèle de la mucoviscidose sous SpaceEx

Dans un second modèle, j'ai voulu me rapprocher de la forme sigmoïdale qui régit les transitions biologiques réelles. J'ai choisi des exponentielles croissantes ou décroissantes, d'une part parce qu'elles approximent correctement le démarrage progressif de l'activation ou de l'inhibition, d'autre part parce que l'équation correspondante est simple et peut être comprise par SpaceEx. La figure 13 montre ce qu'il a fallu entrer dans le créateur de modèle, et la figure 12 (droite) est le résultat de la simulation pour deux intervalles de conditions initiales. On observe que l'un des deux est attiré par l'attracteur $(2, 1)$, signe de la maladie ; en revanche, l'autre effectue une spirale centripète au lieu de se stabiliser dans un cycle.

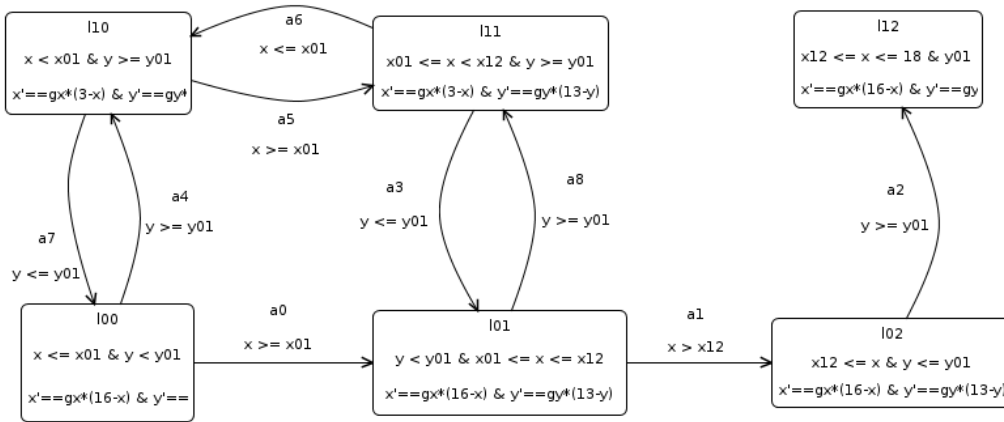


FIGURE 13 – Modèle de la mucoviscidose sous SpaceEx

Ces deux modèles différents montrent l'importance des types d'équation différentielle dans la simulation. N'utiliser que des transitions linéaires ne permet pas d'appréhender toutes les possibilités d'évolution. Néanmoins, l'approximation en exponentielles est loin d'être idéale, ce qui nous confronte de nouveau à un problème logiciel : SpaceEx ne prend pas en charge les équations différentielles à coefficients variables ou non linéaires.

2.3.2 Programme de simplification

La création de modèles par l'éditeur de SpaceEx étant assez longue, et en vue de modéliser des systèmes avec de nombreux gènes, j'ai entrepris de programmer un outil générant les fichiers *.xml* et *.cfg* à partir d'instructions simples. Ce programme a été rédigé en *C++* et s'utilise en console. Il fonctionne soit en mode interactif avec un système de questions-réponses permettant de définir entièrement le modèle, soit en prenant un fichier texte en entrée dans lequel il lit directement les instructions.

En donnant ligne par ligne les instructions (la figure 14 donne un aperçu), on définit le nom du modèle, la vitesse de son horloge, son nombre de gènes et les concentrations correspondant à chacun de leurs niveaux d'activation, les points focaux de chaque état, puis enfin la configuration initiale.

Ce programme m'a été très utile pour créer rapidement des modèles à partir de rien, mais aussi pour adapter les modèles précédemment construits. En effet, lorsque je suis passé des transitions linéaires aux transitions exponentielles, il a suffi d'adapter le programme en conséquence et de réécrire les instructions pour modifier entièrement les modèles.

muco 3	y 2 0	21
timer t	0 3 8	none
genes 2	8 13 15	config
x 3 0	20	1
0 3 7	00	1
7 9 12	21	export
12 16 18	01	quit

FIGURE 14 – Instructions pour le modèle de mucoviscidose

2.3.3 Cas de l’horloge circadienne

Un autre cas important que j’ai traité avec SpaceEx est celui de l’horloge circadienne. Cette horloge biologique que nous possédons tous est constituée d’un ensemble de protéines, gènes et hormones, qui régulent tout le fonctionnement de notre corps, et interagissent continûment pour s’activer et s’inhiber au bon moment. L’idée de cette étude rapide était d’observer le comportement de l’organisme après un désordre survenu dans l’horloge (par exemple, une luminosité contraire au rythme de sommeil, ou la souffrance d’un décalage horaire).

En considérant un modèle où deux protéines sont en interaction avec la luminosité ambiante, j’ai pu simuler l’impact de la variation saisonnière de la durée des journées sur les cycles biologiques. A cause d’une modélisation trop sommaire^[5] et du manque de données expérimentales cohérentes, je n’ai pas réussi à obtenir de résultats significatifs de ce côté. On a néanmoins pu observer des perturbations lorsque les jours sont très longs ou très courts : le cycle correspondant à l’éveil diurne est brisé, ce que pourrait se traduire dans la vie réelle par une irrépressible envie de dormir.

Conclusion

Ce stage de six semaines a commencé par une étude bibliographique des réseaux de régulation biologiques et des modèles les plus fréquemment utilisés. D'un point de vue théorique, j'y ai découvert les graphes d'interaction et les réseaux à frappe de processus; mais ces lectures m'ont aussi beaucoup éclairé sur la biologie elle-même, sur cette fabuleuse capacité qu'ont les gènes de s'influencer mutuellement^[12]. Le premier critère pour le choix de mon stage était de pouvoir appliquer mes connaissances théoriques au monde du vivant, ce qui a été possible grâce à toutes les notions biologiques qui sous-tendent ces articles.

Simplifier des problèmes bien connus en biologie a permis de les traduire en modèles de systèmes hybride, avant d'utiliser le SpaceEx pour simuler leur comportement dans le temps. Après avoir utilisé ce logiciel sur de nombreux cas, notamment l'horloge circadienne et la mucoviscidose, mon maître de stage et moi avons conclu qu'il ne permettrait pas de faire des vérifications formelles très poussées, car ces outils théoriques sont cachés dans le calcul des simulations. Il s'est toutefois montré d'une aide précieuse pour comprendre les interactions géniques et les dérèglements qui peuvent survenir au sein des cycles bien huilés qui nous permettent de vivre.

Pour la fin de mon stage, Olivier ROUX m'a présenté les séries temporelles : ce sont des données biologiques issues d'expériences où la concentration en de nombreuses protéines a été mesurée à intervalles de temps réguliers. Grâce à ces données, on peut observer les effets qu'ont les gènes les uns sur les autres, et essayer de déduire les graphes d'interaction. C'est un autre grand thème sur lequel travaille l'équipe MeForBio, et j'ai eu la chance d'assister à une présentation de leur travail sur le sujet.

Au cours de mon passage à l'IRCCyN, j'ai également vu une conférence sur la détection de structures communautaires dans les graphes, et un rapport sur l'étude de la génomique des océans par *Tara Oceans*. Ces ouvertures thématiques ont accompli de remplir la promesse du monde de la recherche : travailler sur un sujet qui nous passionne tout en s'ouvrant à de nombreux autres domaines.

Références

- [1] J. AHMAD, G. BERNOT, D. LIME, O. ROUX ET J.-P. COMET, *Hybrid modelling and dynamical analysis of gene regulatory networks with delays*, ComPlexUs, (2007).
- [2] R. ALUR, C. COURCOUBETIS, N. HALBWACHS, T. HENZINGER, P.-H. HO, X. NICOLLIN, A. OLIVERO, J. SIFAKIS ET S. YOVINE, *The algorithmic analysis of hybrid systems*, Theoretical Computer Science, 138 (1995), p. 3 – 34.
- [3] G. BERNOTA, J.-P. COMETA, A. RICHARDA ET J. GUESPINB, *Application of formal methods to biological regulatory networks : extending thomas asynchronous logical approach with temporal logic*, Journal of Theoretical Biology, (2004).
- [4] Z. BIBI, J. AHMAD, A. SIDDIQA, R. Z. PARACHA, T. SAEED, I. A. SHEIKH ET O. ROUX, *Formal analysis of the regulatory network of mTOR pathway in survival mechanisms*, PLOS.
- [5] J.-P. COMETA, G. BERNOTA, A. DASB, F. DIENERB, C. MASSOTA ET A. CESSIEUX, *Simplified models for the mammalian circadian clock*, Computational Systems-Biology and Bioinformatics, (2012).
- [6] L. F. DEL CERRO ET K. INOUE, *Logical Modeling of Biological Systems*, 2014.
- [7] M. FOLSCHETTE, L. PAULEVÉ, K. INOUE, M. MAGNIN ET O. ROUX, *Concretizing the process hitting into biological regulatory networks*, HAL archives ouvertes, (2012).
- [8] G. FREHSE, *An introduction to hybrid automata, numerical simulation and reachability analysis*, (2011).
- [9] G. FREHSE, C. L. GUERNIC, A. DONZÉ, S. COTTON, R. RAY, O. LEBTEL, R. RIPADO, A. GIRARD, T. DANG ET O. MALER, *Spaceex : Scalable verification of hybrid systems*, (2011).
- [10] S. HAYAT, K. TAREEN, J. AHMAD ET O. ROUX, *Parametric linear hybrid automata for complex environmental systems modeling*, Frontiers in Environmental Science, (2015).
- [11] T. A. HENZINGER, P.-H. HO ET H. WONG-TOI, *Hytech : A model checker for hybrid systems*, (1997).
- [12] P. MONHONVAL ET F. LOTSTRA, *Epigenetics of early interactions, depression and schizophrenia*, Rev Med Brux, (2012).
- [13] L. PAULEVÉ, C. CHANCELLOR, M. FOLSCHETTE, M. MAGNIN ET O. ROUX, *Analyzing Large Network Dynamics with Process Hitting*, 2014.

-
- [14] L. PAULEVÉ, M. MAGNIN ET O. ROUX, *Tuning temporal features within the stochastic π -calculus*, IEEE Transactions on Software Engineering, (2010).
 - [15] —, *Refining dynamics of gene regulatory networks in a stochastic π -calculus framework*, Transactions on Computational Systems Biology, (2011).
 - [16] —, *Static analysis of biological regulatory networks dynamics using abstract interpretation*, Math. Struct. in Comp. Science, 22 (2012).
 - [17] Z. QURESHI, J. AHMAD, A. ALI ET O. ROUX, *Dynamic analysis of the hedgehog signalling pathway*, Computational Biology and Chemistry, (2016).
 - [18] R. THOMAS, *Regulatory networks seen as asynchronous automata : a logical description*, Journal of Theoretical Biology, (1991).