LABORATOIRE DE MÉTHODES INFORMATIQUES

Journée thématique RIAMS « Réseaux d'interaction : analyse, modélisation et simulation »

Jean-Paul COMET et Sandrine VIAL

LaMI
Tour Évry 2
523 place des terrasses de l'Agora
91000 Évry Cedex, France
Email: {Jean-Paul.Comet,Sandrine.Vial}@lami.univ-evry.fr

— Journée Thématique RIAMS du 30 novembre 2005 —

RAPPORT DE RECHERCHE

N^{o} 121–2005

Novembre 2005

LaMI, UMR 8042 CNRS - Université d'Évry Val d'Essonne, Genopole
Tour Évry 2, 523 place des terrasses de l'Agora
91000 Évry Cedex, France

Jean-Paul COMET et Sandrine VIAL
Journée thématique RIAMS « Réseaux d'interaction : analyse, modélisation et simulation »
56 p.

Journée thématique RIAMS
« Réseaux d'interaction : analyse, modélisation et simulation »

Jean-Paul COMET et Sandrine VIAL

Résumé

L'objectif de cette journée thématique francophone sur l'analyse, la modélisation et la simulation des réseaux d'interaction dans le cadre de la biologie est de réunir toute la comunauté scientifique souhaitant partager ses compétences propres pour la compréhension des réseaux d'interaction biologiques. Cette première rencontre est principalement, mais non exclusivement axée sur les thèmes : analyse des systèmes d'interaction biologiques, les grands réseaux d'interaction, l'évolution des réseaux, la modélisation de systèmes biologiques, la simulation de tels systèmes.
Cette journée a été organisée grâce au soutien de l'ACI VICANNE.

Journée Thématique RIAMS « réseaux d'interaction : analyse, modélisation et simulation »

L'objectif de cette journée thématique francophone sur l'analyse, la modélisation et la simulation des réseaux d'interaction dans le cadre de la biologie est de réunir toute la comunauté scientifique souhaitant partager ses compétences propres pour la compréhension des réseaux d'interaction biologiques. Cette première rencontre est principalement, mais non exclusivement axée sur les thèmes :

- analyse des systèmes d'interaction biologiques,
- les grands réseaux d'interaction,
- l'évolution des réseaux,
- la modélisation de systèmes biologiques,
- la simulation de tels systèmes.

Cette journée a été organisée grâce au soutien de l'ACI[1] VICANNE : Modélisation dynamique et simulation des systèmes biologiques projet financé par le Ministère délégué à la Recherche et aux Nouvelles Technologies, Direction de la Recherche. Pour plus de renseignements sur les activités de l'ACI VICANNE, veuillez consulter :

http ://vicanne.inrialpes.fr/ .

Les articles présentés à RIAMS ont été sélectionnés par les membres du comité scientifique à partir d'un résumé de 6 pages.

Comité scientifique

[1] Action Concertée Incitative IMPbio

Comité d'organisation

Jean-Paul Comet
LaMI UMR 8042 du CNRS,
Université d'Evry Val d'Essonne
91000 Evry Cedex
Jean-Paul.Comet@lami.univ-evry.fr

Sandrine Vial
LaMI UMR 8042 du CNRS,
Université d'Evry Val d'Essonne
91000 Evry Cedex
Sandrine.Vial@lami.univ-evry.fr

Les organisateurs souhaitent remercier spécialement Olivier Gandrillon ainsi que les autres organisateurs de IPG 2005, pour leur soutien. Catherine Meignen, génopole recherche, a aussi contribué à la bonne organisation de cette journée.

Objectifs de la journée thématique

Comprendre les réseaux d'interactions qui effectuent des calculs au sein même de la cellule (les réseaux de transduction du signal, les réseaux génétiques, les réseaux protéiques), est devenu un problème central pour la biologie moléculaire. Ces réseaux offrent la possibilité d'essayer de comprendre le comportement collectif de systèmes d'interactions, dont les entités coopèrent avec une précision remarquable sous des contraintes biologiques très fortes. Le but est de découvrir les principes généraux sousjacents qui gouvernent leur fonctionnement.

Cette journée satellite d'IPG 2005, se focalisera sur différents aspects des réseaux de régulation. L'analyse de systèmes biologiques particuliers comportant des entités en interaction permet de mettre en évidence certaines règles communes du fonctionnement de tels réseaux. La modélisation de tels réseaux offre une approche pour comprendre la complexité des systèmes biologiques. Récemment, la simulation a permis d'appréhender, de manière concluante, des processus biologiques complexes comme les voies métaboliques, les réseaux de régulation génétique et la transduction du signal. Ces modèles ont non seulement permis de générer des hypothèses vérifiables expérimentalement, mais aussi ils ont permis d'avoir un regard nouveau sur le comportement des systèmes biologiques complexes.

Les thématiques principales de cette journée porteront entre autres sur :

- l'analyse des systèmes d'interaction,
- les grands réseaux d'interaction,
- l'évolution des réseaux,
- la modélisation de systèmes biologiques,
- la simulation de tels systèmes.

Table des matières

Exposés invités

Articles RIAMS

Sur les relations entre la topologie des réseaux d'interactions protéiques et la physiologie cellulaire

Marie-Claude Marsolier-Kergoat [1]

[1] Service de Biochimie et de Génétique Moléculaire, CEA/Saclay, 91191 Gif-sur-Yvette, France

Abstract

Le fonctionnement cellulaire fait intervenir des interactions physiques entre différents types de molécules. On peut définir le réseau des interactions protéiques d'une cellule à partir de l'ensemble des liens non-orientés reliant des protéines pour lesquelles la preuve expérimentale d'une interaction physique a été établie. Des études systématiques à grande échelle ont depuis quelques années permis une première description des réseaux d'interactions protéiques dans différents organismes dont l'eucaryote modèle *Saccharomyces cerevisiae*. De nombreuses analyses se sont par la suite attachées à démontrer des corrélations entre la topologie de ces réseaux et le fonctionnement cellulaire, tant à un niveau local que global.

A un niveau local, il a été proposé par exemple que certaines caractéristiques topologiques des protéines dans ces réseaux d'interactions étaient corrélées à une caractéristique physiologique simple, leur essentialité (une protéine est dite essentielle lorsque la délétion du gène correspondant est létale dans des conditions de croissance considérées comme optimales). Les protéines essentielles et non-essentielles d'une cellule ont ainsi été décrites comme différant significativement par le nombre de leurs interactions, le nombre moyen des interactions de leurs plus proches voisins ou leur coefficient de regroupement ('clustering coefficient'). Cependant nous avons montré que ces conclusions étaient fondées sur des données dont de nombreux biais n'avaient pas été pris en compte et qu'en réalité l'essentialité d'une protéine était très faiblement corrélée au nombre de ses partenaires, et non corrélée à tous les autres paramètres topologiques étudiés.

A un niveau global, les réseaux d'interactions protéiques présentent, comme beaucoup d'autres réseaux technologiques ou sociaux, une distribution large du nombre des interactants, avec une majorité de protéines peu connectées (peu d'interactants) et une minorité de protéines très connectées. Il a été proposé que cette caractéristique pourrait contribuer à la robustesse des cellules aux mutations, en particulier au fait que les protéines essentielles d'un organisme semblent systématiquement minoritaires. Cette proposition s'appuyait sur le lien entre l'essentialité des protéines et leur nombre d'interactants, et disparaît donc maintenant que cette prémisse est remise en question. La signification physiologique de la topologie globale des réseaux d'interactions protéiques apparaît donc incertaine, d'autant plus que certains modèles physiques permettent d'obtenir des topologies comparables pour des réseaux protéiques fondés sur des interactions non-spécifiques.

Constraint-based models of metabolism : studying the global metabolism of Acinetobacter ADP1

Vincent Schächter[1]

[1] Genoscope, 2 rue Gaston Crémieux, 91000 Evry, France.

Abstract

The availability of complete genomes, of the first metabolomics datasets, and the rise in expectations toward the explanatory and predictive powers of biological network models has given a new impulse to the study of metabolism, thought by many to be a well-understood field a few years ago. One important aim is metabolic reconstruction : comparative methods confirm that procaryote metabolism, in particular, exhibits huge variability, and there are significant gaps in our knowledge of the best known network of metabolic reactions, that of E.coli. Another aim is to better understand the global metabolic behaviour of a (bacterial) cell, seen as a biochemical transformation machine interacting with its environment. Classical models based on sets of differential equations have limited applicability here, both because of the rarity of experimentally determined kinetic parameters, and because of the size of the networks involved. Constraint-based modeling of metabolism is a semi-formal framework dedicated to the modelling of metabolic processes at steady state, i.e. a global state of the metabolic network is defined as a distribution of fluxes within the network reactions. It emerged in the 90s as a radical simplification of kinetic models and was developed to allow tractable modelling of genome-scale metabolic networks. During the last 4 years, it has been applied successfully to a variety of reconstruction, structural analyses and predictive tasks on large metabolic networks in bacteria and yeast, yielding interesting new biological insights. The steady-state hypothesis positions the framework at a level of detail intermediate between description of static network structure and representation of network dynamics. More importantly, it is designed to represent incomplete information, yet to allow some prediction of metabolic behavior. The focus, rather than being on fully instantiated descriptions of the system?s behavior, is on sets of such descriptions, i.e. sets of flux distributions compatible with a set of constraints representing the current knowledge on the structure of the network, on thermodynamic and kinetic parameters, and on input/output relationships of the network with its environment. This solution set can be refined incrementally as new constraints are added, ensuring some robustness in structural analyses and metabolic behaviour predictions with respect to modifications of the model.

First, we will introduce the steady-state metabolic flux modeling framework and its constraint-based version. We will then focus on the use of this framework within the context of the ?Metabolic Thesaurus? project, an experimental effort aimed at understanding the metabolism of, Acinetobacter ADP1 (a versatile, highly competent, strictly aerobic, gram-negative soil bacterium with biodegradative capabilities) using large-scale phenotypic and biochemical data. We will describe the reconstruction of a global metabolic model of Acinetobacter ADP1, up to a point where good agreement was reached between model predictions and a significant set of experimental data on single-deletion mutant growth phenotypes. Finally, we will sketch ongoing work on two topics : the use of global phenotypic profiles to help with model refinement, and a theoretical study on the variability of flux-coupling patterns across a set of metabolic environments.

Modélisation et Analyse des Réseaux de Régulation Génétique par des Systèmes d'Attracteurs

Michaël Adélaïde

Université d'Oldenburg,
OFFIS, Escherweg 2a, 26121 Oldenburg, Allemagne
michael.adelaide@informatik.uni-oldenburg.de

1 Introduction

Les réseaux de régulation génétique sont les mécanismes mis en œuvre par les cellules pour produire les protéines et contrôler leurs concentrations. Dans un modèle simplifié de la biologie moléculaire, la partie fonctionnelle de l'ADN d'une cellule se divise en gènes et et en sites de fixation. Chaque gène produit une protéine spécifique. La vitesse de production d'un gène dépend alors de l'état des sites de fixation associés à ce gène. Les protéines peuvent se fixer sur les sites ou se détacher suivant leurs concentrations.

D'un point de vue formel, un réseau de régulation est un système hybride paramétré : les variables sont les concentrations des protéines et les constantes (seuils et intensités de production) ne sont pas connues à priori : ce sont donc des paramètres. Les principaux challenges posés par l'analyse de ces réseaux sont des problèmes de l'analyse des systèmes dynamiques à savoir :

1. *états stationnaires :* trouver les états stationnaires et leurs bassins d'attractions ;
2. *accessibilité :* est-il possible de se rendre depuis région donnée (de l'espace des concentrations) vers une autre région ?
3. *reverse engineering :* quelles sont les valeurs des paramètres qui assurent un comportement donné du système ?

Cet article reprend les travaux réalisés avec Grégoire Sutre et présentés dans [1] dans le cadre des systèmes symboliques [7]. On définit un cadre complet : les réseaux de régulation génétique *RRG* sont modélisés par deux hyperarcs : un pour la production et un autre pour la dégradation. La sémantique d'un RRG est donnée par un système différentiel linéaire par morceaux (*SDLM*). Lorsque les dégradations sont uniformes, un SDLM est équivalent à un système d'attracteurs (*SA*). Les systèmes d'attracteurs sont des systèmes symboliques [7]. On peut décrire des semi-algorithmes [6] pour répondre aux trois problèmes définis plus haut.

Le présent document s'organise comme suit. Dans la section 2, on présente les systèmes différentiels linéaires par morceaux (*SDLM*) et les systèmes d'attracteurs (*SA*) qui sont les deux modèles qu'on utilise pour étudier les systèmes dynamiques. Dans la section 3, on définit les réseaux de régulation génétiques (*RRG*) : la sémantique d'un RRG est donnée par un SDLM. Enfin, dans la section 4, un exemple est utilisé pour illustrer l'analyse paramétrique.

2 Systèmes différentiels Linéaires par Morceaux et Systèmes d'Attracteurs

Pour un ensemble E, $\mathcal{P}(E)$ désigne l'*ensemble des parties* de E, $\mathcal{P}_f(E)$ désigne l'*ensemble des parties finies* de E. Soit E un sous ensemble de $\mathbb{R}^V$. L'*adhérence* de E (i.e. le plus petit fermé de $\mathbb{R}^V$ contenant E) est notée $\overline{E}$; l'*intérieur* de E, noté $\overset{\circ}{E}$ est le plus grand ouvert inclus dans E et la frontière de E est $fr(E) = \overline{E} \setminus \overset{\circ}{E}$. Si V est un ensemble fini, on note $\mathbb{R}^V$ l'espace vectoriel des fonctions $\overrightarrow{x}$ de V dans $\mathbb{R}$. Soient deux vecteurs $\overrightarrow{x}$ et $\overrightarrow{y}$ de $\mathbb{R}^V$. On note $\overrightarrow{x} \circ \overrightarrow{y}$ le *produit par composantes* de $\overrightarrow{x}$ et de $\overrightarrow{y}$, i.e. le vecteur $\overrightarrow{z}$ tel que $\forall v \in V : \overrightarrow{z}(v) = \overrightarrow{x}(v).\overrightarrow{y}(v)$.

L'*enveloppe convexe* de E (i.e. le plus petit ensemble convexe fermé contenant E) est notée $Hull(E)$. Soient E un ensemble convexe de $\mathbb{R}^V$ et $\overrightarrow{x} \in \mathbb{R}^V$, on note :

$$Light(\overrightarrow{x}, E) = \{\overrightarrow{y} \in \mathbb{R}^V \mid \exists \lambda \in]0,1] : \exists \overrightarrow{z} \in E : \overrightarrow{y} = \lambda \overrightarrow{x} + (1-\lambda)\overrightarrow{z}\}$$

Ainsi, $Light(\overrightarrow{x}, E) \setminus \{\overrightarrow{x}\}$ représente l'ensemble des points situés sur un segment ouvert dont l'une extrémité est $\overrightarrow{x}$ et l'autre un élément $\overrightarrow{z}$ de E. $\overrightarrow{x}$ joue alors le rôle de "source lumineuse" et $\overrightarrow{z}$, celui d'"attracteur" des rayons issus de $\overrightarrow{x}$. Par exemple, si $\overrightarrow{x} \in \overset{\circ}{E}$, alors $Light(\overrightarrow{x}, E) = \overset{\circ}{E}$. Si E est un polytope (polyèdre convexe borné) et $\overrightarrow{x} \notin E$, alors $Hull(\{\overrightarrow{x}\} \cup E)$ est un polytope et $Light(\overrightarrow{x}, E) = Hull(\{\overrightarrow{x}\} \cup E) \setminus F$ où F est l'ensemble des faces de $\{Hull(\overrightarrow{x}\} \cup E)$ ne contenant pas $\overrightarrow{x}$.

2.1 Systèmes Différentiels Linéraires par Morceaux

Définition 1 (Système Différentiel Linéraire par Morceaux) *Un système différentiel linéraire par morceaux est un tuple* $\mathsf{L} = \langle V, \Theta, \overrightarrow{\kappa}, \overrightarrow{\gamma} \rangle$ *tel que :*

1. V *est un ensemble fini de* composants *;*
2. $\Theta : V \to \mathcal{P}_f(\mathbb{R})$ *associe à chaque composant un ensemble fini de* seuils *;* Θ *permet de définir une partition* D *de* $\mathbb{R}^V$*, appelée* quadrillage *et définie comme suit :*
   (a) *le quadrillage de* $\mathbb{R}^{\{v\}}$ *généré par* $\Theta|_{\{v\}}$ *est la partition de* $\mathbb{R}$ *composée des points* $\{\Theta_i(v)\}$ *pour* $1 \leq i \leq n_v$ *et des intervalles ouverts* $]\Theta_i(v), \Theta_{i+1}(v)[$ *pour* $0 \leq i \leq n_v + 1$*, en posant* $\Theta_0(v) = -\infty$ *et* $\Theta_{n(v)+1}(v) = +\infty$ *;*
   (b) *le quadrillage de* $\mathbb{R}^V$ *généré par* Θ *est la partition de* $\mathbb{R}^V$ *dont les cellules sont de la forme* $c_{v_1} \times ...c_{v_{|V|}}$ *où* c_{v_i} *est une cellule du quadrillage pour* $\Theta|_{\{v_i\}}$ *;*

   Les ensembles ouverts de D *sont appelés les* boites*, l'ensemble des boites de* D *est noté* $Box(\mathsf{D})$ *;*
3. $\overrightarrow{\kappa} : Box(\mathsf{D}) \to \mathbb{R}^V$ *;*
4. $\overrightarrow{\gamma} : Box(\mathsf{D}) \to \mathbb{R}^V_{\geq 0}$ *est telle que* $\forall B \in Box(\mathsf{D}) : \ \forall v \in V : \ \overrightarrow{\gamma}(B)(v) \neq 0$.

Les cellules de D qui ne sont pas des boites sont appelées des *murs*. Une cellule est également appeée *zone*. L'*ensemble des zones* de D est noté $Dom(\mathsf{D})$. L'ensemble des boites *adjointes* à une zone Z est l'ensemble $adj(Z)$ des boites B telles que $\overline{Z} \subseteq \overline{B}$. Pour une zone Z, $\overline{Z} = \bigcap_{B \in adj(Z)} \overline{B}$. De plus, si B est une boite alors $adj(B) = \{B\}$.

La sémantique de L est donné par le système de transitions $\mathsf{S}_\mathsf{L} = \langle \mathbb{R}^V, \underset{\mathsf{L}}{\rightarrow} \rangle$ tel que $\overrightarrow{x} \underset{\mathsf{L}}{\rightarrow} \overrightarrow{y}$ ssi il existe $\delta \geq 0$, $Z \in Dom(\mathsf{D})$ et $\Gamma : [0, \delta] \to \overline{Z}$ tels que :

1. $\overrightarrow{x} \in \overline{Z}$, $\overrightarrow{y} \in \overline{Z}$, $\Gamma\,(]0, \delta[) \subseteq Z$;
2. l'ensemble $\{\tau \in]0, \delta[\mid \frac{d\Gamma}{dt}(\tau) \notin Hull(\overrightarrow{\kappa}[B] - \overrightarrow{\gamma}[B] \circ \overrightarrow{x}) \mid B \in adj(Z)\}$ est négligeable (i.e. inclus dans un ensemble de mesure nulle au sens de Lebesgue).

Ce modèle est utilisé dans [5, 2, 3, 1]. On peut remarquer qu'il n'est pas nécessaire de définir $\overrightarrow{\kappa}$ et $\overrightarrow{\gamma}$ dans toutes les boites. Si $\mathsf{D}_1 \subseteq \mathsf{D}$ est isomorphe à $(0,1)^V$, il suffit de définir $\overrightarrow{\kappa}$ et $\overrightarrow{\gamma}$ sur les boites de D_1 et que pour toute boite "frontière" B de D_1 (i.e. boite B telle que $B \subseteq \mathsf{D}_1$ et $\overline{B} \not\subseteq \mathsf{D}_1$), il existe $Z \in \mathsf{D}_1$ tel que $\frac{\overrightarrow{\kappa}(B)}{\overrightarrow{\gamma}(B)} \in Z$.

2.2 Systèmes Paramétriques d'Attracteurs

Définition 2 (Système Paramétrique d'Attracteurs) *Un système paramétrique d'attracteurs est un tuple* $\mathsf{A} = \langle K, V, \Theta, att \rangle$ *tel que :*

1. K *est un ensemble de* paramètres *et* V *est un ensemble fini de* composants *;*
2. $\widetilde{\Theta} : V \to \mathcal{P}_f(\mathbb{R}[K])$ *associe à chaque composant un ensemble fini de* seuils *(chaque seuil est un polynôme sur les paramètres) ; on note alors* $\widetilde{\Theta}(v) = \left\{\widetilde{\Theta}_1(v), ..., \widetilde{\Theta}_{n_v}(v)\right\}$ *; on définit* $\widetilde{\mathsf{D}}$ *l'*ensemble des prédicats de zones paramétriques *comme suit : un prédicat de* zone paramétrique *pour* $v \in V$ *est un prédicat de la forme* $\bigwedge_{i=1}^{n_v} \overrightarrow{x}(v) - \widetilde{\Theta}_i(v) \prec_i 0$ *où* $\prec_i \in \{<, =, >\}$ *; un prédicat de* zone paramétrique pour V *est un prédicat de la forme* $\bigwedge_{v \in V} \Pi(v)$ *où* $\Pi(v)$ *est un prédicat de zone paramétrique pour* v *; une* boite paramétrique *est donné par un prédicat de zone n'utilisant pas le signe "=" ; l'ensemble des boites et murs paramétrique est noté* $Dom(\widetilde{\mathsf{D}})$ *; l'ensemble des prédicats de boites paramétriques de* $\widetilde{\mathsf{D}}$ *est noté* $Box(\widetilde{\mathsf{D}})$ *;*
3. $att : Box(\widetilde{\mathsf{D}}) \to (V \to \mathbb{R}[K])$ *(i.e.* $att(B)(v)$ *est un polynôme de* $\mathbb{R}[K]$*).*

Pour une valuation $\overrightarrow{k} \in \mathbb{R}^K$ des paramètres, l'ensemble des prédicats Π de $Dom(\widetilde{\mathsf{D}})$ tels que $\exists \overrightarrow{x} \in \Pi\left(\overrightarrow{k}\right)$ définit un quadrillage noté $\widetilde{\mathsf{D}}(\overrightarrow{k})$ dont les boites sont données par des prédicats $\Pi(\overrightarrow{k})$ où Π est un prédicat de boite paramétrique. $\widetilde{\mathsf{D}}$ est ainsi un "quadrillage paramétrique".

La sémantique de A est donnée par le système de transitions $\mathsf{S}_\mathsf{A} = \langle \mathbb{R}^{K \cup V}, \underset{\mathsf{A}}{\rightarrow} \rangle$ tel que $\left(\overrightarrow{k}, \overrightarrow{x}\right) \underset{\mathsf{A}}{\rightarrow} \left(\overrightarrow{k}, \overrightarrow{y}\right)$ ssi il existe $Z \in Dom(\widetilde{\mathsf{D}}(\overrightarrow{k}))$ tel que :

1. $\overrightarrow{x} \in \overline{Z}$, $\overrightarrow{y} \in \overline{Z}$;
2. $\overrightarrow{y} \in Light(\overrightarrow{x}, \{att(B) \mid B \in adj(Z)\}) \cap \overline{Z}$.

De façon informelle, la transition $(\overrightarrow{k}, \overrightarrow{x}) \underset{\mathsf{A}}{\rightarrow} (\overrightarrow{k}, \overrightarrow{y})$ signifie que le système décrit le segment $[\overrightarrow{x}, \overrightarrow{y}]$ tel que $]\overrightarrow{x}, \overrightarrow{y}[\in Z$ et $\overrightarrow{x}, \overrightarrow{y} \in \overline{Z}$ pour un domaine Z de $\widetilde{D}(\overrightarrow{k})$ en tendant vers un attracteur $\overrightarrow{z}$ qui appartient à $Hull(att(B) \mid B \in adj(Z))$. La proposition suivante donne une condition suffisante pour que le système différentiel se comporte comme un système d'attracteurs.

Proposition 1 *Soit* $\mathsf{L} = \langle V, \Theta, \overrightarrow{\kappa}, \overrightarrow{\gamma} \rangle$ *un système différentiel linéaire par morceaux et* $\mathsf{A} = \langle \emptyset, V, \Theta, att \rangle$ *le système d'attracteurs tel que pour toute boite* B, $att(B)(v) = \frac{\overrightarrow{\kappa}(v)}{\overrightarrow{\gamma}(v)}$*. Si pour toute boite* B *la fonction* $\overrightarrow{\gamma}(B) : V \to \mathbb{R}$ *est constante alors les systèmes de transitions* S_L *et* S_A *sont bisimilaires.*

Le principal intéret des systèmes à attracteurs est que le calcul des prédécesseurs des régions peut se faire dans $\langle \mathbb{R}, +, \times, \leq \rangle$. On obtient alors un système symbolique [7].

3 Réseaux de Régulation Génétique

On note $s^+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, la *fonction partielle* s^+ *de Heaviside*.
Elle est définie sur $\mathbb{R}^2 \setminus \{(\xi, \xi) \mid \xi \in \mathbb{R}\}$ par :

$$s^+(x, y) = \begin{cases} 0 & \text{si } y < x \\ 1 & \text{si } x < y \end{cases}$$

On note $s^- : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ la fonction partielle $s^- = 1 - s^+$.

3.1 Hypergraphe Régulateur

Définition 3 (Hypergraphe régulateur) *Un* hypergraphe régulateur *est un tuple*

$$\mathsf{H} = \langle V, E, in, out, \alpha, \theta, \varepsilon, \overrightarrow{\beta_0} \rangle$$

tel que :

1. V *est un ensemble fini de* nœuds *et* E *est un ensemble fini d'*hyperarcs *;*
2. $in : E \to \mathcal{P}_f(V)$ *associe à chaque hyperarc un ensemble fini de nœuds* sources *;*
3. $out : E \to V$ *associe à chaque hyperarc un nœud* cible *;*
4. $\alpha : E \to \mathbb{R}_{>0}$ *associe à chaque hyperarc une constante d'*intensité *;*
5. $\theta : E \times V \to \mathbb{R}_{>0}$ *est la fonction partielle des* seuils *;*
6. $\varepsilon : E \times V \to \{-, +\}$ *est la fonction partielle des* signes *;*
7. θ, ε *sont définies sur* $\{(e, v) \mid v \in in(e)\}$ *;*
8. $\overrightarrow{\beta_0} : V \to \mathbb{R}_{\geq 0}$ *associe une* activité minimale *à chaque nœud.*

Les hyperarcs modélisent la régulation de l'activité considérée. Plus précisément, l'interprétation est la suivante : les protéines de $in(e)$ régulent l'activité de $out(e)$. La fonction d'intensité $\alpha(e)$ quantifie l'effet de la régulation de $out(e)$ par $in(e)$. Cette régulation ne peut cependant prendre que deux valeurs selon les concentrations des protéines de $in(e)$, 0 pour l'absence de régulation ou $\alpha(e)$. Cette intensité est définie sur un boite B par :

$$\alpha(e) \prod_{w \in in(e)} s^{\varepsilon(e)}(\theta(e, w), \overrightarrow{x_B}(out(e)))$$

où $\overrightarrow{x_B}$ est un point quelconque de B.

Définition 4 (Activité Régulée) *L'activié totale régulée par* H *est* $\overrightarrow{\beta} : Box(\mathsf{D}) \to \mathbb{R}^V$ *telle que pour* $v \in V$ *:*

$$\overrightarrow{\beta}(B)(v) = \overrightarrow{\beta_0}(v) + \sum_{\substack{e \in E \\ out(e) = v}} \alpha(e) \prod_{w \in in(e)} s^{\varepsilon(e)}(\theta(e, w), \overrightarrow{x}(v))$$

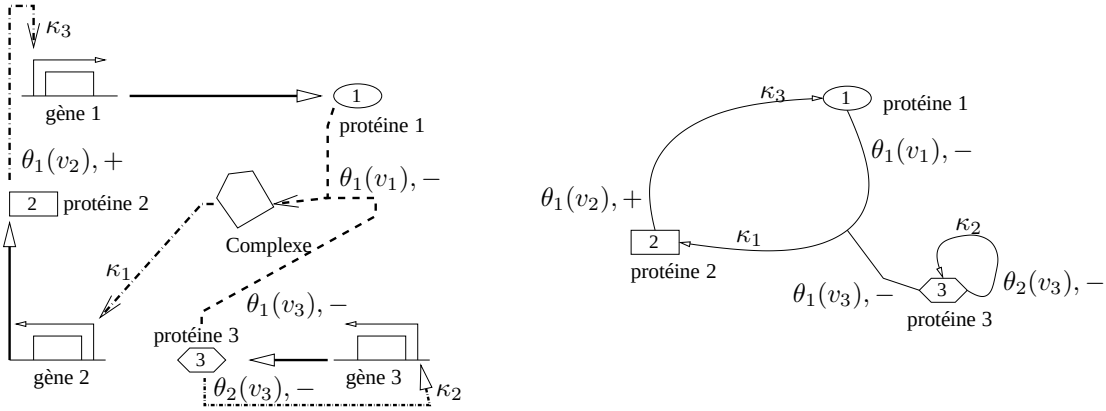


FIG. 1 – diagramme de régulation et son hypergraphe régulateur

Dans la Figure 1, on montre comment on transforme un diagramme de régulation en hypergraphe (seules les intensités minimales ne sont pas représentées).

3.2 Réseaux de Régulation

Définition 5 (Réseau de régulation gńétique) *Soit V un ensemble de protéines (ou de molécules). Un réseau de régulation génétique pour V est un couple $\mathsf{R} = \langle V, \mathsf{H}_\kappa, \mathsf{H}_\gamma \rangle$ tel que :*

1. H_κ *est un hypergraphe régulateur de nœuds V et d'activité κ appelé* hypergraphe regulateur de la production *; on note $\overrightarrow{\kappa}$ l'activité de régulation de la production ;*
2. H_γ *est un hypergraphe régulateur de nœuds V et d'activité γ appelé* hypergraphe regulateur de la dégradation *; on note $\overrightarrow{\gamma}$ l'activité de régulation de la dégradation.*

La sémantique différentielle de R est donnée par le système différentiel linéaire $\mathsf{L} = \langle V, \Theta, \overrightarrow{\kappa}, \overrightarrow{\gamma} \rangle$ où

$$\Theta(v) = \{\theta_{\mathsf{H}_\kappa}(e, v) \mid v \in in_{\mathsf{H}_\kappa}(e)\} \cup \left\{\theta_{\mathsf{H}_\gamma}(e, v) \mid v \in in_{\mathsf{H}_\kappa}(e)\right\}$$

Lorsque les constantes caractéristiques $\Theta(v), \kappa(e, v), \gamma(e, v), \overrightarrow{\beta[\kappa]_0}(v), \overrightarrow{\beta[\gamma]_0}(v)$ ne sont pas connues, on les remplace par des paramètres. Il est toujours possible de définir les prédicats de zones paramétriques $\overrightarrow{x}(v) - \theta \prec 0$ (pour $\theta \in \Theta(v)$) De même, les fonctions d'activités $\overrightarrow{\kappa}$ et $\overrightarrow{\gamma}$ s'étendent au cas paramétrique (elles sont définies à partir des s^ϵ de Heaviside donc en fonction de la satisfiabilité des prédicats de zones). Afin de pouvoir se ramener à l'étude de systèmes d'attracteurs, il suffit que $\overrightarrow{\gamma}$ soit constante dans chaque boite paramétrique.

4 Exemple

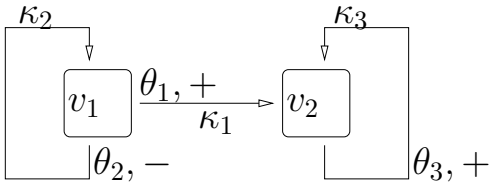


FIG. 2 – Exemple d'hypergraphe de régulaton de la production de deux protéines

On conclut par un exemple extrait de [1]. L'hypergraphe régulateur de la transcription est donné Figure 2. Les κ_i et les θ_i n'étant pas connus, ce sont donc des paramètres. On suppose également que la dégradation est uniforme. Les hypothèses de la proposition 1 sont donc vérifiées (dans un cadre plus général, il suffit que qu'elle soit indépendante de V dans chaque boite, la constante pouvant varier d'une boite à l'autre).

Pour analyser ce système, on utilise les techniques d'abstraction et de rafinement [1, 7]. Si $\mathsf{S} = \langle Q, \underset{\mathsf{S}}{\rightarrow} \rangle$ est un système de transitions et $\sim \subseteq Q \times Q$ est une relation d'équivalence, le système quotient $\mathsf{S}|_\sim = \langle Q, \underset{\mathsf{S}|_\sim}{\longrightarrow} \rangle$ est tel que $B \underset{\mathsf{S}|_\sim}{\longrightarrow} B'$ ssi il existe $q \in B$ et $q' \in B'$ tels que $q \underset{\mathsf{S}}{\rightarrow} q'$. Si $\mathcal{F}$ est un ensemble fini de parties de Q, on note $\underset{\mathcal{F}}{\sim} \subseteq Q \times Q$ la relation d'équivalence telle que $q \underset{\mathcal{F}}{\sim} q'$ ssi $\forall F \in \mathcal{F} : q \in F \Leftrightarrow q' \in F$. Étant donné un ensemble fini $\mathcal{F}_0$ représentant les régions observables de S, on construit des rafinements $\mathsf{S}|_{\underset{\mathcal{F}_i}{\sim}}$ de S, en posant $\mathcal{F}_i = \mathcal{F}_{i-1} \cup \{pre(F) \mid F \in \mathcal{F}_{i-1}\}$ où $pre(R) = \left\{q \in Q \mid \exists q' \in R : q \underset{\mathsf{S}}{\rightarrow} q'\right\}$. $\mathsf{S}|_{\underset{\mathcal{F}_i}{\sim}}$ est alors appelé *abstraction de rang i* de S. Pour l'exemple de la Figure 2, les régions observables donnent les positions relatives des concentrations par rapport aux seuils, i.e. $\mathcal{F}_0 = \{x < \theta_1,\ x = \theta_1,\ \theta_1 < x < \theta_2,\ x = \theta_2,\ \theta_2 < x,\ y < \theta_3,\ y = \theta_3,\ \theta_3 < y\}$.

La Figure 3 donne la position des attracteurs pour les conditions suivantes : $0 < \kappa_2 < \theta_1$, $\theta_3 < \kappa_3 < \kappa_1$ et $\theta_1 < \frac{\kappa_1(\theta_1-\kappa_2)}{\kappa_1-\theta_3} + \kappa 2 < \theta_2$. Le dernier terme de la conjonction vient de la division de la boite B (théroème de Thales appliqué au triangle de sommets $(\kappa_2, 0)$, $(l, 0)$ et (κ_2, κ_3)).

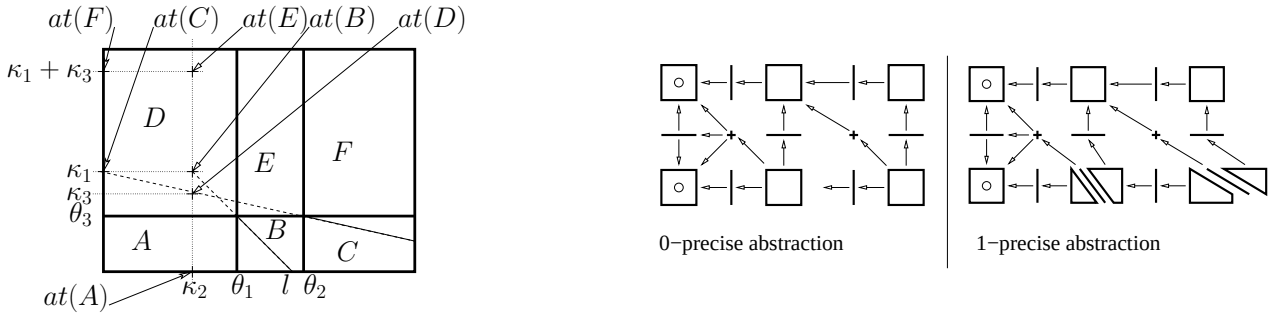


FIG. 3 – Position des attracteurs, abstractions de rang 0 et de rang 1

On peut remarquer que la boite B est divisée lors du calcul des prédécesseurs de $\overline{A} \cap \overline{B} \cap \overline{D} \cap \overline{E}$. L'abstraction de rang 1 restreinte aux valuations des paramètres satisfaisant la contrainte si dessus est donnée Figure 3. Dans cette abstraction, il y a deux points stationnaires représentés par des cercles : un dans la boite A, l'autre dans la boite D.

L'abstraction de rang 0 est celle construite par [4] : les régions sont les obsvervateurs (boites et murs) et aucune région n'est divisée. Dans cette abstraction, il est alors possible de se rendre à la boite A depuis la boite C. Dans l'abstraction de rang 1, toutes les trajectoires issues de C conduisent à la boite D. Ainsi, il n'est pas possible de se rendre dans la boite A depuis la boite C, ce que l'abstraction de rang 0 ne permettait pas de conclure.

Références

[1] M. Adélaïde and G. Sutre. Parametric analysis and abstraction of genetic regulatory networks. In *Proc. 2nd Workshop on Concurrent Models in Molecular Biology (BioCONCUR'04), London, UK, Aug. 2004*, Electronic Notes in Theor. Comp. Sci. Elsevier, 2004. To appear.

[2] H. de Jong. Modeling and simulation of genetic regulatory systems : A literature review. *Journal of Computational Biology*, 9(1) :67–103, 2002.

[3] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann. Qualitative simulation of genetic regulatory networks : Method and application. In *IJCAI*, pages 67–73, 2001.

[4] Hidde de Jong, Jean-Luc Gouzé, , Celie Hernandez, Michel Page, Tewfik Sari, and Johannes Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66 :301–340, 2004.

[5] J-L. Gouzé and T. Sari. A class of piecewise linear differential equations arising in biological models. *Dynamical systems*, 17 :299–316, 2003.

[6] S. Graf and H. Saïdi. Construction of abstract state graphs with PVS. In *lncs*, volume 1254 of *lncs*, pages 72–83, 1997.

[7] T. A. Henzinger and R. Majumdar. A classification of symbolic transition systems. In *lncs*, volume 1770 of *lncs*, pages 13–34. SV, 2000.

Évaluation symbolique appliquée à l'étude de réseaux de régulation génétique

Daniel Mateus[1] & Jean-Pierre Gallois[1] & Pascale Le Gall[2]

[1]CEA/DRT/LIST/DTSI/SOL/LLSP Saclay, F-91191 Gif sur Yvette Cédex
[2]Laboratoire des Méthodes Informatiques, CNRS UMR 8042, Université d'Évry

1 Introduction

La compréhension du fonctionnement des réseaux de régulation génétique nécessite des outils de modélisation et de simulation. En effet, la complexité des interactions entre les composants d'un réseau (principalement gènes et protéines) rend difficile la connaissance de ses évolutions possibles. La modélisation permet l'analyse des réseaux dans un cadre formel. Des propriétés peuvent alors être déduites de la structure même du modèle [9, 3]. La confrontation avec la réalité peut être réalisée par la simulation, qui peut révéler les comportements et propriétés spécifiques du système étudié. La simulation de modèles d'une certaine complexité est rendue possible par des outils informatiques.

Différents formalismes ont été utilisés pour décrire des réseaux de régulation génétique [2]. La précision permise par les différentes modélisations est limitée par le fait que de nombreux paramètres influençant les comportements ne peuvent être déterminés avec exactitude. La logique cinétique, introduite par R. Thomas [8, 9], associe à chaque concentration des substances du réseau une variable ne pouvant avoir qu'un nombre fini de valeurs entières. La description du réseau nécessite alors la connaissance d'un graphe d'interactions : il indique l'interaction positive ou négative de chaque substance sur le taux de synthèse des autres, et son seuil d'influence, i.e. la valeur entière de la variable à partir de laquelle l'influence est effective (la section 2 présente plus précisément les principes de la logique cinétique). L'évolution du système dépend encore des valeurs de paramètres logiques qui sont en général inconnus. Ainsi, une description logique d'un réseau de régulation composé de trois variables pouvant prendre trois valeurs peut être constitué de 24 paramètres logiques[1] ; il y a alors 3^{24} modèles différents possibles (plus de 282 milliards). Des propriétés simples permettent de réduire le nombre de modèles à partir de connaissances ou hypothèses sur le réseau [10], mais la difficulté de confronter le modèle aux hypothèses reste présente.

Dans cette optique, nous traduisons un modèle *incomplet*[2] de la logique cinétique en un *système de transitions étiquetées* équivalent, sur lequel vont pouvoir être appliquées des techniques d'analyse performantes, telles l'*évaluation symbolique* ou le *model-checking* (cf. section 3). En appliquant des techniques d'évaluation symbolique, on engendre un arbre contenant tous les comportements compatibles avec les contraintes connues. Les valeurs des paramètres conduisant au même comportement n'ont pas à être distinguées, ce qui conduit à une représentation compacte. De plus, chaque comportement est associé à une contrainte sur les paramètres : les modèles complets permettant ce comportement sont ceux vérifiant la contrainte. On peut ainsi simuler le système en posant des conditions initiales d'intérêt (aussi bien les valeurs initiales des variables, qu'une contrainte sur les paramètres) : on vérifie alors quels sont les comportements possibles et quelles sont les contraintes supplémentaires associées. D'autre part des techniques de model-checking peuvent être appliquées au graphe des comportements : étant donné une formule dans une logique temporelle linéaire, il est possible d'obtenir les chemins du graphe validant cette formule. Les contraintes correspondant au chemin nous indiquent alors l'ensemble des modèles permettant ce

[1]À chaque variable et chaque sous-ensemble de variables pouvant l'influencer, correspond un paramètre logique. Comme il y a 2^3 sous-ensembles d'un ensemble à trois éléments, il peut y avoir 3×2^3 paramètres.

[2]Le modèle est *incomplet* dans le sens où les paramètres logiques peuvent être inconnus.

comportement. La section 4 applique ces méthodes au réseau lié à la production de mucus chez *P. Aeruginosa*, étudié par ailleurs dans [1]. L'outil AGATHA [4], développé au CEA, permet l'application de ces techniques. AGATHA a été utilisé pour la validation de systèmes réactifs critiques [5]. La caractéristique d'AGATHA est de traiter le problème de l'explosion combinatoire du nombre d'états par des techniques d'évaluation symbolique, qui permettent en particulier de dénoter des ensembles d'états numériques par de simples états symboliques ; de plus ces techniques sont optimisées par des techniques spécifiques de détection des redondances [6].

2 Logique cinétique (dite généralisée)

La présentation qui suit est une adaptation de la modélisation logique de réseaux de régulation génétique présentée dans [1, 7, 8, 9].

Considérons un réseau de régulation génétique constitué de n gènes. À chaque gène du réseau est associé une variable x_i pouvant prendre des valeurs entières comprises entre 0 et b_i. Une *description logique* d'un réseau de n variables est la donnée d'une *matrice d'interactions* et des *paramètres logiques* suivants :

- La *matrice d'interactions* I de n lignes et n colonnes est telle que les coefficients de la colonne j appartiennent à $\{+,-\} \times \{1,\ldots,b_j\} \cup \{(0,0)\}$. Le coefficient $I_{ij} = (\varepsilon, s)$ est appelé interaction de la variable x_j sur la variable x_i, de signe ε et de seuil s. Si $I_{ij} = (0,0)$, on dit que l'interaction est nulle.
- Soit $\mathcal{P}_i$ l'ensemble des variables ayant une interaction non nulle sur x_i. Pour tout $\omega \subseteq \mathcal{P}_i$ on définit un *paramètre logique* $K_\omega^{(x_i)} \in \{0,\ldots,b_i\}$ (ces paramètres expriment l'état vers lequel évolue la variable x_i, qui dépend des influences des variables de ω).

La matrice d'interactions peut aussi bien être remplacée par un *graphe d'interactions* dont les nœuds sont les variables $x_1,\ldots,x_n$ et dont les transitions $x_j \xrightarrow{\varepsilon,s} x_i$ correspondent à l'interaction non nulle de x_j sur x_i de signe ε et de seuil s.

Les définitions suivantes vont permettre de construire le graphe de séquence des états qui traduit la dynamique du réseau.

État d'un réseau Un *état du réseau* est un n-uplet d'entiers $E = (E_1,\ldots,E_n)$ tel que pour tout $i \leq n$, $E_i \leq b_i$.

Valeurs logiques Étant donné un état du réseau $E = (E_1,\ldots,E_n)$, on appelle *valeur logique* de la variable x_i dans l'état E l'entier E_i. On note $x_i(E) = E_i$.

Ressources d'une variable [1] Soit $E = (E_1,\ldots,E_n)$ un état du système. Pour $i \leq n$, la variable x_j appartenant à $\mathcal{P}_i$ est dite *ressource* de x_i dans l'état E si :

- L'interaction de x_j sur x_i (correspondant au coefficient I_{ij} de la matrice d'interactions) est positive et E_j est supérieur ou égal au seuil ;
- L'interaction I_{ij} de x_j sur x_i est négative et E_j est strictement inférieur au seuil.

État successeur Soit E un état du réseau et $R_i(E)$ l'ensemble des ressources de x_i dans l'état E

- Si $x_i(E) < K^{(x_i)}_{R_i(E)}$, on dit que $E' = (E_1,\cdots,E_i+1,\cdots,E_n)$ est un successeur de E.
- Si $x_i(E) > K^{(x_i)}_{R_i(E)}$, on dit que $E' = (E_1,\cdots,E_i-1,\cdots,E_n)$ est un successeur de E.
- Si pour tout $i \leq n$, $x_i(E) = K^{(x_i)}_{R_i(E)}$, alors E est un état stationnaire du réseau.

Graphe de séquence des états Le *graphe de séquence des états* a pour nœuds les états possibles du réseau et chaque arc orienté relie un état à un de ses successeurs.

Exemple 1 *La bactérie* Pseudomonas aeruginosa *produit du mucus dans les poumons de patients atteints de mucoviscidose. Un graphe d'interactions possible correspondant au réseau de régulation contrôlant la production de mucus est donné par la figure 1. La variable x correspond*

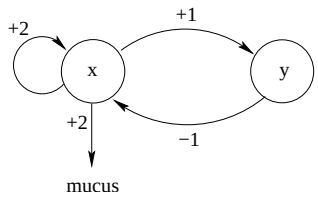


FIG. 1 – Graphe d'interactions

à la protéine AlgU, qui a une influence positive sur sa propre production, sur les gènes intervenant dans la production de mucus ainsi que sur la production d'un complexe inhibiteur anti-AlgU, qui correspond à la variable y. Dans ce modèle $x = 2$ entraîne la production de mucus. Les paramètres logiques sont inconnus [1].

3 Système de transitions étiquetées équivalent au modèle logique

Le système de transitions étiquetées que nous allons utiliser pour traduire la description logique partielle est un cas particulier d'Extended Labelled Transition System (ELTS) décrit dans [6]. Il est constitué de *points de contrôle* liés par des transitions et est associé à un ensemble fini de variables. Chaque transition est constituée d'une *garde*, qui est la condition sur les variables permettant de franchir la transition, et d'*affectations*, qui déterminent la valeur des variables dans la suite de l'exécution.

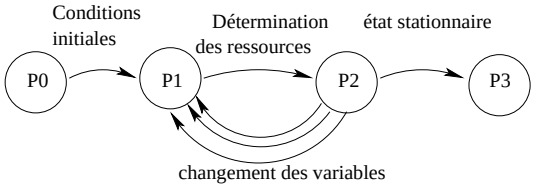


FIG. 2 – ELTS associé au modèle logique

On suppose connue la matrice d'interactions I de la description logique. On note $I_{ij}^{(1)}$ le signe et $I_{ij}^{(2)}$ le seuil de l'interaction I_{ij}. On construit alors l'ELTS associé de la figure 2 ; ses 4 points de contrôle sont $P_0, \ldots, P_3$ (P_0 étant le point de contrôle initial). En P_0, le système n'est pas initialisé. La transition vers P_1 va contraindre les paramètres selon les connaissances ou les hypothèses (contrainte notée $\mathcal{C}$), et éventuellement initialiser les variables dans un état du réseau précis. En P_1, les paramètres vérifient donc les contraintes et le système est dans un état donné. La transition vers P_2 va déterminer quelles sont les ressources dans cet état. Ainsi en P_2 les ressources sont déterminées. Alors, soit les variables doivent être modifiées, ce qui donne un nouvel état et la transition se fait vers P_1, soit l'état est stationnaire, et la transition se fait vers P_3. Ce qui donne en détail la description ci-dessous.

$x_1, \ldots, x_n$ désignent les variables de la description logique et $K_\omega^{(x_i)}$ correspond à un paramètre logique (ω étant une partie de $\{x_1, \ldots, x_n\}$) ; leurs valeurs possibles sont des entiers positifs dont le maximum est fixé en fonction de la matrice I. $R_1, \ldots, R_n$ sont des parties de $\{x_1, \ldots, x_n\}$. La condition de la transition de P_0 vers P_1, notée $\mathcal{C}$, traduit les connaissances initiales sur les paramètres logiques. Dans le cas de l'exemple 1, on peut poser :

$$\mathcal{C} = (K_\emptyset^{(x)} = 0) \wedge (K_\emptyset^{(y)} = 0) \wedge (K_{\{x\}}^{(x)} \leq K_{\{x,y\}}^{(x)}) \wedge (K_{\{y\}}^{(x)} \leq K_{\{x,y\}}^{(x)}) \quad (1)$$

En effet, si on considère que les influences des ressources s'ajoutent, alors un paramètre logique associé à un ensemble de ressources A est inférieur au paramètre logique associé à B lorsque A est inclus dans B [7].

Les transitions de l'ELTS sont alors les suivantes :

Conditions initiales La transition de P_0 vers P_1 a pour condition $\mathcal{C}$.

Affectation des ressources La transition de P_1 vers P_2 affecte les ressources de x_i à R_i pour i variant de 1 à n ; il y a donc n affectations de la forme :

$$R_i := \bigcup_{j\in\{1,\ldots,n\}} \left[((I_{ij}^{(1)} = +) \wedge (I_{ij}^{(2)} \leq x_j)) \vee ((I_{ij}^{(1)} = -) \wedge (I_{ij}^{(2)} > x_j))\right] \{x_j\}$$

Dans cette formule $False\{x\} = \emptyset$ et $True\{x\} = \{x\}$. Ainsi R_i contient x_j si l'interaction de x_j sur x_i est positive et x_j est supérieur au seuil, ou si l'interaction est négative et x_j est inférieur au seuil ; donc x_j appartient à R_i lorsque x_j est ressource de x_i.

Changement des variables Il y a $2n$ transitions de P_2 vers P_1 ; chacune affecte une des variables x_i d'une nouvelle valeur selon le paramètre logique associé. Soit, pour $i \in \{1, \ldots, n\}$:

- n transitions ont pour condition $K_{R_i}^{(x_i)} < x_i$ et pour affectation $x_i := x_i - 1$;
- n transitions ont pour condition $K_{R_i}^{(x_i)} > x_i$ et pour affectation $x_i := x_i + 1$.

État stationnaire La transition de P_2 vers P_3 vérifie si l'état est stationnaire ; la condition est

$$\bigwedge_{i\in\{1,\ldots,n\}} (K_{R_i}^{(x_i)} = x_i).$$

La *sémantique* de l'ELTS est intuitivement l'ensemble de ses exécutions possibles si les variables sont initialisées avec une valeur numérique. La sémantique est un ensemble de *chemins numériques* constitués d'*états numériques*. Un état numérique est caractérisé par un point de contrôle de l'ELTS, et par la valeur numérique de chaque variable. Un chemin numérique est une suite d'états numériques tel que deux états successifs ont des points de contrôle liés par une transition de l'ELTS ; de plus les valeurs numériques des variables vérifient la garde, et correspondent aux affectations de la transition.

Considérons un chemin numérique de l'ELTS représenté par la figure 2, associé à une matrice d'interactions ; la succession des états numériques de point de contrôle P_1 correspond à un chemin du graphe de séquence des états ; la description logique correspondante est celle dont les paramètres logiques sont précisés dans l'état numérique. La sémantique de l'ELTS contient donc tous les modèles associés à la matrice d'interactions (les paramètres pouvant avoir toutes les valeurs numériques possibles), ainsi que les chemins des graphes de séquence des états associés à chaque modèle. Dans la section suivante nous allons appliquer à ce système des techniques d'évaluation symbolique qui vont permettre de connaître tous les comportements sans avoir à énumérer les valeurs des paramètres.

4 Évaluation symbolique du système et exploitation des résultats

L'idée de l'évaluation symbolique est d'associer chaque variable de l'ELTS à un symbole initial, appelé *constante symbolique* et d'exprimer les valeurs prises par les variables durant l'exécution du système en fonction de ces constantes. Les gardes des transitions introduisent des conditions sur les constantes permettant de les franchir : les différents chemins possibles sont donc associés à une condition de chemin (Path Condition ou PC), et qui est nécessaire pour que ce chemin soit emprunté par le système. L'outil AGATHA utilise cette technique pour construire un arbre des comportements. Les nœuds de l'arbre (ou *états symboliques*) sont constitués d'un point de contrôle

correspondant de l'ELTS, d'une condition de chemin, et de l'expression de chaque variable en fonction de sa constante symbolique initiale. À chaque transition issue du point de contrôle d'un nœud correspond un de ses fils. Celui-ci a une condition de chemin obtenue en faisant la conjonction de la condition de chemin précédente et de la garde de la transition, et les variables sont exprimées selon les affectations de la transition. Ainsi la condition de chemin d'un état est la condition qui permet d'atteindre cet état depuis l'origine. Donc les états symboliques dont la condition de chemin est fausse quelles que soient les constantes initiales ne sont pas intégrés dans l'arbre : ces nœuds ne peuvent pas être atteints.

La figure 3 représente l'arbre des comportements de l'ELTS associé au graphe d'interactions de l'exemple 1. Seuls les états symboliques de point de contrôle P_1 ou P_3 ont été conservés (les états de point de contrôle P_3 sont notés $Stat$ sur la figure, pour état stationnaire) et les valeurs de x et y sont indiquées. Nous avons posé comme conditions initiales $x = 0$ et $y = 0$; de plus nous avons supposé que les paramètres logiques vérifient la condition $\mathcal{C}$ de la formule 1. Une flèche

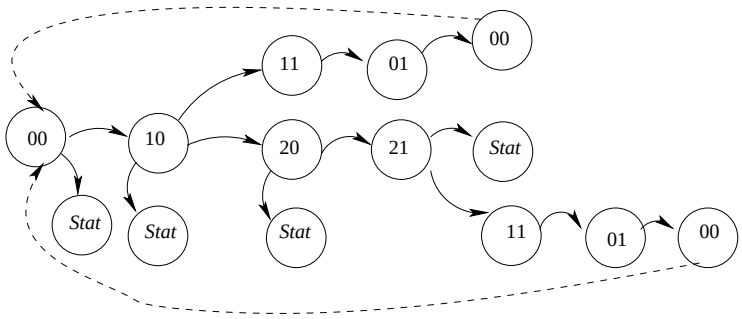


FIG. 3 – Arbre des comportements

en pointillé indique que l'ensemble des valeurs possibles pour les paramètres dans le dernier état symbolique est inclus dans l'ensemble des valeurs possibles d'un état déjà construit (c'est une des conditions d'arrêt possible pour AGATHA). Le chemin qui conduit de (00) à (20) est impossible d'après l'expérimentation (les bactéries ayant une concentration associée à x très faible ne passe pas dans un état de production de mucus). Le PC de cet état est $\mathcal{C} \wedge (K^{(x)}_{\{y\}} = 2)$. Ce qui signifie que dans un modèle logique compatible avec cette donnée expérimentale $K^{(x)}_{\{y\}} < 2$.

Pour automatiser la recherche de chemins précis, il est possible d'adapter des techniques de *model-checking* de formules LTL (linear temporal logic). LTL est une logique permettant d'exprimer des propriétés de chemin. Aux opérateurs de la logique propositionnelle (négation, conjonction, disjonction) s'ajoutent des opérateurs temporels, que l'on peut réduire à Next (X) et Until (U). Si φ et ψ sont des formules, $X\varphi$ signifie que φ est vraie dans l'état suivant du chemin, et $\varphi U \psi$ signifie que φ est vraie jusqu'à ce que ψ soit vraie (ψ finissant par être vraie). On peut en particulier définir les opérateurs Finally (F) et Globally (G) : $F\varphi$ signifie que φ finit par être vraie, soit $(True)U\varphi$, et $G\varphi$ signifie que φ est toujours vraie, soit $\neg(F\neg\varphi)$.

Les propriétés que nous souhaitons vérifier concernant la description logique de l'exemple 1 sont les suivantes [1] : tous les chemins qui commencent en $x = 0$ ne passent pas par un état où $x = 2$, et au moins un chemin commençant en $x = 2$ revient toujours en $x = 2$. Nous cherchons donc les chemins vérifiant $(x = 0) \wedge F(x = 2)$, puis $(x = 2) \wedge G(F(x = 2))$.

Le seul chemin direct vérifiant la première propriété est $00 \rightarrow 10 \rightarrow 20$ (i.e. les autres chemins vérifiant la propriété passent par une transition d'inclusion vers (00) ou (01), ce qui entraîne que les PC des états symboliques sont *plus restrictifs* que ceux du chemin direct). Et donc les modèles logiques ayant un comportement vérifiant la propriété sont ceux vérifiant $K^{(x)}_{\{y\}} = 2$.

La deuxième propriété s'applique spécifiquement à des chemins infinis. Les seuls chemins qui conviennent sont ceux où (21) ou (20) est stationnaire, et $(21 \rightarrow 11 \rightarrow 01 \rightarrow 00 \rightarrow 10 \rightarrow$

$20 \rightarrow 21$) ce qui donne les PC : $(K^{(x)}_{\{x\}} = 2 \wedge K^{(y)}_{\{x\}} = 1)$ ou $(K^{(x)}_{\{x,y\}} = 2 \wedge K^{(y)}_{\{x\}} = 0)$ ou $(K^{(x)}_{\{x\}} < 2 \wedge K^{(x)}_{\{y\}} > 1 \wedge K^{(y)}_{\{x\}} > 0)$.

Ainsi les modèles qui ne vérifient pas la première propriété mais vérifient la seconde sont tels que $(K^{(x)}_{\{y\}} < 2)$ et $((K^{(x)}_{\{x\}} = 2 \wedge K^{(y)}_{\{x\}} = 1) \vee (K^{(x)}_{\{x,y\}} = 2 \wedge K^{(y)}_{\{x\}} = 0))$ (soit 8 modèles).

5 Conclusion

Nous avons vu comment un modèle de logique cinétique de René Thomas connu partiellement peut être traduit en un système de transitions étiquetées (ELTS). Nous pouvons appliquer des techniques d'évaluation symbolique à cet ELTS pour obtenir tous les comportements possibles et les contraintes sur les modèles correspondant à chaque comportement. De plus des techniques de recherche automatique des chemins permettent d'obtenir les modèles ayant une propriété donnée.

Dans [1] pour sélectionner les modèles vérifiant une propriété, tous les modèles sont générés, puis tous les graphes de transitions, sur lesquels peuvent être appliqués les techniques de model-checking. La technique exposée ici ne nécessite pas d'énumérer tous les paramètres, et peut ainsi être plus efficace pour la vérification de certaines propriétés.

Notre approche permet de simuler un modèle sans avoir à préciser la valeur numérique des paramètres logiques. Étant donnée une hypothèse concernant les paramètres, il suffit de spécifier la contrainte correspondante dans les conditions initiales du système de transitions pour obtenir les comportements compatibles. De même, les conditions de chemin associées à une évolution du système constituent de nouvelles contraintes que les paramètres doivent vérifier pour que cette évolution soit possible. Cela fournit des possibilités d'expérimentations facilement exploitables pour raffiner le modèle.

Références

[1] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks : extending thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3) :339–347, 2004.

[2] H. de Jong. Modeling and simulation of genetic regulatory systems : A literature review. *Journal of Computational Biology*, 9(1) :67–103, 2002.

[3] J. Demongeot, J. Aracena, F. Thuderoz, T. Baum, and O. Cohen. Genetic regulation networks : circuits, regulons and attractors. *C. R. Biologies*, 326(2) :171–88, 2003.

[4] J.-P. Gallois, C. Gaston, and A. Lapitre. Agatha, un outil de simulation symbolique. In *AFDL, Besançon, France*, 2004.

[5] D. Lugato, C. Bigot, Y. Valot, J.-P. Gallois, S. Gérard, and F. Terrier. Validation and automatic test generation on uml models : the agatha approach. *STTT*, 5(2-3) :124–139, 2004.

[6] N. Rapin, C. Gaston, A. Lapitre, and J.-P. Gallois. Behavioral unfolding of formal specifications based on communicating extended automata. In *ATVA*, Taipei, Taiwan, 2003.

[7] E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations : A discrete mapping approach. *Dyn. Stability Syst.*, 4 :189–207, 1989.

[8] R. Thomas and R. D'Ari. *Biological Feedback*. CRC Press, Boca Raton, Florida, 1990.

[9] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory.i & ii. *Chaos*, 11(1) :170–95, 2001.

[10] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks –i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57(2) :247–76, 1995.

Réseaux de jeux et modules élémentaires

Matthieu Manceny[1] & Franck Delaplace[1]

[1]Laboratoire de Méthodes Informatiques (LaMI), UMR 8042 CNRS-Université d'Evry, FRANCE

Résumé

Dans cet article nous proposons une extension modulaire originale de la théorie des jeux : la *théorie des réseaux de jeux*. L'objectif de cette extension est de fournir un cadre de travail théorique permettant de modéliser les *dynamiques modulaires* résultant d'*interactions locales* entre différents agents. Les réseaux de jeux décrivent des situations où un agent peut participer à plusieurs jeux de manière simultanée. Nous nous focalisons en particulier sur la détermination d'*équilibres globaux*, résultant de la composition des équilibres locaux à chaque jeu du réseau.

Cependant, plusieurs réseaux de jeux peuvent représenter la même situation. Nous recherchons alors une *forme normale* qui mette en valeur des jeux aussi petits que possible en terme de nombre de joueurs. Ces jeux sont qualifiés de *modules élémentaires*. Un algorithme permettant de décomposer un jeu en modules élémentaires est donné.

1 Introduction

La formalisation des systèmes moléculaires par des réseaux d'interactions propose une représentation du système sous forme d'un graphe où les nœuds du réseau représentent les agents moléculaires (protéines, gènes, complexes, métabolites, ...) et les arcs les relations d'interactions. La nature des interactions dépend du contexte de modélisation et, pour ne donner qu'un exemple, citons les réseaux d'expression des gènes qui décrivent la capacité d'un gène à réguler indirectement, par la protéine produite, l'expression d'un autre gène. Les réseaux d'interactions fournissent essentiellement la structure des interactions. Les propriétés découlant de leur analyse correspondent généralement à des *propriétés potentielles* du système. Par exemple, un réseau dont le degré de connectivité suit une loi puissance (power law) identifie des propriétés de robustesse du système aux « attaques » non ciblées ([5, 1]), grâce à la présence en faible nombres de concentrateurs (hubs), nœuds à forts degrés. De manière plus générale, l'étude des réseaux d'interactions dans un cadre de modélisation (biologique) repose sur un parallèle entre les propriétés du réseau (loi puissance) et celles ayant trait à la dynamique ou à l'évolution du système considéré (robustesse).

Pour pousser plus en avant cette étude vers l'analyse des dynamiques d'interactions moléculaires, il est nécessaire d'adjoindre un modèle examinant comment interagissent ces agents et quel est leur impact sur leur état interne. L'étude de la fonction biologique est au centre de cette analyse car celle-ci résulte en partie des interactions moléculaires et de leur évolution.

L'une des propriétés jugées centrale pour l'étude des systèmes biologiques concerne leur *modularité* ([9]). Schématiquement, un module peut se définir comme un ensemble d'agents, un *support*, agissant de manière coordonnée dans la perspective de réaliser une *fonction* spécifique. Pour l'analyse des réseaux moléculaires nous pouvons assimiler la fonction à une réponse coordonnée à un stimulus environnemental, comme c'est le cas pour l'analyse des réseaux génétiques par les méthodes de puces à ADN. L'intérêt de cette analyse est notamment de fournir des cibles thérapeutiques complexes ne restreignant pas la cible à un unique agent en rapport avec la fonction considérée mais en l'élargissant à plusieurs agents coopérant (le support du module réalisant la fonction considérée).

A la différence d'artefacts où un module s'identifie souvent par des propriété de localité spatiale, il semblerait que l'étude des réseaux moléculaires ne révèle pas cette propriété ([10]). Pour étudier la modularité des réseaux moléculaires, il est nécessaire de considérer des modèles représentant

la dynamique des interactions. Dans le cadre d'une modélisation discrète, nous modéliserons la dynamique des interactiosn par la théorie des jeux et plus précisément la *théorie des réseaux de jeux*. La théorie des jeux a pour objet de modéliser les interactions stratégiques entre des agents par un jeu ([8]). Elle étudie comment des agents (ou joueurs) en interaction font évoluer leur choix (ou leur état) en fonction de la nature des interactions (le jeu). Les applications de la théorie des jeux dépassent le cadre de celui des jeux et couvrent différents champs tels que l'Economie ([4]) et la Biologie ([2, 6]). La théorie des réseaux de jeux étend la théorie des jeux en considérant un ensemble de jeux inter-connectés en réseaux. Ainsi, un joueur peut participer à plusieurs jeux. Schématiquement, chaque jeu représentera la dynamique d'un module qui se définit par les joueurs connectés à celui-ci.

L'article se présente de la manière suivante : la partie 2 présente les notions principales de la théorie des jeux et de la théorie des réseaux de jeux. La partie 3 s'intéresse à la recherche de modules élémentaires dans les réseaux de jeux et donne un algorithme effectuant cette recherche.

2 Théorie des jeux — théorie des réseaux de jeux

La théorie des réseaux de jeux est une extension de la théorie des jeux. La section 2.1 présente les principales notions de théorie des jeux et la section 2.2 s'intéresse à l'extension. Le lecteur peut se référer à [7] ou [3] pour une présentation plus complète.

2.1 Théorie des jeux

Jeux stratégiques. La théorie des jeux stratégiques propose un modèle d'interactions où les agents interagissant choisissent ce qu'ils vont faire, leur *stratégie*, une fois pour toute et de manière simultanée. De plus, chaque agent est *rationnel* — il cherche à maximiser son gain — et *parfaitement informé* des gains des autres agents. Formellement, un jeu stratégique se définit de la manière suivante :

Définition 1 (Jeu stratégique)
Un jeu stratégique Γ *est un triplet* $\langle A, C, u\rangle$ *où :*

- A *est l'ensemble des agents, ou* joueurs.
- $C = \{C_i\}_{i\in A}$ *est un ensemble d'ensembles de* stratégies *;* $C_i = \{c_i^1, ..., c_i^{m_i}\}$ *est l'ensemble des stratégies du joueur* i.
- $u = (u_i)_{i\in A}$ *est le vecteur des* fonctions de gains *;* $u_i : \times_{i\in A} C_i \mapsto \mathbb{R}$ *est une fonction qui attribue un gain au joueur* i *suivant la configuration du jeu i.e. les stratégies des autres joueurs.*

Représentation par tableau. Les jeux stratégiques 2×2 — 2 joueurs ayant chacun 2 stratégies — sont souvent utilisés en théorie des jeux pour en présenter les notions. Un tel jeu est habituellement représenté par un tableau où les stratégies du premier joueur sont en ligne et celles du second joueur en colonnes. Ainsi dans l'exemple de la figure 1, si le joueur x joue sa stratégie `Off` et le joueur y sa stratégie `On`, un gain de 0 est attribué au joueur x et un gain de 2 au joueur y.

Équilibre de Nash. Un concept central en théorie des jeux est la notion d'*équilibres de Nash* qui permet de capturer les configurations stables d'un jeu stratégique :

Définition 2 (Équilibre de Nash)
Soit $\Gamma = \langle A, C = \{C_i\}_{i\in A}, u = (u_i)_{i\in A}\rangle$ *un jeu stratégique. Un équilibre de Nash est une configuration du jeu* $c^* \in \times_{i\in A} C_i$ *telle que :*

$$\forall i \in A, \forall c_i \in C_i, u_i((c^*_{-i}, c_i)) \leq u_i(c^*)$$

Tableau de gains

x/y	Off	On
Off	$(1,1)$	$(0,2)$
On	$(2,0)$	$(-1,-1)$

Equilibres de Nash

$\{(\texttt{On},\texttt{Off}),(\texttt{Off},\texttt{On})\}$

FIG. 1 – Exemple de jeu à deux joueurs x, y ayant chacun deux stratégies

où (c^*_{-i}, c_i) correspond à la configuration c^* dans lequel le joueur i joue sa stratégie c_i (plutôt que c^*_i).

Dans un équilibre de Nash la stratégie jouée par l'agent i est la *meilleure réponse* possible aux stratégies des autres joueurs. L'agent i n'a donc pas d'intérêt à changer, seul, de stratégie.

Dans l'exemple de la figure 1 on trouve deux équilibres de Nash qui correspondent aux configurations $(x = \texttt{Off}, y = \texttt{On})$ et $(x = \texttt{On}, y = \texttt{Off})$.

2.2 Théorie des réseaux de jeux

Réseaux de jeux stratégiques. En théorie des jeux, tous les agents intéragissent les uns avec les autres. La théorie des réseaux de jeux étend la théorie des jeux, et autorise une description modulaire de la dynamique du réseau. Les agents peuvent ainsi participer à plusieurs jeux de manière simultanée. Les jeux du réseau peuvent alors être vus comme des modules dynamiques décrivant les interactions locales entre les agents participant au jeu. Formellement, un réseau de jeux se définit de la manière suivante :

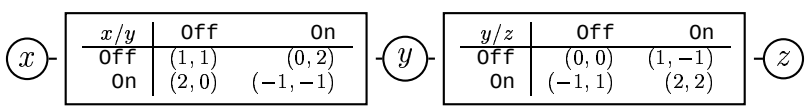
Définition 3 (Réseau de jeux)
Un réseau de jeux est un triplet $\langle \mathcal{A}, C, \mathcal{U} \rangle$ où :

- *$\mathcal{A}$ est l'ensemble des agents, ou joueurs.*
- *$C = \{C_i\}_{i \in A}$ est un ensemble d'ensembles de stratégies ; $C_i = \{c_i^1, ..., c_i^{m_i}\}$ est l'ensemble des stratégies du joueur i.*
- *$\mathcal{U} = \{\langle A_j, u^j \rangle\}$ est un ensemble de jeux avec pour chaque noeud $A_j \subseteq \mathcal{A}$ l'ensemble des agents et $u^j = (u_i^j : \times_{i \in A_j} C_i \mapsto \mathbb{R})_{i \in A_j}$ le vecteur des fonctions de gains des agents impliqués dans le jeu.*

Il n'est pas nécessaire dans les jeux de rappeler l'ensemble des stratégies d'un agent, car celles-ci sont identiques pour tous les jeux auxquels il participe, et sont donc associée à l'agent plutôt qu'au jeu.

Représentation graphique. Les réseaux de jeux se représentent sous forme de graphes bipartis (fig. 2). Dans un tel graphe, les agents sont représentés par un cercle qui contient leur nom et les jeux par des rectangles. Les agents sont reliés aux jeux auxquels ils participent.

Réseau de jeux



x/y	Off	On
Off	$(1,1)$	$(0,2)$
On	$(2,0)$	$(-1,-1)$

y/z	Off	On
Off	$(0,0)$	$(1,-1)$
On	$(-1,1)$	$(2,2)$

Équilibres globaux

$\{(\texttt{On},\texttt{Off},\texttt{Off}),(\texttt{Off},\texttt{On},\texttt{On})\}$

FIG. 2 – Exemple de réseau de jeux à trois joueurs et deux jeux

Équilibres. Deux types de dynamique émergent de la représentation en réseau de jeux : une locale à chaque jeu et une globale sur l'ensemble du réseau. De fait, deux notions d'équilibres sont définies : les équilibres locaux et les équilibres globaux. Les *équilibres locaux* correspondent aux équilibres de Nash de chacun des jeux constituant le réseau. Dans l'exemple de la figure 2, on trouve ainsi deux équilibres locaux pour le jeu x/y ($(x = \texttt{Off}, y = \texttt{On})$ et $(x = \texttt{On}, y = \texttt{Off})$) et deux équilibres locaux pour le jeu y/z ($(y = \texttt{Off}, z = \texttt{Off})$ et $(y = \texttt{On}, z = \texttt{On})$). Les *équilibres globaux* correspondent à une configuration d'équilibres pour l'ensemble des jeux du réseau et sont calculés en combinant les équilibres de Nash des différents jeux. Dans notre exemple, y peut avoir la stratégie `Off` ou `On`, ce qui correspond à deux équilibres globaux : $(x = \texttt{On}, y = \texttt{Off}, z = \texttt{Off})$ et $(x = \texttt{Off}, y = \texttt{On}, z = \texttt{On})$.

3 Recherche de modules élémentaires

3.1 Structure et équivalence de réseaux

Dans le cadre des réseaux de jeux, chacun des jeux constituant le réseau s'identifie naturellement comme étant un module. Un réseau de jeux peut donc être vue comme la composition de modules reliés les uns aux autres au travers des agents. Chaque module définit une *dynamique locale* et la *structure* du réseau — la manière dont sont reliés les modules — une dynamiqque globale. Ces dynamiques peuvent être observées aux travers de leurs états stables respectifs : les équilibres locaux et globaux.

Cependant, différentes structures peuvent modéliser la même dynamique. Dans l'exemple de la figure 4, le réseau à un jeu unique, à trois joueurs à gauche, possède les mêmes équilibres, et la même dynamique, que le réseau à deux jeux, à droite. Les deux réseaux sont alors dits *équivalents*.

De fait, la recherche d'une « *forme normale* », c'est-à-dire la représentation canonique d'un réseau de jeux, est indispensable. La forme normale d'un réseau de jeux se définit comme étant un réseau de jeux équivalent — ayant les mêmes équilibres globaux — et dont les jeux mettent en interactions le moins de joueurs possible. Dans la forme normale d'un réseau de jeux, les jeux sont qualifiés de « *modules élémentaires* ».

3.2 Algorithme

L'algorithme de la figure 3 permet de séparer un jeu en modules élémentaires[1]. Il se fonde sur la notion de dépendance entre agents.

Dépendance Intuitivement, un agent A dépend d'un agent B si les gains de A sont modifiés par les stratégies de B. Plus formellement, la notion de dépendance se définit de la manière suivante :

Définition 4 (Dépendance)
Soit $\langle A, C, u\rangle$ un jeu stratégique et $j, i \in A^2, i \neq j$ deux agents. j dépend *de i, on note $i\delta_u j$, si :*

$$\exists c_i \in C_i, \exists c'_i \in C_i, \exists c_{-i} \in C_{-i}, u_j(c_{-i}, c_i) \neq u_j(c_{-i}, c'_i)$$

Plus précisément, l'algorithme s'attache à la notion de prédécesseur :

Définition 5 (Prédécesseurs)
Soit $\langle A, C, u\rangle$ un jeu stratégique. On note par $\delta_u^-(j)$, $j \in A$, l'ensemble des prédécesseurs de j :

$$\forall j \in A, \delta_u^-(j) = \{i \in A | i\delta_u j \wedge i \neq j\}$$

[1] Pour des raisons de place on ne présente ici que l'algorithme, sa correction est montrée dans [3]

La notion de dépendance, et de prédécesseur, est utilisée pour surligner les interactions entre les agents et déterminer ceux qui participent à un même module élémentaire. En particulier, pour chaque agent il existe un module élémentaire qui le contient lui et ses prédécesseurs. De plus, il ne peut pas y avoir de relation d'inclusion entre les agents participant à deux modules élémentaires différents.

Recherche de gains Une fois trouvés les agents participant aux modules élémentaires, il reste à attribuer les gains. Pour un agent $a \in A$ participant à un module élémentaire G :

- si tous les prédécesseurs de a sont dans G, on peut facilement calculer ses gains car aucun des agents absents n'a d'influence sur a. La fonction *pick* de l'algorithme de la figure 3 choisit une configuration du jeu de départ possédant les mêmes stratégies que les joueurs du module élémentaire et en extrait les gains de a.
- si au moins un des prédécesseurs de a n'est pas dans G, tous les gains de a sont nuls.

La figure 3 décrit l'algorithme de séparation, qui est illustré sur la figure 4.

---

function **separate**($\langle A, C, u\rangle$: un jeu)
  $\mathcal{U}' := \emptyset$; $g := 0$;
  /**Recherche des modules élémentaires à créer**/
  Pour tout $i \in A$
    $g := g + 1$;
    agent$(g) := i \cup \delta_u^-(i)$;
  FinPourtout
  $U = [1 : g]$;
  Pour tout $g' \in [1 : g]$
    $U := U - \{g'' \in U | \textbf{agent}(g'') \subset \textbf{agent}(g') \vee (\textbf{agent}(g') = \textbf{agent}(g'') \wedge g'' < g')\}$;
  FinPourtout
  /**Attribution des gains**/
  Pour tout $g \in U$
    Pour tout $j \in$ **agent**(g)
      Si $\delta_u^-(j) \cap$ **agent**$(g) = \delta_u^-(j)$ Alors
        Pour tout $c \in \times_{i \in \textbf{agent}(g)} C_i$
          $u_j^g(c) :=$ **pick**(c, j)
        FinPourtout
      Sinon
        Pour tout $c \in \times_{i \in \textbf{agent}(g)} C_i$
          $u_j^g(c) := 0$
        FinPourtout
      FinSi
    FinPourtout
    $\mathcal{U}' = \mathcal{U}' \cup \{\langle \textbf{agent}(g), u^g\rangle\}$;
  FinPourtout
  return $\langle A, C, \mathcal{U}'\rangle$;

FIG. 3 – Algorithme de recherche des modules élémentaires d'un jeu

4 Conclusion

Dans cet article nous avons proposé la théorie des réseaux de jeux comme outil d'étude de la modularité des réseaux moléculaires et plus généralement des systèmes complexes. La théorie des réseaux de jeux étend la théorie de jeux en permettant la définition d'interactions locales entres agents. Ces interactions locales sont portées par les différents jeux qui constituent le réseau, et sont

Jeu (tableau de gains)

x	y	z	u^x	u^y	u^z
Off	Off	Off	1	2	-1
Off	Off	On	1	2	0
Off	On	Off	-1	0	-1
Off	On	On	-1	0	0
On	Off	Off	0	0	0
On	Off	On	0	0	-1
On	On	Off	3	3	0
On	On	On	3	3	-1

Forme normale

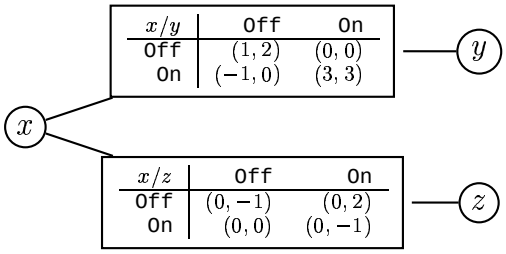


FIG. 4 – Jeu à 3 joueurs et sa forme normale

observées au travers de leur équilibres dits locaux. A l'échelle du réseau complet, ces équilibres locaux se combinent pour former des équilibres globaux.

Une même dynamique complexe peut être représentée par plusieurs réseaux de jeux. Nous nous sommes intéressé à la recherche d'une représentation canonique des réseaux de jeux — la forme normale — où chaque jeu fait participer le moins d'agents possible. Dans la forme normale les jeux sont qualifiés de modules élémentaires et mettent en relation les agents les plus « connectés ». De par leur taille réduite les modules élémentaires devraient être plus compréhensibles que les jeux du réseau de départ, et permettre tout de même d'identifier des structures qui seraient impossibles à caractériser si on ne considérait les agents que pris séparément.

La théorie des réseaux de jeux a été utilisée pour modéliser une partie du système activateur du plasminogène (PAs) intervenant dans la mobilité des cellules cancéreuses. Pour des raisons de place, nous ne développons pas ce travail ici, mais le lecteur pourra se référer à [2] pour plus d'informations. Le réseau de jeux du système PAs, composé de 10 agents biologiques et 6 jeux, a permis de mettre en évidence l'importance de l'inhibiteur de l'activateur du plasminogène (PAI-1) ainsi que l'existence de deux états d'équilibre, un état non migratoire et un état pro migratoire. Ces deux états ont été retrouvés expérimentalement.

Références

[1] A. Barabasi. *Linked : How Everything Is Connected to Everything Else and What It Means*. Plume, 2003.

[2] C. Chettaoui, F. Delaplace, M. Manceny, and M. Malo. Games Network & Application to PAs system. In *Information Processing in Cells and Tissues (IPCAT)*, 2005.

[3] F. Delaplace and M. Manceny. Games network. Technical Report 101-2004, Laboratoire de Méthodes Informatiques (LaMI), CNRS-UMR 8042, Université d'Évry, 2004.

[4] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.

[5] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407 :651–654, 2000.

[6] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge Univ. Press, 1982.

[7] R. B. Myerson. *Game Theory : Analysis of Conflict*. Harvard University Press, 1991.

[8] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*, volume 380. MIT Press, 1994.

[9] E. Segal, N Friedman, N Kaminski, A. Regev, and D. Koller. From signatures to models : understanding cancer using microarrays. *Nature Genetics*, 37(Suppl) :S38–S45, 2005.

[10] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, and D. Koller. Module networks : identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2) :166–176, 2003.

Algorithmes de planification d'expériences pour la détermination de réseaux d'interactions de protéines

Alexis Lamiable et Dominique Barth

Laboratoire PRiSM – Université de Versailles, 45 avenue des États-Unis 78035 Versailles Cedex

1 Introduction

L'identification d'interactions entre protéines était une tâche difficile, jusqu'au développement récent de techniques expérimentales dites à *haut débit* [1]. Produisant de grandes quantités de données, ces techniques permettent de ne plus s'intéresser à des interactions individuelles, mais à des réseaux d'interaction. L'étude de ces réseaux devrait permettre la mise en évidence des propriétés de haut niveau des organismes (comme la résistance aux erreurs dues à des mutations), ainsi que de créer de nouveaux médicaments ciblant des interactions spécifiques [2].

La fiabilité des données provenant des diverses méthodes de détection d'interactions est variable et souvent mal connue [3]. La construction de réseaux d'interactions nécessite donc de disposer d'un modèle permettant d'intégrer ces données hétérogènes en une mesure de confiance. De plus, les expériences *in vivo* ayant un coût, et le nombre de cibles d'expériences possibles étant élevé, il est souhaitable de savoir comment choisir la prochaine expérience à effectuer, afin d'optimiser le gain d'information obtenu. Nous proposons un modèle à base d'inférence bayésienne afin d'exprimer la confiance dans chacune des interactions, et d'étudier différentes stratégies de choix de cibles d'expériences au cours de simulations sur des réseaux aléatoires ainsi que sur le réseau d'interactions de la levure.

2 Modèle et problème

Le réseau d'interactions d'un organisme est classiquement représenté par un graphe non orienté [4], appelé *graphe réel*, dans lequel les sommets représentent les protéines, et une arête représente une interaction entre une paire de protéines. Notre connaissance de ce réseau est également représentée par un graphe, appelé *graphe connu*, possédant le même ensemble de sommets et dans lequel une arête $\{p_1, p_2\}$ est pondérée par une valeur représentant un niveau de confiance dans l'existence d'une interaction entre p_1 et p_2.

Effectuer une expérience de type double-hybride ou de type TAP-MS revient à choisir un sommet, obtenir son voisinage dans le graphe réel, et reporter cette information dans le graphe connu. Toutefois, les expériences produisant des faux négatifs et des faux positifs, le voisinage obtenu peut contenir des erreurs. Nous supposerons que refaire une expérience de même type sur la même cible ne change pas son résultat. En effet, un résultat positif d'expérience de notre modèle correspond en général à une interaction détectée au moins x fois lors d'une série d'expériences de double-hybride, ce qui correspond à nombre de résultats publiés [5, 6].

Si l'on fixe un seuil de confiance au delà duquel une arête est considérée comme *découverte*, le problème du choix des cibles d'expériences peut être formulé de la manière suivante : étant donné un financement pour x expériences, choisir une série de cibles d'expériences qui maximise la quantité d'arêtes réelles découvertes.

3 Approche existante

Le problème du choix d'une cible d'expérience à été abordé par Michael Lappe dans un chapitre de sa thèse [7]. Il considère un modèle simplifié avec un seul type d'expérience, et trois niveaux de confiance : pas d'arête, arête découverte ou arête confirmée. Lorsqu'une interaction est observée pour la première fois entre deux protéines, l'arête est marquée *découverte*. Lorsqu'une interaction est observée une deuxième fois, l'arête est marquée *confirmée* (dans ce modèle, une protéine ne peut être choisie comme appât qu'une fois, et une interaction ne peut donc être observée que deux fois).

Michael Lappe compare une stratégie appelée *pay-as-you-go* à des stratégies témoins (choix aléatoire, choix optimal en connaissant le graphe réel complet...). Cette stratégie exploite le fait que les réseaux d'interaction sont des graphes petit mondes, et cherche à étudier les quelques sommets très connectés le plus rapidement possible, en choisissant comme appât le sommet ayant le plus grand nombre d'arêtes incidentes découvertes. Lors de simulations sur le réseau d'interaction de la levure, en supposant des expériences sans erreurs, il montre qu'après avoir effectué un quart des expériences possibles, on peut découvrir 90 % des arêtes du graphe, et en confirmer environ 40 %. Toutefois, lorsque l'on introduit des expériences avec des erreurs, les performances se dégradent. Avec des taux de faux positifs et de faux négatifs d'environ 40 % (estimation pour le double hybride tirée de [8]), il est impossible de confirmer plus de la moitié des arêtes.

4 Une nouvelle approche

4.1 Intégration de données hétérogènes

Afin d'intégrer les résultats de plusieurs types d'expériences en une unique mesure de confiance, nous avons choisi de modéliser chaque expérience sous la forme d'un *test* avec un taux de faux positifs et de faux négatifs fixés. Un niveau de confiance initial *neutre* est fixé pour chaque interaction possible et représente l'absence de connaissance. La confiance d'une interaction est remise à jour par inférence bayésienne lorsque des tests sont effectués.

Les relations de dépendance conditionnelle entre les différents tests peuvent être représentées sous la forme d'un réseau bayésien. Chaque arête du graphe connu est munie d'un tel réseau, qui reçoit en entrée les résultats des tests effectués sur les extrémités de cette arête, et produit en sortie une valeur de confiance. Ce réseau peut être construit avec l'aide d'un expert, ou appris automatiquement à partir de jeux de données fiables, approche qui a été utilisée efficacement pour prédire des interactions de protéines [9]. Nous avons choisi une troisième approche, qui consiste à utiliser un réseau bayésien dit « naïf », supposant que chaque test est indépendant des autres. Cette supposition peut sembler abusive, mais les réseaux bayésiens naïfs donnent généralement de bons résultats pour un coût en temps de calcul réduit.

4.2 Stratégies utilisant l'inférence bayésienne

Nous avons étudié les quatre stratégies suivantes pour le choix des cibles d'expériences :
– Choix *aléatoire* des sommets

– *Lappe* : une adaptation de la stratégie *pay-as-you-go*, qui consiste à faire la somme des confiances des arête incidentes à chaque sommet, et à choisir un sommet pour lequel cette somme est maximale
– *Bayes* : une stratégie utilisant le réseau bayésien, cherchant à optimiser le gain d'information de chaque expérience. On définit les gains δ_e^+ et δ_e^- comme la différence de confiance sur les arêtes d'un sommet après un résultat positif ou négatif au test e. Le gain espéré pour un sommet et un test donnés est une moyenne pondérée de δ_e^+ et δ_e^-, les coefficients de cette moyenne dépendant de l'objectif fixé. Si l'on cherche à découvrir de nouvelles interactions, on favorisera les réponses positives, et si l'on cherche à réduire le nombre de faux positifs, on favorisera les réponses négatives.
– *Bayes 2* : une deuxième stratégie utilisant le réseau bayésien, similaire à la première, mais qui cherche à maximiser l'écart à la valeur neutre, et non plus le gain d'information.

5 Expériences

5.1 Hypothèses

5.1.1 Données

D'une part, nous avons généré aléatoirement un ensemble de graphes, suivant un processus de génération qui consiste à dupliquer un sommet s_1 en un sommet s_2 ayant les mêmes voisins, à supprimer des arêtes de s_1 ou s_2 avec une probabilité p_d, et à ajouter une interaction entre s_1 et s_2 avec une probabilité p_a. Cet algorithme produit des graphes ayant des propriétés similaires à celles des réseaux d'interaction [10].

D'autre part, nous avons utilisé les données d'interaction de la levure provenant de DIP (*Database of Interacting Proteins*). Nous avons effectué nos simulations à la fois sur l'ensemble de ces données et sur le sous-ensemble CORE, constitué d'interactions vérifiées. Nous présenterons les résultats obtenus sur CORE, dont le graphe contient 2640 sommets et 6378 arêtes.

5.1.2 Tests utilisés

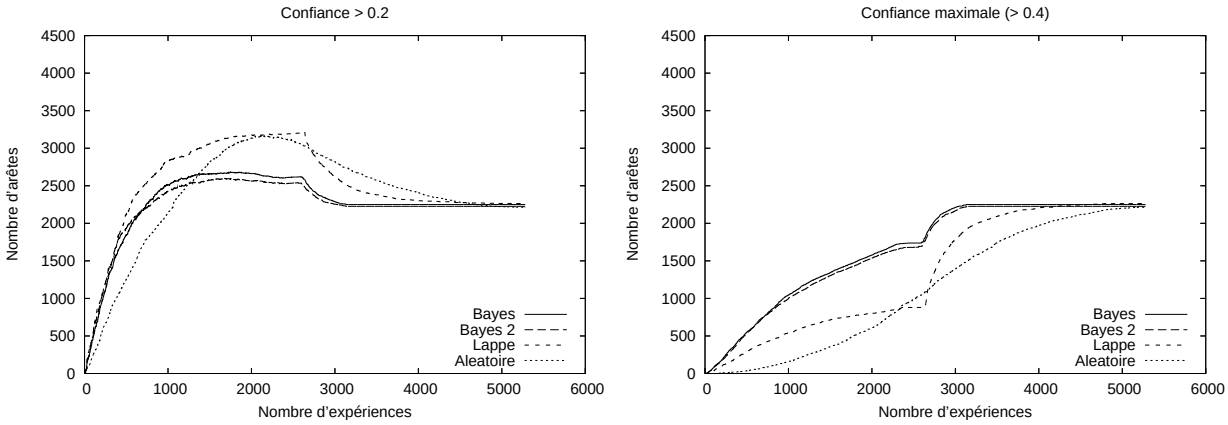
Nous avons distingué trois catégories de tests. Tout d'abord, les *observations biologiques* sont des tests représentant nos connaissances préalables (classification fonctionnelle, complexes de protéines, ...), et sont utilisées pour le calcul initial des niveaux de confiance, avant le début des expériences. Les *tests topologiques* désignent des observations faites sur la structure du graphe connu (et donc incomplet), qui peuvent suggérer de nouvelles interactions. Nous utiliserons un test de voisinage reposant sur le fait que deux protéines interagissant chacune avec une troisième protéine ont plus de chances d'interagir entre elles que deux protéines isolées. Ces tests seront effectués à intervalles réguliers (les effectuer à chaque expérience étant trop coûteux en temps de calcul), et leur fiabilité dépend de notre connaissance du graphe. Enfin, les tests *expérimentaux* correspondent à la simulation d'expériences de double hybride et de TAP-MS, dont les estimations de taux d'erreurs sont tirées de [8] (20 % f.p. et 50 % f.n. pour le double hybride, 20 % f.p. et 30 % f.n. pour TAP-MS). Ce sont ces tests pour lesquels nous désirons proposer une stratégie de choix d'appâts.

5.1.3 Simulations

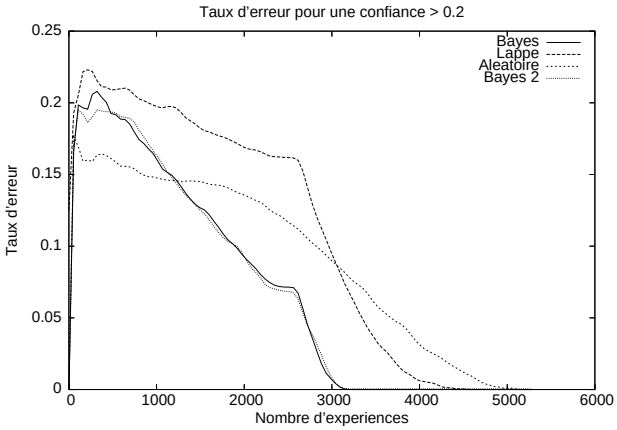
Nous avons effectué plusieurs séries de simulations sur chaque jeu de données. Tout d'abord, nous n'avons utilisé que les tests expérimentaux, puis nous avons introduit une classification fonctionnelle et un test de voisinage. Dans chaque cas, nous avons alterné des séries de 20 expériences de type double-hybride et de type TAP-MS, et poursuivi les simulations jusqu'à ce que chaque expérience possible soit effectuée, pour chaque stratégie (pour chaque type d'expérience, chaque protéine est prise comme cible une fois).

Nous avons considéré trois critères qui permettent d'évaluer la qualité d'une stratégie : le nombre d'arêtes atteignant un seuil de confiance fixé à un instant t, le taux d'erreur pour un seuil de confiance fixé à un instant t, et la séparation entre la distribution des confiances des arêtes réelles et celles des arêtes fausses, c'est à dire avec quelle vitesse la stratégie réduit la superposition de ces distributions en augmentant la confiance des arêtes réelles, et en diminuant celle des arêtes fausses.

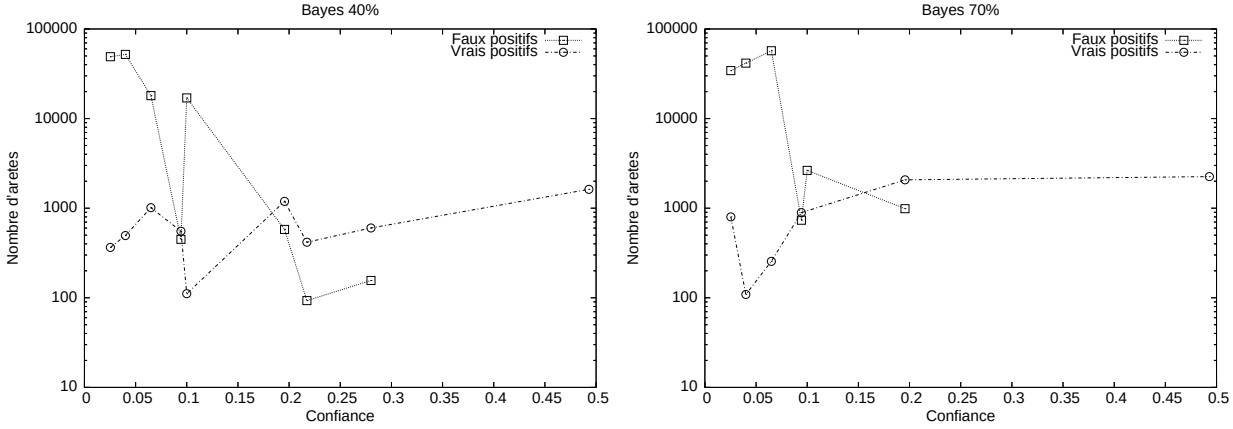
5.2 Résultats



Les figures ci-dessus représentent le nombre d'arêtes atteignant un niveau de confiance fixé, au fil des expériences, pour les quatre stratégies. Pour un seuil de confiance élevé, les stratégies utilisant le réseau bayésien sont plus efficaces que les autres. Après 20 % des expériences possibles, environ 15 % des arêtes réelles ont été découvertes avec un niveau de confiance maximal, contre 7.5 % pour notre adaptation de la stratégie *pay-as-you-go*. Toutefois, pour un niveau de confiance plus bas, l'adaptation de *pay-as-you-go* est plus efficace. Nous pouvons observer un changement brusque de la pente des courbes des stratégies non-aléatoires lorsqu'environ la moitié des expériences ont été effectuées : après avoir exploré une grande surface du graphe, ces stratégies tentent de confirmer des interactions en effectuant une deuxième expérience sur elles. Les résultats positifs provoquent une forte augmentation du niveau de confiance d'une partie des arêtes (à droite), et les faux négatifs provoquent une diminution du nombre d'arêtes ayant une confiance moyenne (à gauche).



Si l'on fixe un seuil de confiance "moyen", correspondant à des interactions détectées par une expérience mais qui restent à confirmer, et que l'on observe le taux d'erreur des arêtes atteignant ce seuil, nous pouvons constater que le taux d'erreur des stratégies bayésiennes baisse plus vite que celui des autres stratégies. Après 20 % des expériences, ce taux devient inférieur à celui de l'aléatoire, et après 60 % des expériences, il est nul.



Les figures ci-dessus représentent l'évolution de la distribution des confiances des arêtes réelles et des arêtes fausses, après 40 % et 70 % des expériences, pour une des stratégies bayésiennes. Une bonne stratégie est une stratégie qui fait rapidement évoluer la courbe des arêtes réelles vers des confiances élevées, et la courbe des arêtes fausses vers des confiances faibles. Le choix d'un seuil de confiance en dessous duquel on considère qu'une interaction n'a pas lieu permet de définir un compromis entre fiabilité et exhaustivité. Dans l'exemple représenté, après 40 % des expériences, un seuil de confiance de 0.25 permet d'obtenir un graphe contenant environ 7 % de faux positifs mais ne contenant qu'un tiers des arêtes réelles.

6 Conclusion et perspectives

Nous avons proposé un modèle permettant d'intégrer des données hétérogènes afin de représenter l'évolution de nos connaissances sur le réseau d'interactions d'un organisme au fil des expériences. Nous avons montré qu'il était possible d'établir des stratégies efficaces de choix de

cibles d'expériences qui ne supposent pas de connaissance particulière de l'organisme étudié, mais qui savent exploiter ces connaissances lorsqu'elles sont disponibles.

Il existe une limite qu'aucune stratégie ne peut permettre d'améliorer, qui est atteinte lorsque toutes les expériences possibles ont été effectuées. Ainsi, dans nos simulations, il est impossible que plus d'un tiers des interactions aient une confiance supérieure à 40 %. Pour dépasser cette limite, il est nécessaire d'avoir une meilleure connaissance des taux d'erreurs des expériences actuelles, et de diversifier les méthodes de détection d'interactions utilisées. De plus, il serait probablement utile de ne plus restreindre une expérience à un résultat binaire, mais par exemple introduire une différence de confiance entre les interactions ayant été détectées deux fois lors d'une expérience de double hybride et celles l'ayant été trois fois.

Cependant, afin d'améliorer les performances du modèle en conservant les mêmes méthodes, il est possible d'abandonner l'hypothèse d'indépendance des tests, et d'utiliser un réseau bayésien plus complexe, élaboré par des experts ou appris à partir de données sûres. Il serait intéressant d'étudier si un réseau plus complexe conduit à une meilleure précision, et si le coût en temps de calcul supplémentaire est acceptable.

Enfin, il est possible d'imaginer de nouvelles stratégies, adaptées à divers objectifs. Parmis les pistes intéressantes, nous pouvons envisager de tenir compte du coût de chaque type d'expérience, et établir une stratégie permettant de choisir à la fois un type d'expérience et une protéine cible. Nous pourrions également tenir compte du fait que les expériences *in vivo* sont faites par séries, et que l'information obtenue n'est disponible qu'à la fin de la série, et non après chaque expérience.

Références

[1] Alain Bernot. *Analyse de Génomes, Transcriptomes et Protéomes*. Dunod, 2001.

[2] G. Apic, T. Ignjatovic, S. Boyer *et al.* Illuminating drug discovery with biological pathways. *FEBS Letters*, 579, 2005.

[3] Lukasz Salwinski et David Eisenberg. Computational methods of analysis of protein-protein interactions. *Current Opinion in Structural Biology*, 13 :377–382, 2003.

[4] Claude Berge. *Graphes*. Dunod, 1970.

[5] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8) :4569–4574, January 2001.

[6] Peter Uetz, Loic Giot, Gerard Cagney *et al.* A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403 :623–632, February 2000.

[7] Michael Lappe. *Novel Algorithms for Protein Interaction Networks*. PhD thesis, University of Cambridge, November 2003.

[8] Chandra L. Tucker, Joseph F. Gera, et Peter Uetz. Towards an understanding of complex protein networks. *TRENDS in Cell Biology*, 11(3), March 2001.

[9] Ronald Jansen, Haiyuan Yu, Dov Greenbaum *et al.* A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302, October 2003.

[10] Manuel Middendorf, Etay Ziv, et Chris Wiggins. Inferring network mechanisms : The *drosophila melanogaster* protein interaction network. *PNAS*, 2004.

Séparation de graphes pour l'identification de voies métaboliques

Antoine Joulie[1], Maria Pentcheva[2], Dominique Barth[1]

1.PRiSM, Université de Saint-Quentin en Yvelines, France
2.LORIA-CNRS, Nancy, France

1 Introduction

L'évolution des techniques expérimentales fournit non seulement des connaissances sur les composants des grands réseaux biologiques (interactions protéines-protéines, métabolisme, ...) mais aussi sur leurs interactions. Ces relations deviennent l'information biologique majeure de ces réseaux et les graphes en sont un modèle de représentation judicieux [10]. A partir de ces réseaux d'interaction, une des problématique biologique est de déterminer des sous-structures (sous-graphes) ayant des fonctions ou des propriétés biologiques identifiées.
Les réseaux métaboliques qui représentent les réactions chimiques intervenant dans le fonctionnement des cellules sont une bonne illustration de cette évolution. Ils peuvent être décomposés en sous-ensembles de réactions participant à une même fonction dans la cellule. On appelle ces sous-réseaux les *voies métaboliques*. Certaines sont déjà très bien connues (le cycle de Kreb, le cycle de citrate, la glycolyse, ...), mais nombre d'entre elles restent encore à déterminer via le réseau métabolique. Il est en effet important de comprendre quelles propriétés topologiques du réseau peuvent caractériser efficacement ces sous-structures biologiquement pertinentes. Une idée naturelle est de les séparer en sous-réseaux potentiellement pertinents [9]. Dans le cas du réseaux de métabolisme, on parle de *reconstruction métabolique* [3].
Notre travail a été réalisé en collaboration avec l'équipe *Biochimie et structure des protéines* de l'INRA de Jouy-en-Josas qui étudie le protéome de la bactérie Lactococcus lactis (L.lactis). L'étude de plusieurs souches de LLa élevées dans des conditions expérimentales différentes nous fournit, grâce aux protéines présentes dans les cellule de la bactérie, des liste de réactions chimiques ayant eu lieu. Ces listes nous permettent de pondérer les réseaux métaboliques et ainsi d'orienter les algorithmes de séparations que nous utilisons.
Les réseaux sont modélisés ici par des hypergraphes orientés dans lesquels les sommets correspondent au composants chimiques et les hyperarcs modélisent les réactions chimiques. L'algorithme que nous utilisons détermine des *séparateurs-sommets [7]* sur les hypergraphes. Il les sépare en deux sous-hypergraphes disjoints de tailles équivalentes en supprimant un nombre minimum de sommets. Ces deux sous-hypergraphes sont eux même séparés. Le processus s'arrête lorsque nous obtenons des parties de tailles équivalentes aux plus petites voies métaboliques connues (une dizaine de réactions).

L'objectif de ce travail est de comprendre le fonctionnement du réseau métabolique de L.lactis sous deux angles différents. Premièrement, la séparation nous permet de faire ressortir des ensembles de réactions participant aux mêmes fonctions dans la cellule. Deuxièmement, la séparation hiérarchique nous informe sur l'arrangement hiérarchique des différentes voies métaboliques à l'intérieur du réseaux.

2 Modélisation

Avant de décrire la modélisation des réseaux de métabolisme, nous introduisons la notion d'hypergraphe orienté. Soit un ensemble $S = \{s_1, s_2, \ldots, s_n\}$ et une famille $E = \{e_1, e_2, \ldots, e_m\}$ où chaque élément e_i est un de couples de partie non vides disjointes de S de la forme $e_i = (e_i^-, e_i^+)$. On dit que (S, E) constitue un *hypergraphe orienté* où S est l'ensemble des sommets et où chaque e_i est un hyperarc. On notera l'hypergraphe $H(S, E)$ ou $H(\{e_1, e_2, \ldots, e_m\})$ (le lecteur pourra se référer à [2] pour plus d'information sur les hypergraphes)
La notion d'hypergraphe nous permet de modéliser les réseaux métaboliques, composés d'un ensemble de réactions chimiques. Une réaction chimique est définie par un ensemble de composés chimiques en entrée (les *substrats*), un ensemble de composés chimiques en sortie (les *produits*) et un ensemble de protéines (ou enzymes) catalysant la réaction. Il y a réaction lorsque sous l'influence des enzymes, les substrats sont transformés pour donner les produits. Le réseau métabolique est modélisé par un hypergraphe orienté et étiqueté sur les hyperarcs tel que chaque réaction est représentée par un hyperarc. Considérons une réaction contenant un ensemble $\{s_1, \ldots s_s\}$ de substrats, un ensemble de produits $\{p_1, \ldots p_p\}$ et qui est catalysé par un ensemble d'enzymes $\{e_1, \ldots e_e\}$ Pour chaque réaction de ce type, l'hypergraphe contient un hyperarc (u^-, u^+) où $u^- = \{s_1, \ldots s_s\}$ et $u^+ = \{p_1, \ldots, p_p\}$. Il est étiqueté par l'ensemble des enzymes $\{e_1, \ldots, e_e\}$.
Nous utilisons donc une représentation des réactions dans laquelle chaque composant chimique est un sommet de l'hypergraphe. Il peut être substrat dans une réaction et produit dans une autre. Chaque réaction est donc un hyperarc orienté (ie. un hyperarc dans lequel les sommets correspondant aux substrats (u^-) sont séparés des sommets correspondant aux produits (u^+)).

3 Données

3.1 Hypergraphe de métabolisme d'un organisme

La base de donnée en ligne *KEGG* [8] contient toutes les réactions chimiques pouvant se produire dans les cellules d'organismes vivants (végétal, animal ou bactérien). Le regroupement de l'ensemble de ces réactions forme le réseau métabolique complet que nous modélisons par ce que l'on appelle l'hypergraphe *complet*. Cette base de donnée fournit également pour une centaine d'organismes l'ensemble des protéines intervenant dans les cellules de chacun. A partir de cet ensemble de protéines, il est possible d'induire un ensemble de réactions ayant lieu potentiellement dans un organisme donné.
Nous nous intéressons ici à la bactérie Lactococcus lactis. Nous avons donc isolé l'ensemble des protéines correspondant à cet organisme. A partir de l'hypergraphe complet et de l'ensemble des réactions catalysées par ces protéines, nous déduisons un hypergraphe en ne retenant que les hyperarcs correspondant aux réactions identifiées. Nous l'appelons hypergraphe *théorique* du L.lactis.

Il contient en effet toutes les réactions qui peuvent théoriquement avoir lieu dans les cellules de l'organisme.

3.2 Prise en compte d'expérience

Nous avons vu que pour chaque organisme est déterminé un hypergraphe théorique. On peut aussi vouloir tenir compte de résultats expérimentaux. Prenons l'exemple de L.lactis. Deux souches de L.lactis ont été élevées dans deux milieux de culture différents et leur croissance a été arrêtée à trois stades distinct. Nous avons identifié les protéines présentes dans les cellules de L.lactis. Chaque expérience fournit une liste de protéine, il y en a 12 au total.
Dans le but d'orienter nos algorithmes de séparation, nous introduirons des informations biologiques dans les hypergraphes sous forme de pondération. L'idée est que si deux réactions ne se produisent jamais dans les même conditions (souche/milieu/croissance), elles ont peu de chance de participer à une même fonction. Inversement, deux réactions se produisant souvent ensemble ont plus de chances de participer à la même fonction de l'organisme. A partir des ensembles de protéines fournis par les expériences, nous calculons une matrice de corrélation entre les réactions du réseau. A chaque couple de réaction, nous affectons une valeur allant de 0, si les réaction ne se produisent jamais dans les mêmes conditions, à 12, si les deux réactions apparaissent à chaque fois ensemble. Nous verrons au paragraphe 4.2 comment cette matrice de corrélation intervient dans la séparation.

4 Techniques de séparation de graphes

4.1 Principe

Étant donné un graphe $G = (V, E)$ (resp. un hypergraphe orienté $H = (S, E)$), un *k-séparateur* de G (resp. H) est un ensemble de sommets (séparateur-sommet) ou d'arêtes (séparateur-arête) dont la suppression laisse k sous-graphes (resp. sous-hypergraphes) sans aucune arête les reliant. Un 2-séparateur arête est aussi appelé un bissecteur. Il a été montré que déterminer un séparateur (sommet ou arête) de cardinalité ou de poids minimal est un problème difficile dans les graphes comme dans les hypergraphes et qu'il n'est pas approximable [4]. De nombreuse approches heuristiques ont donc été conçues pour résoudre ces problèmes dont les plus connues sont :
- L'approche combinatoire qui est basée sur l'algorithme de Keringhan et lin [7].
- L'approche géométrique qui utilise la localisation géométrique des sommet et demande donc de connaître les coordonnés de chaque sommet.
- La séparation spectral qui utilise les vecteurs propres de la matrice de Laplace représentant le graphe [12].
Une dernière approche, appelé multi-niveaux consiste à contracter le graphe pour en réduire la taille et ainsi séparer un graphe de taille raisonnable par l'une des méthodes précédentes. Une phase d'expansion est ensuite nécessaire pour retrouver le graphe d'origine [1].

4.2 Algorithme choisi

Chercher un séparateur-sommet d'un hypergraphe revient à chercher un séparateur arête dans le *linegraph* de l'hypergraphe. Soit un hypergraphe $H = (S, \{e_1, \ldots, e_n\})$. Le *linegraph* $G = (V, E)$ de H est le graphe où chaque sommet $e_1, \ldots e_n$ correspond respectivement à un hyperarc

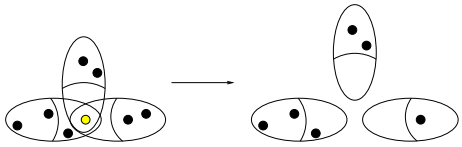


Figure 1: Lorsque l'on met le sommet jaune dans le séparateur, les trois hyperarêtes sont deux à deux déconnectées

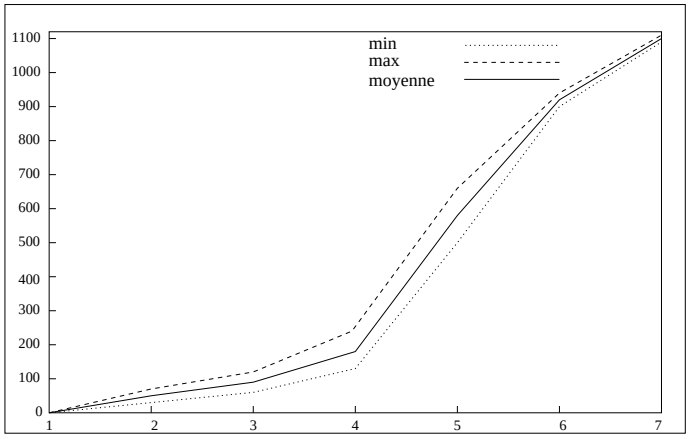
$E_1, \ldots, E_n$ de H et tel que deux sommets e_i et e_j sont adjacent si et seulement si $(e_i^- \cup e_i^+) \cap (e_j^- \cup e_j^+) \neq \emptyset$. Cette transformation a l'avantage de permettre une séparation plus fine. En effet, lorsque au moins trois hyperarcs partagent un même sommet, si ce sommet est mis dans le séparateur, les trois hyperarcs se trouvent deux à deux déconnectés (voir figure 1). L'utilisation du *linegraph* permet d'éviter ce cas et de séparer un hyperarc des deux autres. Le *linegraph* obtenu est pondéré sur les arêtes avec la matrice de corrélation calculée au paragraphe 3.2. Nous rappelons que cette matrice contient les corrélations entre tous les couples de réactions. Les sommets du *linegraph* représentent les hyperarcs, donc les réactions du réseau. Ainsi, les arêtes du *linegraph* modélisent un (ou plusieurs) composant commun entre les deux réactions. Plus la corrélation entre deux réactions est forte, plus le poids de l'arête joignant les deux sommets est élevé. L'algorithme utilisé cherche donc un séparateur de poids minimal dans un graphe.
Le poids d'un séparateur arête dans un graphe est la somme des poids des arêtes qui le composent. Considérons un séparateur arête S du graphe G. Soit v un sommet. Nous définissons le gain de v, noté $g(v)$, comme la différence entre le poids du séparateur S actuel et le poids du séparateur obtenu en changeant le sommet v d'ensemble. L'algorithme que nous utilisons est une adaptation de l'algorithme de Kerighan et Lin. Il fonctionne en deux phases. La première phase retourne une première bissection avec un algorithme glouton. Puis la procédure BKL (Boundary Kerighan-Lin) mise au point dans [7] améliore cette première solution. Le schéma fonctionne sur une bissection hiérarchique du graphe de départ. Une première bissection est effectuée qui donne deux sous-graphes disjoint. Une bissection est effectuée sur ces deux nouveaux graphes et ainsi de suite jusqu'à ce que les graphes obtenus contiennent moins de 10 sommets. Le sommet de départ de l'algorithme (choisi au hasard) ayant un impact non négligeable sur la qualité de la solution renvoyée, nous effectuons 10 fois ce schéma.

5 Application

5.1 Résultats topologiques

Pour chaque niveau de la séparation hiérarchique nous relevons le minimum et le maximum et la moyenne du poids des séparateurs pour les dix tests (voir Figure 2-1) pour la courbe). Le résultat principal qui découle de l'analyse de cette courbe est un point d'inflexion très important au niveau 4 de notre séparation hiérarchique. Après quatre séparations successives le poids du séparateur augmente plus fortement. Cette cassure correspond au niveau où l'algorithme n'a plus d'autre choix que de casser des arêtes de fort poids pour continuer la séparation. Deux hypothèses sont avancées pour comprendre cette cassure. Premièrement, la structure petit-monde du *linegraph* en serait la cause directe. Une étude plus fine est en cours sur la structure topologique du *linegraph*

1)



2)

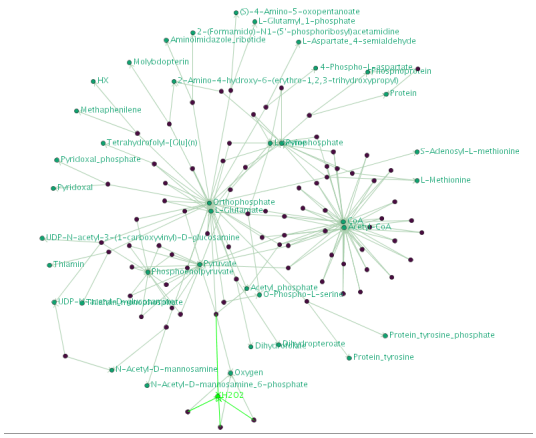


Figure 2: 1) Le poids Min, Max et $Moyen$ des séparateurs en fonction de l'étage de la séparation. 2) L'hypergraphe correspondant à la voie du métabolisme du Glutamate affichée avec Biolayout

pour comprendre ce phénomène. Deuxièmement, la pondération que nous utilisons introduirait trop de disparités, ce qui l'emporterait sur l'aspect topologique du graphe. En effet, nous remarquons que lors des quatre premiers niveaux de séparation, l'algorithme ne met dans le séparateurs que des arêtes de poids très faible (1 à 3). Ainsi, à partir du niveau 4 les arête mises dans le séparateurs sont de poids élevé. Plus d'expériences apporteront plus d'équilibre dans les pondérations tout en accentuant plus les vrais corrélations entre réactions.

5.2 Résultats biologiques

Le schéma de séparation hiérarchique décrit au paragraphe 4.2 retourne une séparation du *linegraph* en quelques centaines d'ensembles. Nous avons dans un premier temps vérifié la validité de notre séparation. Pour cela nous avons retrouvé dans les ensembles séparés des voies métaboliques déjà connues. Pour chaque ensemble séparé du *linegraph*, nous reconstruisons l'hypergraphe correspondant. Nous détectons les hypergraphe dont plus de 50% des réactions le composant appartiennent à une même voie métabolique de KEGG (condition 1). Si de plus, dans ces réactions, au moins 50% sont catalysées par des enzyme apparues dans les expériences du paragraphe 3.2, alors cet hypergraphe est un bon candidat et correspond donc à une voie métabolique de L.lactis. Nous avons dégagé une dizaine d'hypergraphes respectant la condition 1. Parmi ceux-ci, seuls deux respectent aussi la condition 2. Un correspond à la voie du métabolisme du Glutamate de KEGG (voir la figure 2-2). L'autre candidat ne correspond pas à une voie métabolique connue de KEGG. Les autres hypergraphe issus de la séparation sont en cours d'analyse pour déterminer les parties intéressantes.

6 Conclusion

L'élaboration des hypergraphes complet et de l'hypergraphe théorique de L.lactis rencontre une difficulté. En effet, les données de KEGG sont encore incomplètes. Certaines enzymes que l'on

s'attendrait à voir apparaître chez L.lactis n'ont pas encore pu être détectées. Les hypergraphes que nous utilisons dans ce papier sont donc incomplets. De plus, certaines enzymes fournies dans les jeux de données expérimentales ne pourrons être associées à aucune des réactions du graphe théorique de L.lactis. Ces protéines sont en fait les protéines dont on ne connaît ni la fonction ni la réaction qu'elles catalysent. Il faut tenir compte de cela lors de l'analyse.
Les résultats obtenus par notre algorithme sont encourageants du point de vue de la reconstruction métabolique, dans la mesure où quelques voies métaboliques ont été retrouvées. Cependant, on peut espérer qu'un réglage plus fin des algorithmes nous permettrait d'en retrouver d'autres. En particulier, nous adapterons notre algorithme à la structure en petit monde des hypergraphes sur lesquels nous travaillons. De plus, les pondérations du *linegraph* sont issues d'une série de 12 expériences. Ce nombre s'avère trop faible pour avoir un impact biologique. De telles expériences sont couteûses et difficiles à mettre en oeuvre à plus grande échelle. Il serait donc utile d'utiliser les données sur le transcriptome de L.lactis mises à disposition sur internet par [11]. Sans rentrer dans le détail, ces données sont moins "fines" que nos données expérimentales mais elles sont disponibles en plus grand nombre. De plus, le transcriptome d'autres organismes est disponible. Ce qui nous permettra de comparer plusieurs séparations de réseaux métaboliques.

References

[1] Cleve Ashcraft and Joseph W.H. Liu. Using domain decomposition to find graph bisectors. Technical report, 1995.

[2] Claude Berge. Two theorem in graph theory. *Proceeding of the National Academy of Sciences*, 3:842–844, 1957.

[3] Frédéric Boyer. *Reconstruction ab-initio des voies métaboliques- Formalisation et approche combinatoire*. PhD thesis, Université Joseph Fourier, Grenoble, France, 2004.

[4] Thang Nguyen Bui and Curt Jones. Finding good approximate vertex and edge partitions is np-hard. *infor. Proces. Lett.*, 42(3):153–159, 1992.

[5] Ildefonso Cases, Anton Enright, and Leon Goldovsky. http://maine.ebi.ac.uk:8000/services/biolayout/.

[6] George Karypis. Multilevel algorithm for multi-constraint hypergraph partitionig. Technical report, 1999.

[7] George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. Technical report, 1995.

[8] http://www.genome.ad.jp/kegg/kegg2.html.

[9] Hongwu Ma and An-Ping Zeng. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatic*, 19:270–277, 2003.

[10] Hiroyuchi Ogata, Wataru Fujibuchi, Susumu Gato, and Minoru Kanehisa. A heuristic graph comparison algoithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 28:4021–4028, 2000.

[11] ExpressDB Information Page. http://salt2.med.harvard.edu/expressdb/.

[12] A. Pothen, H. Simon, and K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Math. Anal. Appl.*, 11(3):430–452, 1990.

Algorithms in graph partitioning for interaction network analysis

Alain Guénoche

IML-CNRS, 163 Av. de Luminy, 13288 Marseille cedex 9, guenoche@iml.univ-mrs.fr

Abstract

We first describe four recent methods to cluster vertices of an undirected connected graph. The three first ones are based on very different principles, and the last one is a combination of classical ideas in clustering by optimization. We compare these methods according to their ability to recover classes initially introduced in random graphs.

1 Introduction

The partitioning problem in graphs has a long history we don't detail here. It becomes important in practice with the pagination of electronic circuits and VLSI design (Alpert and Kang, 1995). There are also many contributions linked to the graph drawing problem (Batagelj *et al.*, 1999). The aim of these clustering processes was initially to minimize the number of inter-class edges. It has been reactivated these last years in three domains :

- biological problems modelled by graphs (Barabasi, 2000, Bader and Hogue, 2003);
- the study of large networks, like WEB (Moody, 2001), or
- the definition of *communities* in social networks (Girvan and Newman, 2002).

In these domains, the aim is to join together vertices sharing a large number of edges, making some *high density zones*, compared to the percentage observed in the whole graph.

Several ideas appeared these last years and lead to different algorithms.

- The first one is based on a density value associated to each vertex, which is as large as there are many links in its neighborhood (Colombo et al, 2003).
- The second one is based on a dynamical weight function (called *betweenness* in the article), associated to the edges and a pruning of the graph until disconnection (Girvan, 2001).
- The third one is the result of random walks from any vertex in the graph, which is denoted *Markov clustering (MCL)* by the author (van Dongen, 2000).
- Finally, the fourth one is a classical strategy in clustering, optimizing a criterion, adapted to the graph partitioning problem.

In this text we compare these four methods testing their ability to recover high density zones in a graph, when they exist. For that we realize simulations generating random graphs where some classes have been introduced

Let X be a set of n vertices, E the set of m edges and $\Gamma = (X, E)$ the corresponding graph. It is assumed to be connected ; otherwise its different connected components are handled separately. For any part Y of X, let $\Gamma(Y)$ be the set of vertices out of Y that are adjacent to Y

$$\Gamma(Y) = \{x \in X \setminus Y \text{ such that } \exists y \in Y, (x, y) \in E\},$$

and $\overline{\Gamma(Y)} = Y \bigcup \Gamma(Y)$. The neighborhood of x is $\Gamma(x)$. The degree of vertex x is denoted $Dg(x) = |\Gamma(x)|$ and let δ be the maximum degree in the graph. The internal edges of a class $Y \subset X$ is denoted

$$E(Y) = \{(x, y) \in E \text{ such that } x \in Y \text{ and } y \in Y\}.$$

2 Clustering by density

Clustering methods based on density have been introduced by Wishart in 1976; the idea was to build classes around elements having many neighbors in a threshold graph associated to a distance on X. They have not been largely used because, the simple degree was the only proposed density function, and it gives poor results, even if so, non convex or nested clusters can be recovered. Recently, several authors - Rougemont & Hingamp (2003) and for simple graphs, Guénoche (2004) for distance arrays - have reactivated this approach.

2.1 Density function

A density function De is a map from X to R_+ varying increasingly with the number of vertices close to an element. In Colombo et al. (2003) several functions based on the percentage of edges in the neighborhood of a vertex have been compared. The core index introduced by Seidman (1983) is another promising density function. A vertex x has a core value equal to k if there are k vertices in $\Gamma(x)$ having also a core index larger than or equal to k, and k is a maximum for this property. A maximal clique of p elements has core $p-1$ and all the vertices of a tree have core 1. The core index can be calculated recursively, pruning the graph from a vertex of minimum degree, and a $O(m)$ algorithm has been proposed by Batagelj and Zaveršnik (2001).

All the simulations we realized with this kind of density functions are disappointing, mainly because the density values are not spread enough, especially for the core index. For this latter, the set of nodes having the maximum core value in not included in one of the classes introduced in our random graphs. To recover the satisfying results we get from distance matrix, we first evaluate the Sczekanovski-Dice distance between vertices.

$$D(x,y) = \frac{|\Delta(\overline{\Gamma}(x), \overline{\Gamma}(y))|}{|\overline{\Gamma}(x)||\overline{\Gamma}(y)|}$$

where Δ denotes the symmetrical difference between two sets. We retain this local distance because it is very accurate for graphs since two vertices having no common adjacent vertex are at maximum distance value (equal to 1.), and consequently it can be computed in $O(n\delta^2)$.

The density function in x is then defined from the average distance values between x and its adjacent vertices

$$De(x) = 1 - \frac{\sum_{y\in\Gamma(x)} D(x,y)}{Dg(x)}.$$

2.2 Clustering Algorithm

The algorithm is progressive, in two parts :

In the first one, we consider the *local maximum values* of the density function to identify the *seeds* of the classes. A seed, denoted S, can be a singleton or several connected vertices sharing the same density value. When the number of classes is fixed, the highest values are retained or, if they are not enough, some non connected vertices with the largest density values are added. When it is free, only seeds having a density larger than or equal to the average $\overline{De}$ are kept. So the number of classes denoted by p can be given or not to the algorithm.

Then, the seeds are extended recursively through out the addition of connected vertices having a density larger than the average. We assign to each seed all the vertices adjacent to only one seed. Doing so, we avoid any ambiguity in the assignment, postponing the decision when several are possible.

In the second part, these seeds are extended to make a partition $(Q_1, ..Q_p)$. Each remaining element is assigned to one class maximizing criterion C_i :

$$C_i(x) = \frac{\Gamma(x) \bigcap Q_i}{Dg(x)}.$$

Compared to the other methods, this algorithm is very efficient. To find the local maximum density values is in $O(n\delta)$. It allows the treatment of large graphs, which is essential in the biological context.

3 Disconnecting the graph

The main idea of this method has been given by Newman (2001). It consists in an iterative procedure in two steps :

- evaluate the weight $B(x, y)$ of an edge (x, y) as the number of shortest paths, between any two vertices, passing through (x, y);
- eliminate the edge having the greatest weight.

The weight of an edge can be evaluated by function $B : E \mapsto N$

$$B(x, y) = |(z, t) \in X^2 \text{ such that } L(x, y) = L(x, z) + 1 + L(t, y)|$$

where $L(x, y)$ denotes the length (number of edges) of a shortest path between x and y. It is at least 1, that can also be its largest value, as for cliques. Other functions, taking into account the number of shortest paths between any two vertices can be tested.

It is easy to understand that when there are high density zones, the paths between them receive the largest weights. It suffices to delete the corresponding edges to obtain classes.

So this method is a pruning algorithm removing edges to establish clusters as connected components. Clearly, these clusters are nested and a complete divisive hierarchy can be established. To use it as a partitioning algorithm, the number of classes must be given. If only p classes are searched, it is easy to stop the procedure when there are at least p components, but it could be necessary to perform $O(n^2)$ steps to get the first subdivision. And the practical problem is the time complexity of this procedure : the author claims a $O(mn)$ algorithm to evaluate the weights of all the edges, but there are $O(m)$ steps for a complete hierarchy !

Observing, when there are few connections between what will become separate clusters, that removing edges one by one does not modify the classes, we adopt the following strategy to eliminate several edges in the same step :

- evaluate the betweenness function for the edges ; let $Bmax$ be its maximum value ;
- build a minimum spanning tree of this weighted graph ; Let $Lmax$ be the length of its longest edge.
- remove all the edges longer than or equal to threshold $(Lmax + Bmax)/2$.

With the $Lmax$ threshold, the graph would be disconnected in one step, but this could give an uncorrect subdivision. With value $(Lmax + Bmax)/2$, the split appears when $Bmax = Lmax$ and the number of steps is largely reduced. But the average time to establish a partition remains very much longer than the other methods described here.

4 Markov clustering (MCL)

The idea of this method is based on the following principle : performing a random walk (the next edge from vertex x is selected at random in $\Gamma(x)$) from a vertex belonging to a high density zone, leads to stay in the same zone, and the probability to reach another one, after a large number of steps, is very small. Because the next adjacent vertex is selected at random, and as the probability sum to reach them is equal to 1, this walk can be described as a Markov process.

4.1 From the adjacency matrix to a stationary process

Let A be the adjacency matrix of graph Γ ($A(x, y) = 1$ iff $(x, y) \in E$). It is well known that multiplying the adjacency matrix of a graph by itself determines the number of paths of length 2, and so on when the power increases. To ovoid parity dependence on the path length, all the loops are added ; if I denotes the identity matrix $(A + I)$ is used instead of A.

The Markov matrix M associated to Γ is defined by $M(x, y) = \frac{(A+I)(x,y)}{(Dg(y)+1)}$. It is column stochastic, and can be interpreted as each node (a column y) is equally attracted by its neighbors (row x such that $M(x, y) > 0$). According to the author, a scaling operator S_2 applied to matrix M is performed column after column. It has been introduced to reinforce the attracting strength of a row (it preserves the values ordering) and to maintain a stochastic matrix. Acting on column y, it evaluates $S^2(y) = \sum_{x \in X} M(x, y)^2$ and replaces $M(x, y)$ by $S_2(M) = \frac{M(x,y)^2}{S^2(y)}$.

One iteration of the MCL algorithm (van Dongen, 2000) is to multiply matrix M by itself and to apply the S_2 normalization. As any Markov matrix the powers M^k have a limit and the algorithm stops when two consecutive matrices are identical.

4.2 Attractors and classes

The result is a stochastic idempotent matrix M. Often, in a column y there is one element x for which $M(x, y) = 1.0$ and all the other values are equal to 0.0. Element x is said to be the *attractor* of y and, in that case, x is also the attractor of x. Sometimes, there is a value in column y which is close to 1.0 and the complementary part denotes the attraction of another element. Rarely, the attraction is equally shared by several elements to constitute a class of attractors.

In this method, an attractor (or a set of equilibrate attractors) plus the set of attracted elements constitute a class. The number of classes is unpredictable and cannot be specified. It seems to produce a large number of classes when the rate of edges in the graph is low.

5 An optimization method

To apply an optimization procedure, a wanted number of classes p and a function $\Im$ defined from the metric are needed ; the Sczekanovski Dice distance has been retained. We try to optimize $\Im$ over the set of all the partitions of X in p classes. We establish an algorithm very close to the famous $Kmeans$ iterative strategy.

An initial partition is given and the center of the classes are computed. Then all the objects are assigned to the closest center, and the new centers are re-calculated until they remain unchanged. This algorithm has been designed for points distributed in an euclidian space, and the center of a class is just the barycenter of its points. It gives very good results when the function to optimize corresponds to an inertia, for instance the sum, over all the elements, of the square distances to the center of their class. More, it converges very quickly.

Here, for a distance array, there is no barycenter. We retain as center the median element of the class, the one for which the sum of its distance values to the others is minimum. And as function

$\Im$ the sum of squares of the average distance to the elements of its class :

$$\Im = \sum_{x \in X} (\frac{\sum_{y \in class(x)} D(x,y)}{|class(x)|})^2$$

The iterative procedure, i.e. assigning elements to centers and re-computing the median elements, continues as long as the $\Im$ function decreases. At the end a Tabou-search heuristic is performed, trying to put one element into another class without changing the number of classes. After n unsuccessful trials to decrease $\Im$, the best partition is kept. Surprisingly, the inertia criterion give better results than a simple density criterion as the percentage of internal edges in the classes. according to simulations, this algorithm overpasses several others optimization strategies.

To apply this method the number of classes is required as an initial partition. When the number of class is given, a random partition is used as the initial one. But when it is not, we use the first step of the density algorithm to get a number of class equal to the number of seeds, and the seeds as the initial partial partition. So, we combine the density and the optimization strategy in the following procedure :

1. From graph $\Gamma = (X, E)$, compute the Czekanovski-Dice distance on X;
2. Evaluate the density function appropriate to distance array D ;
3. Select the initial classes as the seeds of the density algorithm (section 2);
4. While the inertia criterion $\Im$ decreases
   - Determine the center of each class as its median vertex ;
   - Assign each element to its closest center according to D;
5. Apply a Tabou search heuristic to improve the criteria, as long as it decreases.

6 Validation by simulations

In order to evaluate the ability of these methods to detect high density zones, we try to recover classes initially introduced in a graph. We first develop a generator of random graphs in which there are clusters having more edges between elements in the same class - the internal edges - than between elements in separate classes - the external ones. For that two parameters, p_i and p_e fixing the probabilities of internal and external edges are given. The initial partition is denoted P and each algorithm built for each trial another partition Q, having not necessarily the same number of classes.

In order to compare these four algorithms, we evaluate how far is Q from P using four criteria. The three first ones are evaluated comparing the classes of Q to those of P and the last one is based on an editing distance between partitions, the minimum number of transfers (of one element from its class to another) necessary to turn P into Q.

The results will be detailed during the lecture.

Acknowledgements

This work has been realized with the help of ACI-IMPBio.

7 Bibliography

C.J. ALPERT AND A. KANG (1995) Recent direction in netlist partitioning : a survey, *Integration : the VLSI Journal*, 19, 1-2, 1-81.

G.D. BADER AND C.W. HOGUE (2003) An automated method for finding molecular complexes in large protein interaction networks, *BMC Bioinformatics*, 4, 2, 27 p.

L. BARABASI (2000) The large-scale organization of metabolic networks, *Nature*, 407, 651-654.

V. BATAGELJ AND M. MRVAR (1999) Partitioning approach to visualisation of large graphs, *Lecture Notes in Computer Science* 1731, Springer, 90-97.

V. BATAGELJ AND M. ZAVERŠNIK (2001) An $O(m)$ algorithm for Cores Decomposition of Networks, *submitted*.

S. VAN DONGEN (2000) Graph Clustering by Flow Simulation, *PhD thesis*, University of Utrecht.

T. COLOMBO, A. GUÉNOCHE, Y. QUENTIN (2003) Looking for high density areas in graph : Application to orthologous genes, JIM, 203-212.

M. GIRVAN AND M.E.J. NEWMAN (2002), Community structure in social and biological networks, *Proc. Natl. Acad. Sci. USA*, 99, 7821-7826.

A. GUÉNOCHE (2004) Clustering by vertex density in a graph, Proceedings of IFCS congress *Classification, Clustering and Data Mining Applications*, D. Banks et al. (Eds.), Springer, 15-24.

J. MOODY (2001) Identifying dense clusters in large networks, *Social Networks*, 23, pp. 261-283.

M.E.J. NEWMAN (2001), Scientific Collaboration Networks : Shortest paths, weighted networks and centrality, *Phys. Rev.*, 64.

J. ROUGEMONT AND P. HINGAMP (2003) DNA microarray data and contextual analysis of correlation graphs, *BMC Bioinformatics*, 4:15.

S.B. SEIDMAN (1983) Network structure and minimum degree, *Social Networks*, 5, pp. 269-287.

D. WISHART (1976) Mode analysis : generalization of nearest neighbor which reduces chaining effects, *Numerical taxonomy*, Academic Press, 282-311.

BioΨ langage de description de données fonctionnelles

Pierre Mazière[1] & Franck Molina[2]

[1]EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK
[2]Centre de Pharmacologie et Biotechnologie pour la Santé, UMR 5160, 15 av. Charles Flahault, BP 14491, 34093 Montpellier Cedex 5, France

Résumé

L'utilisation des informations liées aux processus biologiques dans le contexte de logiciel de simulation dépend de leur accessibilité. Malgré le succès de GeneOntology, ces données restent encore pour la plupart enfouies dans la littérature ou les bases de données, sans possibilité d'être exploitées de manière automatisée. Le formalisme défini par le langage de description BioΨ tente de combler cette lacune tout en offrant de nouvelles voies d'analyse des processus biologiques associés aux molécules du vivant.

Introduction

La description fonctionnelle des molécules du vivant utilise des données aussi nombreuses qu'hétérogènes. La dispersion de ces données dans la littérature nécessite de rassembler plusieurs articles couvrant une large période d'étude pour avoir une idée des processus impliquant une molécule particulière dans un contexte précis. Loin de se limiter à une simple énumération d'informations quantitatives ou qualitatives, cette description implique une formulation en langage naturel dont la complexité est le principal frein à un accès et un traitement automatisés de la littérature. Les bases de données sont censées regrouper ces informations pour en offrir une synthèse dont la complétude laisse parfois à désirer. Ce défaut d'information, qui peut se justifier par un choix éditorial, est comblé par des références à d'autres ressources, et/ou la désignation d'un expert de la molécule en question. Les formats utilisés par ces bases de données pour rendre accessibles leurs informations fonctionnelles sont peu nombreux: langage naturel, classifications (ex. Enzyme Commission [1, 2], Transporter Classification [3, 4]), taxonomies (FunCat [5], MetaCyc [6]) et ontologie (GeneOntology [7]). Cette faible hétérogénéité est suffisante pour limiter les possibilités de traitements automatiques de ces données par des outils informatiques appropriés.Les champs rédigés en langage naturel génèrent les mêmes difficultés que celles rencontrées dans les articles de la littérature. Les trois autres formats peuvent être regroupés sous l'appellation de vocabulaire dirigé. Leur utilisation autorise l'étiquetage d'une molécule avec un terme dont la définition se rapporte à un processus biologique. Même si la grille de lecture offerte par la classification, la taxonomie ou l'ontologie dont il est issu apporte une contexte sémantique supplémentaire, l'information associée à ce terme est bien souvent d'une portée limitée. Dans un domaine autre que les bases de données, les langages de descriptions de modèles biologiques sont tout aussi appropriés pour représenter la connaissance fonctionnelle associée à une molécule. Parmi ceux ci, deux se distinguent particulièrement de part leur utilisation et leur mise en relation avec des logiciel de simulation et de modélisation. Le Systems Biology Markup Language [8, 9] (SBML) et le Cell Markup Language [10] (CellML) ont tous deux pour objectifs d'une part de faciliter les intéractions entre les logiciels de simulation et de modélisation, et d'autre part d'éliminer les difficultés liées à la publication de modèles sous forme mathématique dans la littérature dont l'implémentation n'est pas toujours transposable d'un système informatique à un autre. Ces deux langages basés sur l'Extensible Markup Language [11] (XML) ont chacun leurs avantages et leurs inconvénients. Tous deux utilisent MathML [12] pour représenter les équations mathématiques sur lesquels reposent les modèles qu'ils décrivent. Si cela permet une structuration et une identification des types de données numériques utilisés dans les équations constituant ces modèles,

c'est aussi un inconvénient limitant les modèles aux systèmes d'équation différentielle. Contrairement à SBML, CellML permet de rendre compte de la compartimentalisation des environements biologiques tels que les cellules, autorisant la modularisation des modèles. Par contre, les modèles décrits en CellML font abstraction des molécules impliquées. Ainsi, il est impossible de relier un processus modélisé à une ou plusieurs molécules précises. De son coté, SBML se révèle difficilement adapté à la représentation de modules fonctionnels au sein des modèles qu'il décrit, puisqu'il intègre toutes les données dans une même définition mathématique. En conséquence, aucun de ces deux langages n'est véritablement adapté à la représentation de la connaissance fonctionnelle telle qu'elle existe dans la littérature. Il manque donc aux biologistes un langage facile d'accès, tolérant l'ensemble des informations utilisées en biologie pour décrire les implications fonctionnelles d'une molécule, tout en étant manipulable par des outils informatiques. BioΨ [13] a pour ambition de répondre à ces critères en proposant un système de description des processus biologiques, basé sur des élements d'action élémentaire. Cette description est réalisée de manière distincte de la description physique des molécules qui supportent les processus, et organisée sur quatre niveaux de manière être directement utilisée comme source d'information pour les logiciels de représentation graphique, de simulation ou de modélisation.

Résultats

BioΨ est un langage permettant de retranscrire le plus fidèlement possible l'état des connaissances liées aux processus associés aux molécules et complexes moléculaires biologiques. Il s'inspire grandement de la théorie du système général qui définit un processus comme une fonction de l'état des composants d'un système, ainsi que du temps et de l'espace associés à ce système. Appliqué à un environnement biologique tel que la cellule, cela revient à définir un processus biologique par, d'une part l'état des molécules composants la cellule, et d'autre part le temps et l'espace nécessaire à la réalisation de ce processus. Chacune de ces données est plus ou moins présentes dans les bases de données existantes: l'état des molécules correspond aux modifications dont elles peuvent faire l'objet (modifications post-traductionnelles dans le cas des protéines); le temps correspond aux données cinétiques qui sont souvent inexistantes ou fausses car mesurées hors du contexte réel de réalisation du processus; l'espace correspond à la localisation des molécules dans la cellule. Pour représenter un processus biologique particulier il faut donc distinguer trois informations: l'ensemble des actions realisées par ce processus, les molécules impliquées dans ce processus et le contexte spatio-temporel dans lequel ce processus est réalisé [14].

Dans cette optique, BioΨ définit un premier niveau de description correspondant à des transformations pseudo-chimiques. Ce niveau décrit une centaine d'action élémentaire formalisée via le langage SMIRKS couramment utilisé en chimie [15]. SMIRKS permet de décrire des motifs de structure chimique au sein de transformation tout en identifiant les correspondances atomiques entre substrats et produits. Ces motifs peuvent être utilisés pour cibler des molécules décrites au format SMILES [16]. Tout un ensemble de logiciels existe pour manipuler ces formats de représentation, ainsi que prédire les produits de transformation en fonction d'une transformation SMIRKS et de la représentation SMILES des substrats (fig 1). Les SMIRKS définies par BioΨ sont appelées Basic Elements of Action (BEAs) et constituent le niveau le plus bas de cette description des processus biologiques. Les BEAs font l'objet d'un classification les regroupant par type d'actions et nature des liaisons chimiques ou molécules impliquées [13]. Les 3 autres niveaux mis en place par BioΨ se définissent par une combinaison de leur niveau directement inférieur.

Les Biological Activities (BAs) sont une combinaison de BEAs décrivant un processus biologique qui agit à l'échelle des domaines structuraux des molécules. La centaine de BEA définies par BioΨ permet de représenter l'ensemble des réactions enzymatiques répertoriées par l'Enzyme Commission ainsi que les processus non enzymatiques sous forme de BAs. Un processus BA_kinase

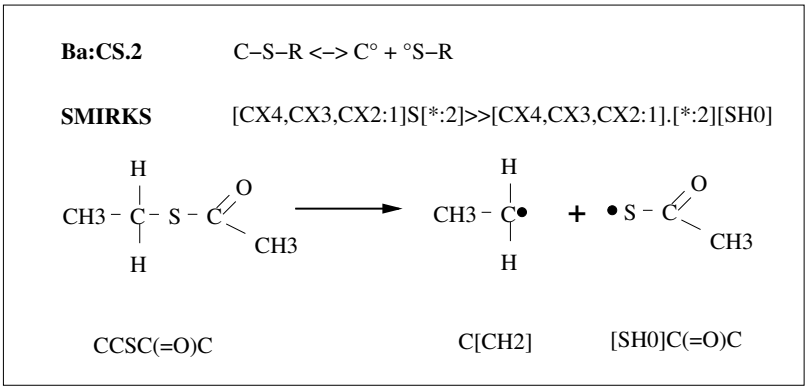


Figure 1: **Exemple de BEA**. L'élément d'action Ba:CS.2 décrit la séparation d'une liaison entre un atome de carbone et un atome de souffre. Le nom choisi pour ce BEA correspond aux différents niveaux de la classification des BEAs: B=Transfer; Ba=Chemical Group Transfer; Ba:CS=Chemical Group Transfer acting on a carbon sulfur bond; Ba:CS.2=second entry of the Chemical Group Transfer acting on a carbon sulfur bond level. La première ligne présente ce BEA dans une forme simplifiée tandis que la deuxième correspond à sa définition SMIRKS. Les deux dernières lignes donnent un exemple d'utilisation de ce BEA: la représentation SMILES de chaque molécule mise en jeu est placée sous sa représentation graphique.

décrira l'ordonnancement des différentes transformations pseudo-chimiques nécessaires à l'ajout d'un groupe phosphate sur une molécule. Cette ordonnancement est réalisé à l'aide de contraintes d'ordonnancement qui précisent les séquences temporelles de réalisation des BEAs, incluant les possibilités de réalisation parallèle. Ce deuxième niveau introduit deux autres types de contrainte qui viennent renforcer la description du contexte biologique associé à ce processus: les contraintes de spécificité permettent d'affiner le motif de structure chimique nécessaire à la réalisation d'une transformation dans le cadre précis de cette BA; les contraintes cinétiques imposent un cadre temporel au processus décrit.

Le 3ème niveau, les Biological Functionalities (BFs), sont une combinaison de BAs, décrivant la manière et les conditions de réalisation d'un processus réalisé par une molécule ou un complexe moléculaire au moyen de processus agissant à une échelle structurale inférieure. Si les BAs peuvent être associées à des domaines fonctionnels ou structuraux, lorsque ces domaines sont intégrés dans une molécule ou un complexe moléculaire, ils subissent des contraintes structurelles qui influent sur leur capacité à réaliser un processus. Ainsi les BA utilisées pour décrire une BF verront certaines de leurs caractéristiques altérées, et des relations d'interdépendance pourront être misse en place (activation ou inhibition d'un processus). Pour rendre compte de cette réalité biologique, de nouvelles contraintes sont introduites pour ce 3eme niveau de description. Les contraintes de localisation permettent de préciser une zone biologique spécifique dans laquelle la molécule réalisant le processus décrit doit se trouver pour autoriser le déroulement du processus tel qu'il est décrit. Les contraintes d'état et de conformation précisent respectivement les modifications éventuelles (phosphorylations, methylations ...) et la conformation associées à la molécule réalisant ou subissant le processus. L'expression de toutes ces contraintes est formalisée de manière à venir complémenter ou altérer les contraintes définies au niveau des BAs. Il est important de noter qu'à aucun moment les entités moléculaires ne sont définies de manière définitives: seules les caractéristiques chimiques ou biochimiques des entités impliquées dans le processus sont précisées. L'entité moléculaire supportant le processus n'est quant à elle jamais identifiée. Un exemple de description du processus associé à la SuccinateCoA ligase est présenté en figure 2.

Le dernier niveau de description de BioΨ, les Biological Roles (BRs), combine différentes BFs pour décrire un processus faisant intervenir plusieurs molécules ou complexes moléculaires dans le cadre d'un évènement cellulaire (cycle de Krebs, apoptose ...). On retrouve de la même

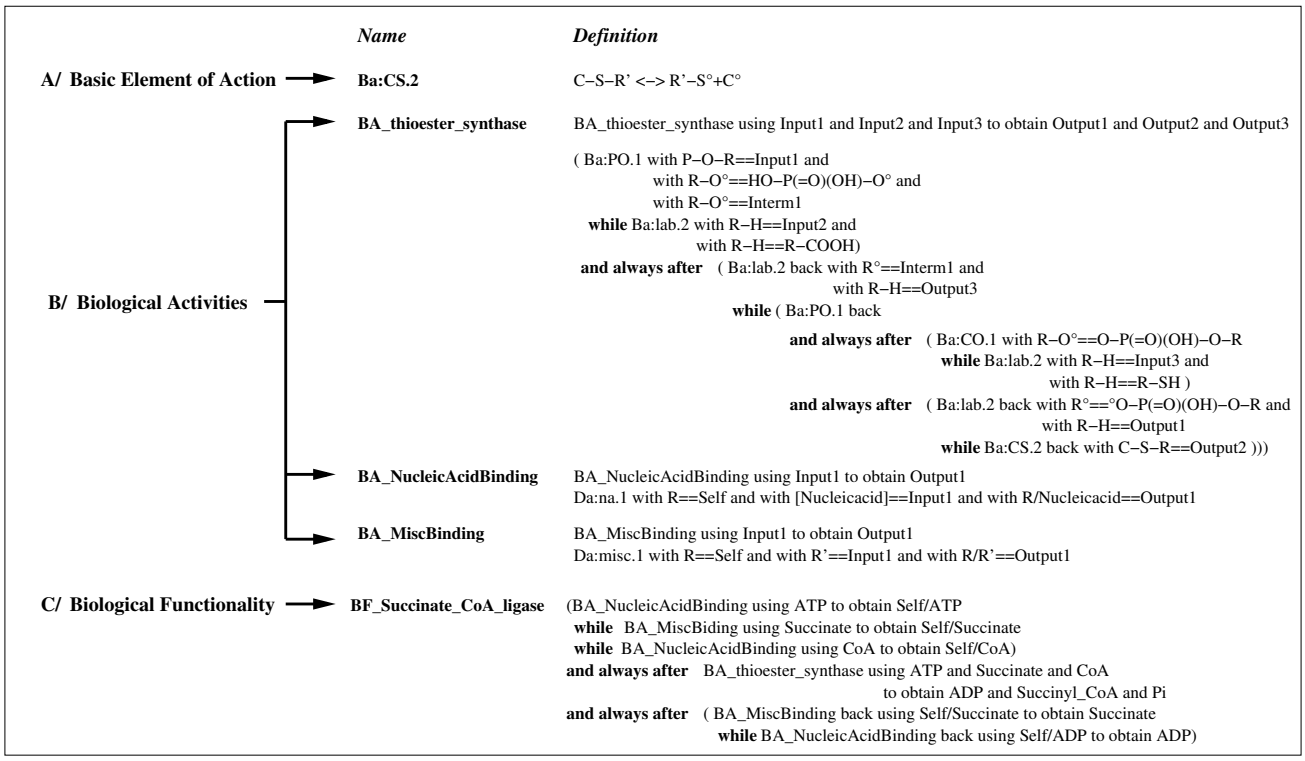


Figure 2: **Exemple de description utilisant BioΨ**. Chacun des trois premiers niveaux de description de BioΨ est illustré via différents processus.

manière les contraintes d'ordonnancement, de spécificité, de cinétique, d'état et de conformation qui viennent compléter ou modifier les informations définies aux niveaux inférieurs de description. L'organisation verticale de BioΨ autorise l'inclusion des descriptions de transformations chimiques associées aux processus décrits jusqu'à une échelle de description cellulaire. Ainsi les effets de processus de niveau inférieur participent à la description des processus de niveau supérieur, conservant la chaine de causalité des évènements menant à la réalisation d'un processus de niveau cellulaire.

Discussion

De part ses caractéristiques, BioΨ se révèle être un langage aux potentialités beaucoup plus larges que les solutions évoquées précédemment. Il permet de représenter l'ensemble des types d'informations relatives à la description des processus biologiques que l'on peut trouver dans la littérature. Hors cette littérature ne rend compte que des processus biologiques connus dans les limites de notre capacité à les étudier. Cette capacité, ainsi que l'éventualité de découvrir de nouveaux types de processus, est soumise à une évolution constante. La gamme de combinaisons offerte par la centaine de BEA définie par BioΨ permet de considerer des processus biologiques qui dépassent l'ensemble actuellement décrit. BioΨ est ainsi prêt à représenter des processus biologiques aujourd'hui inconnus sans pour autant devoir introduire de nouvelles actions élémentaires dans le corpus existant. Son organisation sous forme de niveaux imbriqués autorise une extension de son champ de description au delà du domaine cellulaire. Il est tout a fait possible d'envisager un 5ème niveau combinant les BRs pour décrire le processus associé à un type cellulaire particulier. De nouvelles contraintes spécifiques à ce niveau devraient probablement être introduites, mais elles ne perturberaient pas l'organisation originale du langage. Le système de contraintes

permet d'imposer une sémantique biologique aux descriptions utilisant BioΨ. Ainsi, si deux BFs sont combinées dans un BR de manière séquentielle alors que l'une nécessite une localisation mitochondriale et l'autre une localisation nucléaire, l'absence de continuité spatiale de cette description pourra être mise en évidence, traduisant une déficience dans la description du processus. A l'origine implémenté dans un formalisme propre, BioΨ s'est vu transcrit en une définition de type de document (DTD) et un schema XML, permettant ainsi sa manipulation plus aisée par les logiciels actuels. Ainsi, dans l'éventualité oÃ[1] les descriptions utilisant BioΨ se voyaient intégrées directement dans les bases de données, l'information fonctionnelle formalisée serait directement accessible aux logiciels de simulation et de modélisation qui pourront en dériver des modèles plus adaptées à leur besoin. Le principal intérêt de BioΨ est qu'il permet d'intégrer toute la connaissance existante relative à un processus biologique, laissant par la suite le choix à l'utilisateur de simplifier cette description sous la forme d'un modèle.

BioΨ a été utilisé dans le cadre du projet MitoScop pour représenter la connaissance associée au cycle de Krebs [17]. Cette première mise en application de BioΨ utilisant de véritables données biologiques a permis de confirmer ses avantages dans le domaine de la description de processus, et plus particulièrement en tant qu'aide à la comparaison automatisée de processus biologiques. Les propriétés de formalisation des processus biologiques de BioΨ permettent de réaliser des comparaisons fonctionnelles de protéines indépendamment des informations de séquence. Il autorise la mise à disposition d'informations auparavant enfouies dans la littérature, ou tout simplement proposées de manière inutilisable informatiquement dans les bases de données. Le couplage de BioΨ à une base de données dédiée aux processus biologiques ainsi qu'à des logiciels de représentation graphique et de simulation constituent les prochaines étapes de la mise en application de ce langage afin de lui donner une plus large visibilité.

DTD,schema XML et un exemple de description utilisant BioΨ sont disponibles à l'adresse suivante:
http://cpbs.univ-montp1.fr/supmaterials/bioinformatics/maziere_et_al_RIAMS_IPG05

References

[1] Nomenclature Committee of the International Union of Biochemistry and Molecular Biology. *Enzyme Nomenclature*. Academic Press, San Diego, California, 1992. Historical introduction.

[2] A. Fleischmann, M. Darsow, K. Degtyarenko, W. Fleischmann, S. Boyce, K.B. Axelsen, A. Bairoch, D. Schomburg, K.F. Tipton, and R. Apweiler. IntEnz, the integrated relational enzyme database. *Nucleic Acids Res*, 32(1):D434–7, 2004.

[3] W. Busch and M.H. Saier, Jr. The transporter classification (TC) system, 2002. *Crit Rev Biochem Mol Biol*, 37(5):287–337, 2002.

[4] W. Busch and M.H. Saier, Jr. The IUBMB-endorsed transporter classification system. *Methods Mol Biol*, 227:21–36, 2003.

[5] H.W. Mewes, K. Albermann, K. Heumann, S. Liebl, and F. Pfeiffer. MIPS: a database for protein sequences, homology data and yeast genome information. *Nucleic Acids Res*, 25(1): 28–30, 1997.

[6] P. D. Karp, M. Riley, M. Saier, I. T. Paulsen, S. M. Paley, and A. Pellegrini-Toole. The ecocyc and metacyc databases. *Nucleic Acids Res*, 28(1):56–9, Jan 1 2000.

[7] M. Ashburner. Creating the gene ontology resource: design and implementation. *Genome Res*, 11(8):1425–33, 2001.

[8] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J.C. Doyle, H. Kitano, 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 Novere, 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):524–31, 2003.

[9] A. Finney and M. Hucka. Systems biology markup language: Level 2 and beyond. *Biochem Soc Trans*, 31(Pt 6):1472–3, 2003.

[10] Warren J. Hedley, Melanie R. Nelson, David P. Bullivant, and Poul F. Nielsen. A short introduction to CellML. *Phil. Trans. R. Soc. Lond.*, 359:1073–89, 2001.

[11] T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler. Extensible markup language (XML) 1.0 (2nd edn). *W3C recommendation (http://www.w3c.org)*, 6 October 2000.

[12] R. Ausbrooks, S. Buswell, S. Dalmas, S. Devitt, A. Diaz, R. Hunter, B. Smith, N. Soifer, R. Sutor, and S. Watt. Mathematical markup language (MathML) version 2.0. *W3C proposed recommendation (http://www.w3c.org)*, 8 January 2001.

[13] P. Maziere, C. Granier, and F. Molina. A description scheme of biological processes based on elementary bricks of action. *J Mol Biol*, 339(1):77–88, 2004.

[14] J. Van Helden, A. Naim, C. Lemer, R. Mancuso, M. Eldridge, and S. J. Wodak. From molecular activities and processes to biological function. *Brief Bioinform*, 2(1):81–93, March 2001.

[15] Daylight. Daylight Theory Manual. http://www.daylight.com, 2003.

[16] D. Weininger. Smiles 1. introduction and encoding rules. *J. Chem. Inf. Comput. Sci*, 28(31), 1988.

[17] P. Mazière, N. Parisey, M. Beurton-Aimar, and F. Molina. Formal TCA cycle description based on elementary actions. *submitted*, 2005.

LABORATOIRE DE MÉTHODES INFORMATIQUES

Journée thématique RIAMS « Réseaux d'interaction : analyse, modélisation et simulation »

Jean-Paul COMET et Sandrine VIAL

Résumé

L'objectif de cette journée thématique francophone sur l'analyse, la modélisation et la simulation des réseaux d'interaction dans le cadre de la biologie est de réunir toute la comunauté scientifique souhaitant partager ses compétences propres pour la compréhension des réseaux d'interaction biologiques. Cette première rencontre est principalement, mais non exclusivement axée sur les thèmes : analyse des systèmes d'interaction biologiques, les grands réseaux d'interaction, l'évolution des réseaux, la modélisation de systèmes biologiques, la simulation de tels systèmes.
Cette journée a été organisée grâce au soutien de l'ACI VICANNE.

LaMI, UMR 8042 CNRS - Université d'Évry Val d'Essonne, Genopole
Tour Évry 2, 523 place des terrasses de l'Agora
91000 Évry Cedex, France