Modélisation thermodynamique des réseaux de régulation de gènes

Xavier ERNY, Nick BARTON, BARTON's group/IST Austria

31 juillet 2016

Le contexte général

De quoi s'agit-il ? D'où vient la question ? Quels sont les travaux déjà accomplis dans ce domaine dans le monde ?

Un gène peut être soit actif soit inactif, quand il est actif il produit des protéines qui peuvent soient aider d'autres gènes à s'activer soit les en empêcher. C'est ainsi que certains gènes se régulent entre eux, en formant une sorte de réseau. Ce stage consistait à modéliser cette régulation, en utilisant un modèle thermodynamique. Les réseaux de régulation de gènes sont étudiés par beaucoup de biologistes, afin de comprendre leur fonctionnement et leur rôle dans l'organisme des individus. De plus, le modéle thermodynamique a déjà été étudié dans plusieurs papiers, et a permis de comprendre le fonctionnement de certains gènes.

Le problème étudié

Quelle est la question que vous avez résolue ? Pourquoi est-elle importante, à quoi cela sert-il d'y répondre ? Pourquoi êtes-vous le premier chercheur de l'univers à l'avoir posée ?

La question générale est : Comment étudier efficacement les propriétés dynamiques de ces réseaux ? Étudier ces réseaux de gènes est important car les gènes sont à l'origine de presque tous les processus des cellules, et, de plus, ces réseaux sont un point clef pour comprendre l'évolution, car les mutations d'un individu peuvent se voir au niveau de ces réseaux. Le modèle thermodynamiqe a déjà été utilisé pour analyser les gènes, mais seulement sur des gènes isolés et non sur des réseaux de gènes. De plus, ce modèle n'a été utilisé que pour analyser l'activité stationnaire des gènes, et non leur activité dynamique.

La contribution proposée

Qu'avez vous proposé comme solution à cette question ? Attention, pas de technique, seulement les grandes idées ! Soignez particulièrement la description de la démarche *scientifique*.

Pour étudier un réseau, on a écrit un algorithme qui, étant donné un réseau et une propriété de ce réseau (propriété liée à la dynamique du réseau), renvoie des conditions sur les paramètres du réseau sous lesquelles la propriété est satisfaite. Plus exactement l'algorithme renvoit une formule logique dont les propositions atomiques sont des conditions, de telle sorte que chaque modèle de cette formule représente une conjonction de conditions sous lesquelles la propriété est satisfaite.

Les arguments en faveur de sa validité

Qu'est-ce qui montre que cette solution est une bonne solution ? Des expériences, des corollaires ? Commentez la *stabilité* de votre proposition : comment la validité de la solution dépend-elle des hypothèses de travail ?

Dans un premier temps on a simulé la dynamique de certains réseaux, qui avaient été étudiés expérimentalement, pour vérifier que le modèle utilisé donnait des résultats cohérents avec la réalité, ce qui fut le cas. Ensuite on a comparé les conditions données par notre algorithme avec les conditions des expériences liées à ces réseaux, on a constaté des résultats très satisfaisants. Comme notre approche du problème dépendait d'une hypothèse, on a pensé à une généralisation de cette approche pour supprimer cette hypothèse. L'idée semblait donner quelques résultats intéressants mais nous n'avons pas eu suffisamment de temps pour la formaliser entièrement.

Le bilan et les perspectives

Et après ? En quoi votre approche est-elle générale ? Qu'est-ce que votre contribution a apporté au domaine ? Que faudrait-il faire maintenant ? Quelle est la bonne *prochaine* question ?

Nous avons introduit une syntaxe général permettant de représenter beaucoup de réseaux de gènes. En résumé, on a développé un "outil" permettant d'étudier les propriétés de ces réseaux. Il y a plusieurs choses que l'on pourrait faire :

— étendre la syntaxe pour modéliser plus de réseaux de régulation, en gardant une syntaxe "propre"
— changer l'approche du problème pour supprimer les hypothèses simplificatrices de la modélisation
— modéliser les conséquences de l'évolution sur les réseaux (ie les mutations)
— trouver une application concrète en biologie

Ecole Normale Supérieure de Cachan

Département Informatique

Rapport de stage M1

Modélisation thermodynamique des réseaux de régulation de gènes

Auteur : Xavier Erny
Encadrant : Nick Barton
Encadrant : Tiago Paixão
Equipe : Barton's group
Laboratoire : IST Austria

1er Février 2016 – 31 Mai 2016

Table des Matières

1 Introduction

1.1 Encadrement

J'ai fait ce stage au laboratoire IST Austria (Institute of Science and Technology) au sein du "groupe Barton" sous la supervision de Nick Barton, professeur à l'IST en biologie, et Tiago Paixão, chercheur postdoctoral à l'IST en biologie.

Le champs de recherche du groupe Barton est la biologie évolutive, c'est-à-dire l'étude de l'évolution, ses effets et son fonctionnement. Pour cela, il faut recourir à des observations et des expériences, mais aussi à des modèles mathématiques pour analyser les données et ainsi mieux les comprendre. Au cours du stage, j'ai travaillé sur une de ces modélisations dans un projet avec Tiago Paixão, Mirco Giacobbe, doctorant en informatique à l'IST, et Tatjana Petrov, chercheur postdoctoral à l'IST en informatique.

Pour ce projet, nous sommes partis d'un ancien projet [Giacobbe *et al.*, 2015], en affinant la modélisation pour qu'elle corresponde plus à la réalité, et qu'elle puisse ainsi donner des informations utilisables en pratique. J'ai principalement travaillé avec Mirco sur le projet pour ce qui concerne les questions comme : Ce qu'on peut faire ? Comment le faire ? Le rôle principal de Tiago était déterminer de si nos hypothèses de travail et ce que l'on voulait faire avait bien un sens en biologie, ce qui est fondamental en modélisation. Tatjana a été très occupée à cette période mais a quand même pris le temps de suivre le projet, ainsi qu'à penser à certaines conférences que ce projet pouvait intéresser, comme CMSB ("Computational Method in Systems Biology").

1.2 Sujet de recherche

Le but du stage était de modéliser les réseaux de régulation de gènes. Pour comprendre comment fonctionnent ces réseaux, il faut savoir qu'un gène peut être soit actif soit inactif. Quand il est actif il produit des protéines qui peuvent soit aider d'autres gènes à s'activer soit les en empêcher en se liant à l'ADN. C'est ainsi que les gènes se régulent entre eux. La quantité de protéines qu'un gène produit s'appelle l'expression du gène.

L'expression des gènes est à l'origine de presque tous les processus des cellules, et il existe des arguments montrant que la régulation de cette expression est un point clef pour comprendre la diversification des espèces, c'est-à-dire l'évolution, comme le suggère [Monteiro, 2011]. En effet, les effets d'une mutation influencent directement les réseaux de régulation. Comprendre et prédire le comportement de ces réseaux de régulation est un problème fondamental de la biologie moléculaire et de la biologie synthétique. Il est donc intéressant d'avoir un modèle mathématique efficace pour vérifier des hypothèses et prédire des résultats dans ce domaine.

Le modèle sur lequel on a décidé de travailler est le modèle thermodynamique introduit par Shea et Ackers [Shea et Ackers, 1985], et développé par d'autres [Bintu *et al.*, 2005]. L'idée de base du modèle est que l'expression d'un gène peut être représentée par la probabilité que ce gène soit actif. Cette probabilité peut être calculée par des méthodes de physique statistique. Il est important de noter que ces probabilités ne dépendent que de quantités mesurables expérimentalement, comme des concentrations de protéines et des énergies de liaison. Ce type de modèle a déjà montré qu'il pouvait prédire précisément l'expression de gènes, et a permis de mettre en évidence l'architecture de certains gènes [Hermsen *et al.*, 2006]. Toutefois, on n'a pas encore utilisé de modèle thermodynamique sur des réseaux, mais seulement sur des gènes seuls. De plus, l'activité dynamique de ce type de modèle sur des gènes n'a pas encore été explorée, contrairement à l'activité stationnaire des gènes.

Nous avons introduit une syntaxe et une sémantique des réseaux de régulation de gènes qui permettent d'utiliser le modèle thermodynamique sur beaucoup de réseaux. Nous avons vérifié la cohérence de notre sémantique en simulant la dynamique de certains réseaux connus. De plus, nous avons élaboré une méthode permettant de vérifier et de prédire les propriétés des réseaux de régulation. Nous justifions la fiabilité de cette méthode en comparant les résultats que nous obtenons avec des résultats obtenus expérimentalement sur des réseaux de régulation de gènes connus.

2 Modèle biologique

2.1 Structure

Pour bien comprendre comment fonctionne la régulation des gènes, il faut d'abord savoir ce qu'est un gène et ce qu'est l'ADN.

L'ADN est une molécule formée de deux séquences complémentaires de bases nucléiques. Il existe quatre bases nucléiques : adénine, thymine, cytosine et guanine. L'adénine et la thymine sont complémentaires, ainsi que la cytosine et la guanine, comme on peut le voir sur la figure 1.

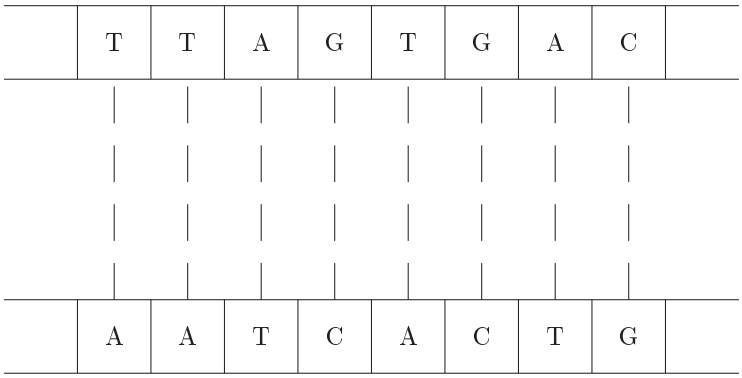


FIGURE 1 – Exemple de fragment d'ADN
A désigne l'adénine, T la thymine, C la cytosine et G la guanine. Chaque base est reliée à sa base complémentaire dans l'autre séquence de l'ADN par une liaison hydrogène (représentée en ligne pointillée).

Les gènes sont des fragments d'ADN (ce sont donc des séquences de bases nucléiques) qui ont un site promoteur situé sur un des brins de l'ADN (ce site est donc aussi une séquence de bases nucléiques). Un gène peut produire des protéines qui vont ensuite influencer la production de protéines d'autres gènes, c'est ainsi que fonctionne la regulation des gènes.

2.2 Production de protéines

La production de protéines par un gène se fait en trois étapes :
— il faut qu'un ARN polymérase (qui est une protéine) se lie sur le site promoteur du gène.
— **transcription** : ce gène pourra alors créer l'ARN messager associé à l'ARN polymérase. L'ARN messager contient l'information nécessaire à la synthèse d'une protéine.

— **traduction** : les ribosomes pourront finalement créer une protéine à partir de l'ARN messager.

2.3 Régulation

Les protéines sont capables de s'accrocher à l'ADN, comme le fait l'ARN polymérase. Chaque protéine aura une certaine affinité avec chaque séquence de bases nucléiques. Il y a donc pour chaque protéine, des sites sur l'ADN où la protéine peut se fixer. Ce sont des sites opérateurs. Si une protéine a un site opérateur proche du site promoteur d'un gène, alors cette protéine peut influencer l'expression de ce gène de deux manières : une protéine peut être un répresseur ou un activateur d'un gène.

La répression est un processus simple : il suffit d'imaginer une configuration dans laquelle un site opérateur et un site promoteur se chevauchent, de telle sorte qu'il soit impossible que les protéines associées à ces sites soient liées en même temps. Autrement dit, la protéine associée à ce site opérateur peut empêcher l'ARN polymérase de se lier au site promoteur du gène, et ainsi empêcher le gène de produire des protéines. Ceci est illustré sur la figure 2 où deux sites ont une base nucléique en commun.

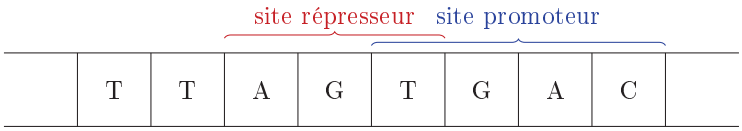


FIGURE 2 – Exemple de fragment d'ADN avec une répression

Pour comprendre, l'activation, il faut savoir que les gènes peuvent se lier entre eux, en plus de se lier à l'ADN. Ainsi, si un site operateur est très proche d'un site promoteur, alors quand les protéines vont se lier sur les deux sites, elles vont aussi se lier entre elles. Par conséquent, la protéine associée au site opérateur peut aider le gène à produire des protéines, puisque si elle se lie au site opérateur, alors l'ARN polymérase aura une meilleure affinité avec le site promoteur car, en plus des liaisons avec l'ADN, il aura des liaisons avec la protéine. Le site opérateur doit donc être suffisamment proche du site promoteur pour que les protéines puissent se lier entre elles, mais ces sites doivent être suffisamment éloignés pour que les protéines ne se gènent pas mutuellement comme lors d'une répression.

On vient de décrire l'activation et la répression du site promoteur, mais il est aussi possible de d'activer et de réprimer des sites opérateurs de la même façon, ce qui peut conduire à des régulations complexes.

Les mutations d'un organisme modifient certaines bases nucléiques, et donc peuvent modifier les réseaux de régulation en modifiant les affinités des protéines sur les sites opérateur. Autrement dit, les mutations peuvent enlever ou créer des régulations.

3 Modèle théorique

3.1 Syntaxe

On modélise ces régulations sous la forme d'un graphe orienté un peu spécial. Les nœuds représentent tous des gènes, sauf un, noté P, qui représente l'ARN polymérase. Les arcs représentent les régulations du réseau, il y en a deux types : l'activation, notée $\rightarrow$, et la répression, notée $\dashv$. Ici les arcs sont des relations entre les nœuds et d'autres arcs, car ce que font les régulations est en fait d'aider ou d'empêcher d'autres régulations. Chaque arc correspond à un site opérateur ou un site promoteur de l'ADN. Appellons ces graphes particuliers, des graphes de régulation et définissons les rigoureusement.

Définition 1 (Graphe de régulation)
Un graphe de régulation est un couple $(G \sqcup \{P\}, R)$ *tel que :*
— G *est l'ensemble des gènes du réseau et* P *désigne l'ARN polymérase*
— $R = (R^+, R^-)$ *est un ensemble de régulations associé à* (G, P) *où* R^+ *est l'ensemble des relations d'activation et* R^- *est l'ensemble des relations de repression*

Pour définir les régulations proprement, on les définit par niveau. Les régulations de niveau 0 sont exactement l'activation des gènes par l'ARN polymérase, les régulations de niveau 1 sont celles qui régulent celles de niveau 0, et ainsi de suite. Les ensembles de régulations sont définis rigoureusement de la manière suivante :

Définition 2 (Ensemble de régulations)
Un ensemble de régulation $R = (R^+, R^-)$ *associé à* (G, P) *peut s'écrire sous la forme*

$$R^+ = \bigcup_{n=0}^{+\infty} R_n^+ \quad et \quad R^- = \bigcup_{n=0}^{+\infty} R_n^-$$

en vérifiant les conditions suivantes :
— $R_0^+ = \bigcup_{g \in G} \{(P, g)\}$
— $R_0^- = \varnothing$
— $\forall n \in \mathbb{N}, R_{n+1}^+ \subseteq G \times (R_n^+ \cup R_n^-)$
— $\forall n \in \mathbb{N}, R_{n+1}^- \subseteq G \times (R_n^+ \cup R_n^-)$
— $\bigcup_{n=0}^{+\infty} R_n^+$ *et* $\bigcup_{n=0}^{+\infty} R_n^-$ *sont finis*

La figure 3 montre un exemple de graphe de régulation. Dans ce graphe, il y a deux gènes A et B. La seule régulation que subit A est celle de l'ARN polymérase, autrement dit, la zone de régulation de ce gène se résume à un site promoteur qui n'est ni réprimé ni aidé. La régulation de B est plus compliquée, car son site promoteur est réprimé par un site où les protéines de A peuvent se lier, et ce site est lui-même activé par un autre site où les protéines de B peuvent se lier, comme illustré à la figure 4.

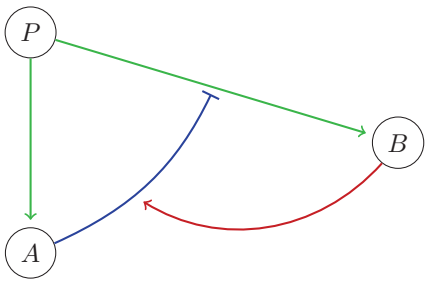


FIGURE 3 – Exemple de graphe de régulation
Les arcs verts représentent les régulations de niveau 0, le bleu la régulation de niveau 1 et le rouge celle de niveau 2.

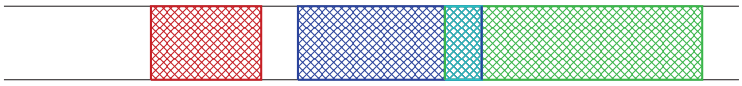


FIGURE 4 – Zone de régulation de B dans le graphe de la figure 3
Les couleurs des zones sont les mêmes que celles des régulations du graphe de la figure 3. La zone verte est le site promoteur, la bleue est le site qui réprime le promoteur et la rouge est celle qui active la répression. Ici on suppose que la zone rouge est suffisamment éloignée de la bleue pour ne pas la réprimer (ie deux protéines peuvent se lier en même temps aux deux sites), mais suffisamment proche pour l'aider (ie quand deux protéines se lient aux deux sites, elles se lient aussi entre elles).

3.2 Sémantique thermodynamique

Le modèle thermodynamiqe est stochastique. Le principe est le suivant : étant donné un gène g, on examine toutes les configurations possibles de sa zone de régulation en leur donnant un poids, et on distingue les cas où g est actif (ie où un ARN polymérase est lié au site promoteur de g) des cas où il est inactif (cf figure 5). En notant $Z_{on}(g)$ la somme des poids des configurations où g est actif et $Z_{off}(g)$ la somme des poids des autres configurations, on peut calculer la probabilité $\mathbb{P}_{on}(g)$ d'activation de g par la formule :

$$\mathbb{P}_{on}(g) = \frac{Z_{on}(g)}{Z_{on}(g) + Z_{off}(g)}$$

On peut remarquer que $\mathbb{P}_{on}(g)$ doit en fait dépendre du nombre de protéines présentes dans le système (ie les protéines produites par chacun des gènes qui n'ont pas encore été dégradées), puisque, plus il y a de protéines d'un certain type, plus la probabilité qu'une protéine de ce type se lie sur un site est grande. On notera donc plutôt $\mathbb{P}^{\sigma}_{on}(g)$ au lieu de $\mathbb{P}_{on}(g)$, où σ est un élément de $\mathbb{N}^G$ (ie σ indique pour chaque gène, le nombre de protéines que ce gène a créées). Et on notera de la même façon $Z^{\sigma}_{on}(g)$ au lieu de $Z_{on}(g)$ et $Z^{\sigma}_{off}(g)$ au lieu de $Z_{off}(g)$.

On peut donc représenter la dynamique d'un réseau de régulation de gènes avec une chaîne de Markov à temps continue (cf annexe) avec un nombre infini d'états, en définissant les états comme étant les éléments de $\mathbb{N}^G$. Pour définir les taux des transitions de la chaîne de Markov, il faut se donner pour chaque gène g, un taux de production de protéine λ_g (ie le taux avec lequel un gène

produit une protéine quand un ARN polymérase s'est lié à son site promoteur), et un taux de dégradation de protéine γ_g. On définit deux types de transitions :

— Celles qui vont d'un état $\sigma = (\sigma_g)_{g\in G}$ vers $(\sigma_g + \delta_{gg'})_{g\in G}$ pour un certain $g' \in G$ (où $\delta_{gg'}$ vaut 1 si $g = g'$ et 0 sinon). Le taux de cette transition est $\lambda_{g'} \times \mathbb{P}^{\sigma}_{on}(g')$. En effet, cette transition correspond à la réaction de production d'une protéine, il faut donc prendre en compte le taux de production et la probabilité qu'un ARN polymérase se lie.

— Celles qui vont d'un état $\sigma = (\sigma_g)_{g\in G}$ vers $(\sigma_g - \delta_{gg'})_{g\in G}$ pour un certain $g' \in G$ tel que $\sigma_{g'} > 0$. Le taux de cette transition est $\sigma_{g'} \times \gamma_{g'}$.

Pour énumérer toutes les configurations d'une zone de régulation, il suffit de choisir pour chacune de ces régulations si elle est effective (ie si la protéine associée à la régulation s'est liée ou non au site de la régulation). La seule restriction est que si on considère une répression $A \dashv r$, alors il est impossible que les relations $A \dashv r$ et r soient effectives en même temps. Cette méthode d'énumération est illustrée à la figure 5.

Notons $c \in \{0,1\}^{R^+ \cup R^-}$ une configuration (ie une fonction qui décide pour chaque relation si elle est effective ou non), et expliquons comment calculer le poids $p(c)$ de cette configuration. On peut séparer ce poids en deux parties :

$$p(c) = \prod_{\substack{(A,r)\in R^+\sqcup R^- \\ c(A,r)=1}} [A] e^{\varepsilon_{Ar}/k_BT} \times \prod_{\substack{(A,(B,r))\in R^+ \\ c(A,(B,r))=1 \\ c(B,r)=1}} e^{\varepsilon_{AB}/k_BT}$$

où ε_{Ar} est l'énergie de liaison protéine-ADN entre les protéines produites par A et le site opérateur de la régulation (A,r), ε_{AB} est l'énergie de liaison protéine-protéine entre les protéines produites par A et celles produites par B, k_B est la constante de Boltzmann et T est la température, que l'on supposera constante et fixée dans toute la suite. L'intérêt de diviser les énergies par k_BT dans la formule, est de rendre les arguments des exponentielles adimensionnés.

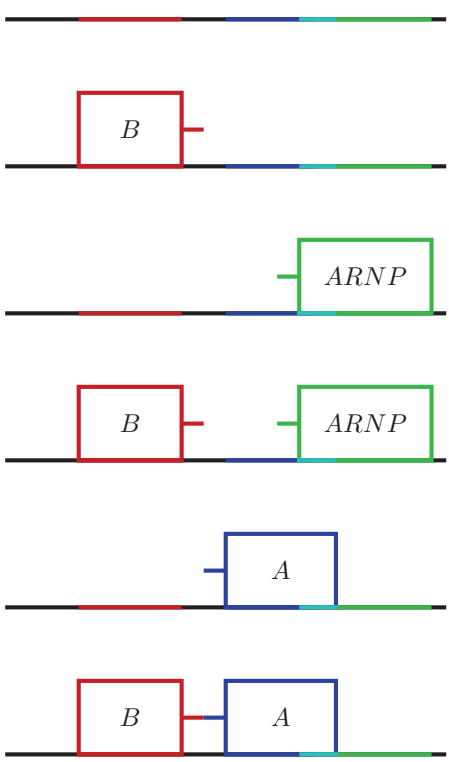
Dans la formule précédente, on considère que le poids de l'événement "une protéine particulière de A se fixe sur le site opérateur (A,r)" est une exponentielle de l'énergie de liaison entre cette protéine et ce site de l'ADN (plus cette énergie est élevée, plus la protéine aura tendance à se lier à se site), il s'agit d'un poids de Boltzmann. Donc le poids de l'événement "une protéine quelconque de A se fixe sur le site opérateur (A,r)" (ie la régulation (A,r) est active) doit être ce poids multiplié par la concentration de protéines produites par A. On notera $[A]$ la concentration de protéines produites par A encore présentes dans le système. En réalité $[A]$ est un abus de notation, car il faut que cette quantité soit adimensionnée pour que le calcul ait un sens, dans nos applications numériques $[A]$ ne désigne pas N_A/V, mais plutôt $N_A/(N_S \times \exp(\overline{\varepsilon_A}/k_BT))$ (où N_A est le nombre de protéines produites par A, V le volume du système, N_S est le nombre de sites de l'ADN où les protéines de A peuvent se lier et $\overline{\varepsilon_A}$ est la moyenne des énergies de liaison entre ces protéines et ces sites). On peut trouver plus de détails sur ce modèle dans [Bintu *et al.*, 2005].

Pour le second produit de la formule de $p(\sigma)$, on prend en compte l'énergie apportée par les liaisons protéines-protéines des activations, qui doivent augmenter l'énergie totale de la configuration, et donc aussi son poids. Une énergie de laison protéine-protéine sera toujours positive.

Si on reprend l'exemple de la figure 3, dans lequel on suppose que l'état du système σ définit les concentrations de protéines produites $[A]$ et $[B]$, on a

$$Z^{\sigma}_{on}(B) = [ARNP] e^{\varepsilon_{ARNP}/k_BT} + [ARNP] e^{\varepsilon_{ARNP}/k_BT} [B] e^{\varepsilon_B/k_BT}$$

ie la somme des énergies des deux configurations où un ARN polymérase est lié.



$p = 1$

$p = [B]e^{\varepsilon_B/k_BT}$

$p = [ARNP]e^{\varepsilon_{ARNP}/k_BT}$

$p = [ARNP]e^{\varepsilon_{ARNP}/k_BT}[B]e^{\varepsilon_B/k_BT}$

$p = [A]e^{\varepsilon_A/k_BT}$

$p = [A]e^{\varepsilon_A/k_BT}[B]e^{\varepsilon_B/k_BT}e^{\varepsilon_{A-B}/k_BT}$

FIGURE 5 – Les énergies associées à chaque configuration de la zone de régulation du gène B dans le graphe 3

On suppose que le système est dans l'état σ qui définit les concentrations de prodution de protéines $[A]$ et $[B]$.

$$Z^{\sigma}_{off}(B) = 1 + [B]e^{\varepsilon_B/k_BT} + [A]e^{\varepsilon_A/k_BT} + [A]e^{\varepsilon_A/k_BT}[B]e^{\varepsilon_B/k_BT}e^{\varepsilon_{A-B}/k_BT}$$

ie la somme des énergies des quatre configurations où il n'y a pas d'ARN polymérase lié.

Et on a donc

$$\mathbb{P}^{\sigma}_{on}(B) = \frac{[ARNP]e^{\varepsilon_{ARNP}/k_BT} + [ARNP][B]e^{(\varepsilon_{ARNP}+\varepsilon_B)/k_BT}}{1 + [B]e^{\varepsilon_B/k_BT} + [A]e^{\varepsilon_A/k_BT} + [A][B]e^{(\varepsilon_A+\varepsilon_B+\varepsilon_{A-B})/k_BT} + [ARNP]e^{\varepsilon_{ARNP}/k_BT} + [ARNP][B]e^{(\varepsilon_{ARNP}+\varepsilon_B)/k_BT}}$$

La taille des expressions $\mathbb{P}^{\sigma}_{on}(g)$ croît exponentiellement avec le nombre de régulations du graphe, puisque chaque régulation correspond à un site opérateur de la zone de régulation, et que le calcul des $Z^{\sigma}_{on}(g)$ et $Z^{\sigma}_{off}(g)$ correspond à une énumération des configurations possibles de la zone. Il est donc intéressant de développer un outil pour analyser automatiquement ces expressions.

3.3 Simulation

Tout d'abord, pour vérifier si le modèle thermodynamique décrit bien la réalité, on a simulé sa dynamique. Pour cela on a utilisé l'algorithme de Gillespie. Cet algorithme prend en entrée l'état initial d'un système, un ensemble de réactions chimiques qui peuvent se produire dans ce système, ainsi qu'un taux de réaction pour chaque réaction, et permet de simuler la dynamique du système. L'algorithme est explicité ci-après.

Algorithme 1 : l'algorihme de Gillespie

Données : un système S dans un état σ
Résultat : la mise à jour du système
Entrées : $(R_i)_{i\in I}$ (ensemble de réactions), $(\lambda^{\sigma}_i)_{i\in I}$ (taux de réactions dépendant de l'état du système), T (durée maximale de la simulation)
$t \leftarrow 0$;
tant que $t < T$ **faire**
  $\lambda^{\sigma} \leftarrow \sum_{i\in I} \lambda^{\sigma}_i$;
  $t \leftarrow t + \mathcal{E}(\lambda^{\sigma})$;
  choisir une réaction R_i avec probabilité $\lambda^{\sigma}_i/\lambda^{\sigma}$;
  mettre à jour σ par rapport à R_i;
  mettre à jour λ^{σ}_i pour tous les $i \in I$
fin

Remarque : $\mathcal{E}(\lambda)$ désigne la loi exponentielle de paramètre λ. $t \leftarrow t + \mathcal{E}(\lambda)$ est un abus de notation, cela signifie qu'à chaque itération de la boucle on définit une nouvelle variable aléatoire t' qui suit $\mathcal{E}(\lambda)$, et on remplace la valeur de t par $t + t'$.

Dans notre cas, les états du sytèmes sont $\mathbb{N}^G$, et il y a deux réactions chimiques par gène g, la réaction de production d'une protéine de taux $\lambda_g \times \mathbb{P}^{\sigma}_{on}(g)$ et la réaction de dégradation d'une protéine de taux $\sigma_g \times \gamma_g$.

On a simulé la dynamique de deux réseaux de régulation et de deux de leurs variantes (figures 6 et 7), qui ont été étudiés expérimentalement dans [Gardner *et al.*, 2000] et [Elowitz et Leibler, 2000].

Pour chaque simulation on trace un graphe montrant l'évolution de l'expression de chaque gène, sur les figures 8 et 9. Pour les ordres de grandeur des paramètres, on utilise les valeurs obtenues

FIGURE 6 – Le "toggle switch" de [Gardner *et al.*, 2000].
Le réseau du papier [Gardner *et al.*, 2000] est celui de droite.

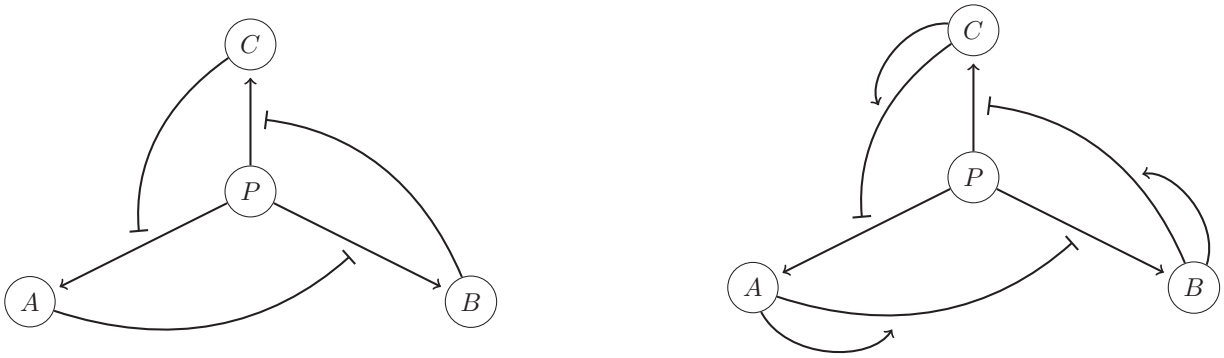


FIGURE 7 – Le "repressilator" de [Elowitz et Leibler, 2000].
Le réseau du papier [Elowitz et Leibler, 2000] est celui de gauche.

expérimentalement dans [Bintu *et al.*, 2005]. On suppose que l'ordre de grandeur du paramètre $e^{-\varepsilon/k_BT}$ vaut à peu près 50 si ε désigne l'énergie de laision protéines-ARN polymérase, 100 pour une liaison protéine-protéine, 0.1 pour un liaison protéine-ADN et 0.67 pour une liaison ARN polymérase-ADN. On suppose aussi que la concentration d'ARN polymérase est constante et vaut à peu près 5. Plusieurs paramètres ont été testés, car ces valeurs dépendent des gènes considérés. De plus considérer plusieurs paramètres permet de vérifer la robustesse du modèle.

La première chose que l'on peut noter est qu'ajouter une auto-activation des répressions revient exactement à augmenter la force des répressions, ce qui est naturel. En effet, sur les figures 8 et 9, les courbes des graphiques $A2$ et $B1$ ont le même comportement (pour la figure 8, les deux gènes ont au début la même expression, puis l'un deux s'exprime au détriment de l'autre), ainsi que les courbes $B2$ et $C1$ (pour la figure 8, un gène s'exprime fortement très rapidement, puis cette expression oscille).

On peut aussi remarquer que dans les cas où les répressions sont faibles, les réseaux ont le même comportement, tous les gènes s'expriment à peu près de la même façon. Cela est logique puisque, s'il n'y a pas de régulation, les gènes vont simplement produire leur concentration de saturation de protéines.

La dynamique du "toggle switch" décrite par [Gardner *et al.*, 2000] est la suivante : si les répressions sont suffisamment fortes, alors le système est stable dés que l'un des gènes s'exprime fortement

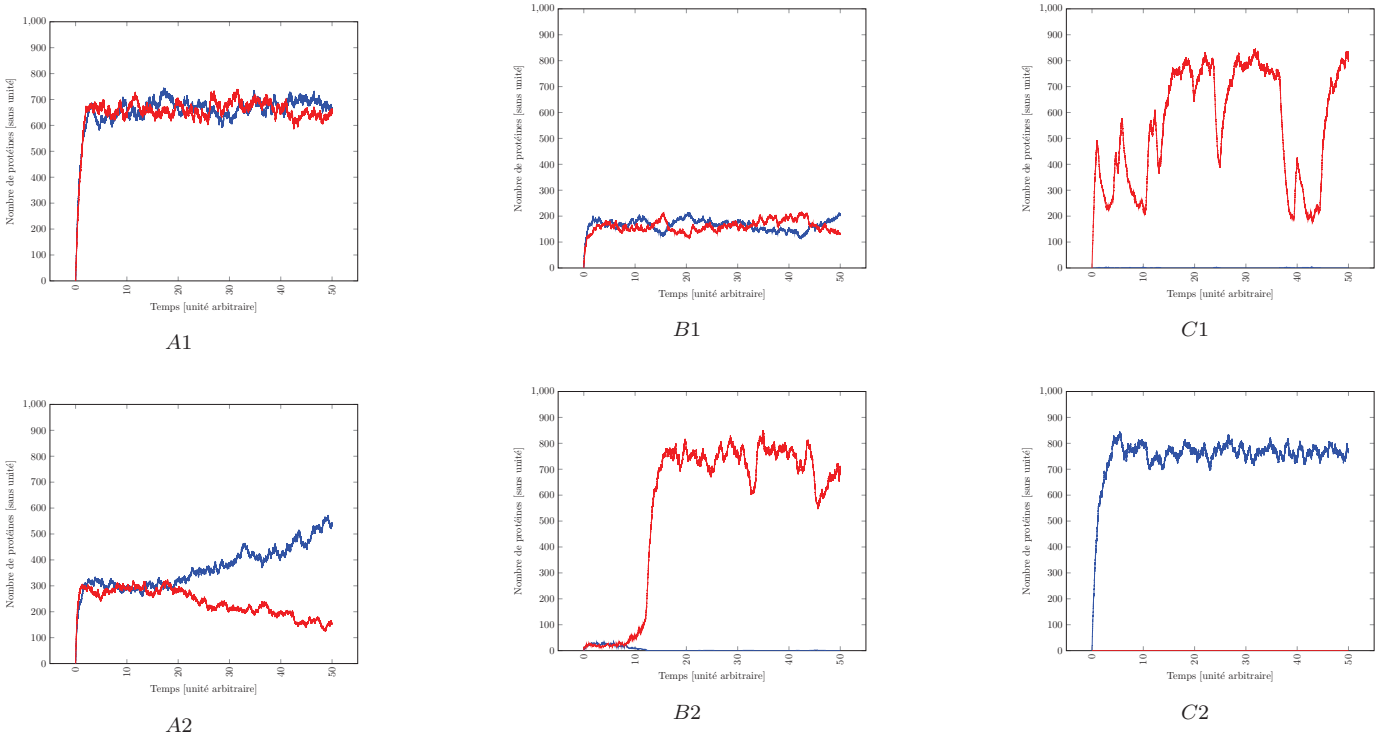
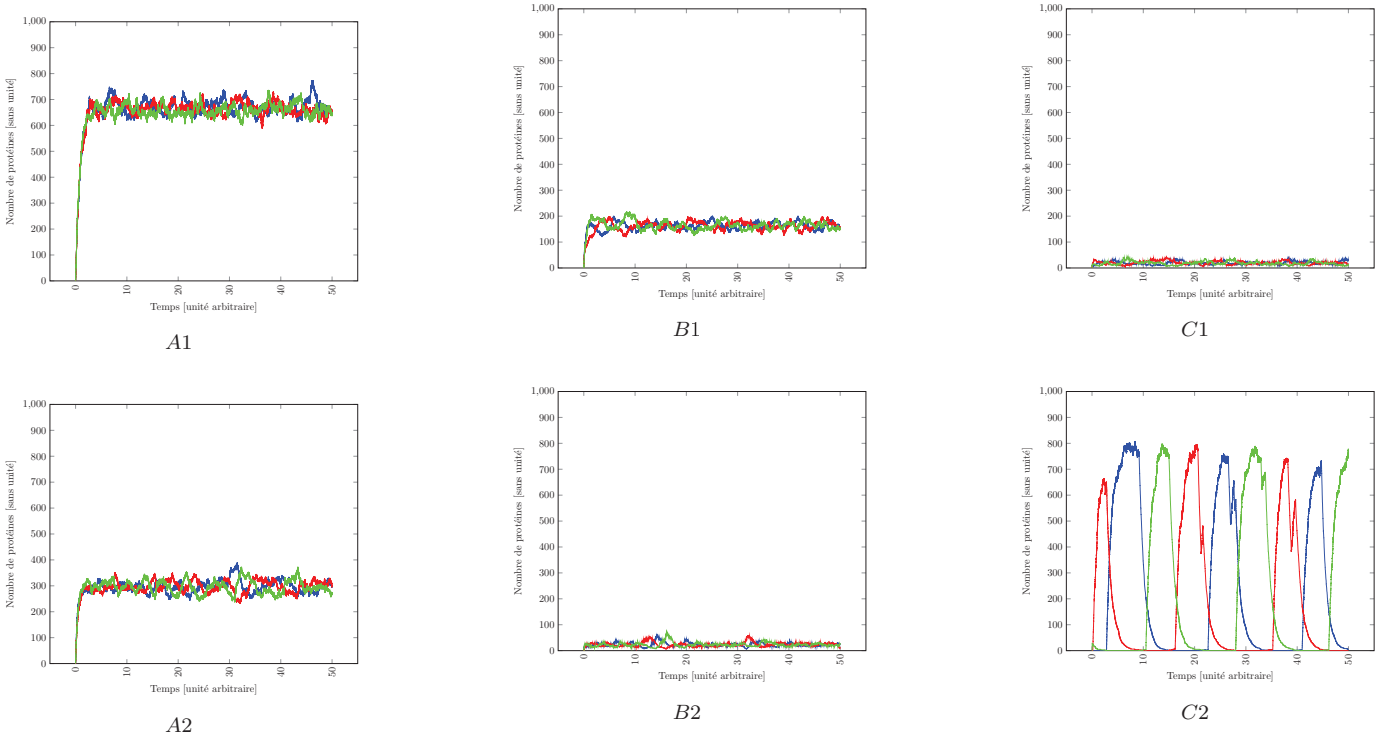


FIGURE 8 – Toggle switch
Les graphes du haut ($A1$, $B1$ et $C1$) correspondent au réseau de la partie gauche de la figure 6, ceux du bas ($A2$, $B2$ et $C2$) à celui de la partie droite. De gauche à droite, la valeur du paramètre e^{ε/k_BT} (où ε désigne l'énergie de liaison protéines-ADN) vaut 0.001,0.1,10.

par rapport à l'autre, et cela arrive rapidement, comme le montrent les graphiques. Il est toutefois possible en pratique que les expressions des gènes s'inversent (ie un gène est fortement exprimé, puis cette expression faiblit en faveur de celle de l'autre gène), d'où le nom "toggle switch", et c'est aussi quelque chose que l'on peut voir dans des simulations plus longues que celles de la figure 8, et que l'on voit presque avec les pics de décroissance du graphe $C1$.

Pour le réseau de la figure 7, selon [Elowitz et Leibler, 2000], si les répressions sont efficaces, le réseau doit avoir un comportement oscillatoire, c'est-à-dire que les gènes vont à tour de rôle s'exprimer fortement au détriment des autres. Ceci correspond bien à ce que l'on voit dans le graphique $C2$ de la figure 9.

On peut donc en conclure que le modèle thermodynamique a l'air de bien représenter la dynamique des réseaux de régulation.



$A1$ $B1$ $C1$

$A2$ $B2$ $C2$

FIGURE 9 – Repressilator

Les graphes du haut ($A1$, $B1$ et $C1$) correspondent au réseau de la partie gauche de la figure 6, ceux du bas ($A2$, $B2$ et $C2$) à celui de la partie droite. De gauche à droite, la valeur du paramètre e^{ε/k_BT} (où ε désigne l'énergie de liaison protéines-ADN) vaut 0.001,0.1,10.

4 Synthèse de paramètres qualitatifs

Le problème que l'on veut résoudre avec ce modèle est le suivant :

$$\begin{cases} \textbf{INPUT} : & \text{un réseau de régulation de gènes et une propriété sur ce réseau} \\ \textbf{OUTPUT} : & \text{des contraintes sur les paramètres sous lesquelles la propriété est satisfaite} \end{cases}$$

En pratique une propriété d'un réseau est un ensemble de transitions de la chaîne de Markov, dont on veut spécifier des conditions sur leurs probabilités comme : la probabilité est nulle, strictement postive, strictement plus petite qu'un ou égal à un.

Pour vérifier ce type de conditions sur les probabilités, il faut générer des contraintes asymptotiques. Ce que nous avons choisi, c'est de représenter ces contraintes sous forme d'inégalités asymptotiques entre les paramètres du réseau de régulation (une inégalité asymptotique permet de dire qu'un paramètre est négligeable comparé à un autre).

Résoudre le problème avec des inégalités asymptotiques est intéressant, car il est beaucoup plus simple de mettre en place des expériences en demandant seulement quels paramètres doivent être meilleurs que quels autres, plutôt qu'en demandant des valeurs exactes pour chaque paramètre. De

plus, dans un réseau de régulation de gènes, dire qu'une régulation est efficace revient en fait à dire qu'elle est plus efficace que d'autres. Il est donc plus intéressant d'avoir des contraintes relatives entre les paramètres.

Pour faire cette modélisation, on a besoin de simplifier le modèle. Au lieu de considérer comme états les éléments de $\mathbb{N}^G$on va simplement considérer les éléments de $\{0,1\}^G$. Au lieu de compter les protéines pour chaque gène, on va seulement regarder pour chaque gène, s'il est actif ou non. L'hypothèse de modélisation que l'on utilise ici est la suivante : quand un ARN polymérase se lie au site promoteur d'un gène, ce gène produit rapidement sa quantité de saturation de protéines, et quand il n'y a pas d'ARN polymérase lié, la quantité de protéines présentes tombe vite à zéro. Cette hypothèse n'est pas absurde, en pratique certains réseaux peuvent avoir un tel comportement, comme on peut le voir dans les graphes $C2$ des figure 8 et 9. De plus, on va aussi supposer que l'échelle de temps est discrète et que les états des gènes se mettent à jour de manière synchrone.

Ainsi on peut représenter la dynamique d'un réseau de régulation avec une chaîne de Markov à temps discret (cf annexe) avec un nombre fini d'états. Les états sont les éléments de $\{0,1\}^G$, et on définit la probabilité de la transition $\sigma \to \sigma'$ par :

$$\prod_{g \in G} {\mathbb{P}'_{on}}^{\sigma}(g)$$

avec ${\mathbb{P}'_{on}}^{\sigma}(g) = \begin{cases} \mathbb{P}^{\sigma}_{on}(g) & \text{si } \sigma'(g) = 1 \\ 1 - \mathbb{P}^{\sigma}_{on}(g) & \text{sinon} \end{cases}$

Définissons maintenant les paramètres utiles à l'algorithme :

Pour chaque gène A, on introduit le paramètre $P_A = [ARNP] \times e^{\varepsilon_{ARNP-A}/k_B T}$ (ie la concentration en ARN polymérase multipliée par l'affinité de liaison entre l'ARN polymérase et le site promoteur de A), avec ε_{ARNP-A} l'énergie de liaison entre l'ARN polymérase et le site promoteur de A.

Pour chaque régulation (activation ou répression) $r = (A, r')$, on introduit $A_r = [A] \times e^{\varepsilon_{Ar}/k_B T}$ (ie la concentration de protéines produite par le gène A multipliée par l'affinité de liaison entre ces protéines et le site opérateur de r).

Pour chaque activation $(A, (B, r))$, on introduit le paramètre $A_B = e^{\varepsilon_{A-B}/k_B T}$ (ie l'affinité de liaison entre les protéines produites par A et celles produites par B).

4.1 Algorithme

L'idée de l'algorithme n'est pas de générer directement un ensemble d'inégalités asymptotiques, mais de générer une formule logique dont les propositions atomiques sont des inégalités asymptotiques entre les paramètres, et dont les modèles sont les conditions sous lesquelles la propriété étudiée est satisfaite.

4.1.1 Encodage

Pour chaque transition de l'entrée du problème (donc de la forme $(\sigma_g)_{g \in G} \to (\sigma'_g)_{g \in G}$), et pour chaque gène g, on considère la probabilité $\mathbb{P} = \mathbb{P}^{\sigma}_{on}(g) = \frac{Z_{on}}{Z_{on}+Z_{off}} = \frac{1}{1+x/y}$ où x et y sont des expressions mathématiques sur les paramètres du réseau de régulation ($x = Z_{off}$ et $y = Z_{on}$). Ce que l'on veut faire, c'est construire des formules logiques à partir des conditions : $\mathbb{P}$ tend vers 1 et $\mathbb{P}$ tend vers 0. L'idée pour faire cela, est de dire :

— "$\mathbb{P}$ tend vers 1" signifie $y \gg x$ (ie y est beaucoup plus grand que x)

— "$\mathbb{P}$ tend vers 0" signifie $x \gg y$ (ie x est beaucoup plus grand que y)

Il ne reste plus qu'à déconstruire les expressions de x et y pour obtenir des formules logiques où les variables sont des inégalités entre les paramètres, et non entre des sommes et/ou des produits de paramètres. Pour plus de clarté, on introduit la notation $x \gtrsim y \equiv \neg(x \ll y)$ ce qui signifie que x et y sont du même ordre ou que x est beaucoup plus grand que y. L'idée est simplement d'appliquer récursivement les règles suivantes (les règles sont données par ordre de priorité, la première est prioritaire) :

— $(x + y \ll z) \to (x \ll z) \wedge (y \ll z)$

— $(x \ll y + z) \to (x \ll y) \vee (x \ll z)$

— $(x + y)z \ll w \to xz + yz \ll w$

— $x \ll (y + z)w \to x \ll yw + zw$

— $p_1 p_2 \dots p_n \ll q_1 q_2 \dots q_n \to \bigvee_{\sigma \in \mathfrak{S}_n} \bigvee_{i=1}^{n} \left[(p_i \ll q_{\sigma(i)}) \wedge \bigwedge_{j \neq i} (p_j \lesssim q_{\sigma(j)}) \right]$

— $p_1 p_2 \dots p_n \ll q_1 q_2 \dots q_m \to p_1 p_2 \dots p_n \ll q_1 q_2 \dots q_m . \underbrace{1.1 \dots 1}_{n-m}$ si $n > m$

— $p_1 p_2 \dots p_n \ll q_1 q_2 \dots q_m \to p_1 p_2 \dots p_n . \underbrace{1.1 \dots 1}_{m-n} \ll q_1 q_2 \dots q_m$ si $n < m$

Les règles concernant les sommes sont simples, car la somme est considérée comme un opérateur max, puisqu'on fait des "calculs asymptotiques". La règle du produit est plus compliquée, elle dit la chose suivante : pour dire qu'un produit $p_1 p_2 \dots p_n$ est beaucoup plus petit qu'un autre $q_1 q_2 \dots q_n$, il faut faire correspondre deux-à-deux les termes de chaque produit (ie trouver un élément σ de $\mathfrak{S}$) de telle sorte que, pour un indice i on a une grande inégalité $p_i \ll q_{\sigma(i)}$, et tel qu'aucun autre indice j ne renverse l'inégalité (ie $p_j \lesssim q_{\sigma(j)}$).

Comme la règle du produit est plus compliquée, elle est aussi moins sûre, car on exige des termes qu'ils correspondent deux-à-deux, alors qu'il est envisageable en pratique que le produit de deux termes soit du même ordre qu'un terme de l'autre produit. Ainsi pour éviter de propager une erreur faite au début du calcul, on n'applique la règle du produit qu'à la fin, sur des produits dont tous les termes sont directement des paramètres, et non des expressions dépendant de plusieurs paramètres. Cela est fait à l'aide de la troisième règle et de la quatrième règle.

En résumé, la règle du produit est moins sûre car la formule que l'on obtient n'est pas équivalente à la formule de départ. Toutefois la formule d'arrivée implique la formule de départ. Donc finalement, la formule obtenue à la fin de l'algorithme va bien impliquer la formule correspondant à la propriété de départ. Autrement dit, les modèles de la formule finale vont bien représenter des ensembles de contraintes sous lesquelles la propriété spécifiée en entrée est valide, mais il risque d'y avoir des ensembles de contraintes qui ne correspondent à aucun modèle de cette formule.

Dans les deux dernières règles, les paramètres "1" que l'on ajoute ont bien un sens en pratique, puisqu'il s'agit du poids de la configuration vide du modèle thermodynamique.

L'algorithme qui calcule la formule logique associée à un ensemble S de transitions est décrit ci-dessous :

— Étant donné une transition $\sigma \to \sigma'$ et un gène g, on calcule $\varphi^g_{\sigma,\sigma'}$ la formule associée à la condition $(P^\sigma_{on}(g) \to 1)$ si $\sigma'(g) = 1$, et associée à $(P^\sigma_{on}(g) \to 0)$ sinon. On peut aussi demander $(P^\sigma_{on}(g) > 0)$ au lieu de $(P^\sigma_{on}(g) \to 1)$, en définissant $(P^\sigma_{on}(g) > 0) \equiv \neg(P^\sigma_{on}(g) \to 0)$, et de même avec $(P^\sigma_{on}(g) < 1)$ au lieu de $(P^\sigma_{on}(g) \to 0)$.

— Pour toutes les transitions $t = (\sigma \to \sigma')$, on définit $\psi_t = \bigwedge_{g \in G} \varphi^g_{\sigma,\sigma'}$

— Finalement, la formule associée à l'ensemble de transition S est :

$$\bigwedge_{t \in S} \psi_t$$

4.1.2 Théorie

Si on considère simplement la formule précédente, on peut avoir des modèles qui n'ont pas de sens. Par exemple, si x et y sont des paramètres, alors $(x \ll y)$ et $(y \ll x)$ sont des variables logiques, mais les assignations de valeurs de vérités qui rendent vraies ces deux variables en même temps n'ont pas de sens. C'est pour cela qu'il faut aussi générer une formule qui représente la théorie mathématique d'un ordre. Cette théorie est la conjonction des formules suivantes :

— $\neg((x \ll y) \wedge (y \ll x))$ $\forall x, y$ paramètres
— $\neg(x \ll x)$ $\forall x$ paramètres
— $((x \ll y) \wedge (y \ll z)) \Rightarrow (x \ll z)$ $\forall x, y, z$ paramètres
— $((x \ll y) \wedge (y \lesssim z)) \Rightarrow (x \ll z)$ $\forall x, y, z$ paramètres
— $((x \lesssim y) \wedge (y \ll z)) \Rightarrow (x \ll z)$ $\forall x, y, z$ paramètres
— $((x \lesssim y) \wedge (y \lesssim z)) \Rightarrow (x \lesssim z)$ $\forall x, y, z$ paramètres
— $(A_B \gtrsim 1)$ $\forall A, B$ gènes

La dernière règle vient du fait que A_B est une exponentielle d'une énergie positive.

4.2 Expériences

Afin d'examiner la pertinence de l'algorithme, on a comparé ses résultats avec des résultats obtenus expérimentalement.

4.2.1 Interprétation des cas de base

Tout d'abord, on va examiner des réseaux de régulation simples, pour mieux comprendre le sens des formules logiques. Dans la suite, quand on calculera des probabilités d'activation $\mathbb{P}_{on}$, on calculera la probabilité dans l'état où tous les gènes sont actifs (ie on écrit $\mathbb{P}_{on}$ au lieu de $\mathbb{P}^{\sigma}_{on}$, avec $\sigma_g = 1$ pour tout gène g).

Si un gène A n'est régulé par aucun gène (cf figure 10), alors on a :

$$\mathbb{P}_{on}(A) \to 1 \Leftrightarrow P_A \gg 1$$

et

$$\mathbb{P}_{on}(A) \to 0 \Leftrightarrow P_A \ll 1$$

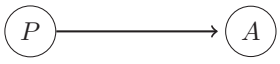


FIGURE 10 – absence de régulation

On voit donc que l'on peut spécifier la force d'un promoteur sans régulation, en comparant P_A avec le paramètre 1.

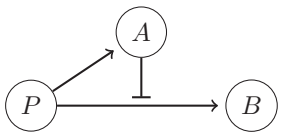


FIGURE 11 – répression simple

Si maintenant, on a un gène B régulé par une répression $r = (A, (P, B)) \in R^-$ (cf figure 11), on a alors :

$$\mathbb{P}_{on}(B) \to 1 \Leftrightarrow (1 \ll P_B) \wedge (A_r \ll P_B)$$

La signification de la formule est la suivante : pour que le gène B soit sûr d'être activé malgré la répression, il faut que que le promoteur du gène soit naturellement efficace (ie $1 \ll P_B$) et qu'il soit meilleur que la répression (ie $A_B \ll P_B$).

On trouve une formule similaire avec l'autre condition :

$$\mathbb{P}_{on}(B) \to 0 \Leftrightarrow (1 \gg P_B) \vee (A_r \gg P_B)$$

Ici, on voit que la condition est "plus simple" à réaliser puisqu'il suffit que le promoteur soit naturellement faible ou plus faible que la répression.

Si maintenant, on a un gène B régulé par une activation $r = (A, (P, B)) \in R^+$ (cf figure 12), on a alors :

$$\mathbb{P}_{on}(B) \to 1 \Leftrightarrow (P_B \gg 1) \vee [(P_B \gtrsim 1) \wedge (A_r \gtrsim 1) \wedge (A_B \gg 1)]$$

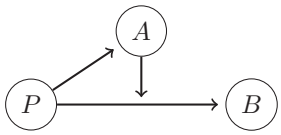


FIGURE 12 – activation simple

La signification de cette formule est : soit B est sûr de s'activer même sans l'aide de A (ie $P_B \gg 1$) soit B peut plus ou moins être actif seul (ie $P_B \gtrsim 1$) et les protéines de A peuvent participer à la régulation (ie $A_r \gtrsim 1$) et cette participation est très efficace (ie $A_B \gg 1$).

On a aussi :

$$\mathbb{P}_{on}(B) \to 0 \Leftrightarrow (P_B \ll 1) \wedge [(A_r \ll 1) \vee [(A_r \lesssim 1) \wedge (A_B \lesssim 1)]]$$

La signification de cette formule est : B est très faible sans aide (ie $P_B \ll 1$), et soit A ne peut pas participer à la régulation (ie $A_r \ll 1$), soit il est possible que A participe un peu (ie $A_r \lesssim 1$) mais cette régulation n'est pas efficace (ie $A_B \lesssim 1$).

4.2.2 Repressilator

Pour tester notre algorithme, nous avons comparé nos résultats avec ceux de [Elowitz et Leibler, 2000], concernant le réseau de la figure 13. Dans ce papier, il est écrit que le circuit peut osciller, et que ce comportement est favorisé par les conditions suivantes :

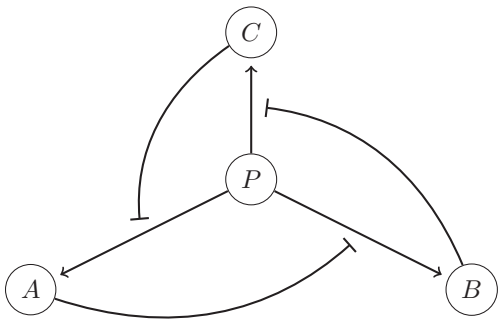


FIGURE 13 – repressilator

— des promoteurs forts (ie $(P_A \gg 1) \wedge (P_B \gg 1) \wedge (P_C \gg 1)$)

— des répressions élevées par rapport aux promoteurs (ie $(P_A \ll C_c) \wedge (P_B \ll A_a) \wedge (P_C \ll B_b)$)

Remarque : on note a la répression $(A, (P, B))$, b la répression $(B, (P, C))$ et c la répression $(C, (P, A))$.

Appelons H la conjonction de ces conditions et T la formule représentant la théorie mathématique.

Ce que l'on appelle une oscillation est une dynamique du réseau dans laquelle on a :

$$100 \rightarrow^* 001 \rightarrow^* 010 \rightarrow^* 100$$

Remarque : on note les états avec des nombre à trois bits, comme 100, au lieu de $(A = 1, B = 0, C = 0)$.

La façon la plus naturelle d'avoir une oscillation dans la topologie du "repressilator" est avec le chemin suivant :

$$100 \rightarrow 101 \rightarrow 001 \rightarrow 011 \rightarrow 010 \rightarrow 110 \rightarrow 100$$

Ce chemin est naturelle, car sous les hypothèses H, un gène non-réprimé sera actif dans le prochain état, et un gène réprimé sera inactif.

Appelons φ la formule que nous donne notre algorithme à partir de l'ensemble de transitions $\{100 \rightarrow 101, 101 \rightarrow 001, 001 \rightarrow 011, 011 \rightarrow 010, 010 \rightarrow 110, 110 \rightarrow 100\}$, de telle sorte que chacune de ces transitions aient une probabilité 1 dans la chaîne de Markov.

On obtient que la formule $T \Rightarrow (H \Leftrightarrow \varphi)$ est valide. Ce qui signifie que les conditions que l'on obtient pour que le système oscille sont équivalentes aux conditions obtenues expérimentalement dans [Elowitz et Leibler, 2000].

4.2.3 Toggle switch

Nous avons aussi comparé nos résultats avec ceux de [Gardner *et al.*, 2000] à propos de la "bistabilité" du réseau de la figure 14. La bistabilité est la stabilité des deux états $(A = 0, B = 1)$ et $(A = 1, B = 0)$ (ces états seront abrégés respectivement en 01 et 10). La bistabilité est donc la propriété liée à l'ensemble de transitions $\{01 \rightarrow 01; 10 \rightarrow 10\}$.

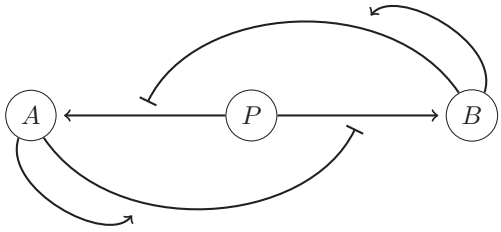


FIGURE 14 – toggle switch

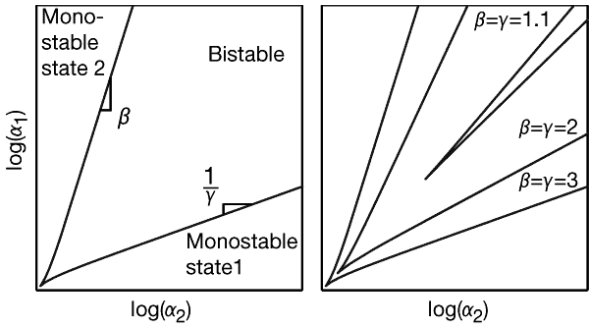


FIGURE 15 – la bistabilité du "toggle switch"
β et γ décrivent l'effet de l'énergie apportée par les liaisons protéines-protéines. Ici, on a donc $\gamma = A_A$ et $\beta = B_B$ avec nos notations. L'image de gauche montre les conditions sous lesquelles il y a une "bistabilité" du système, et les conditions sous lesquelles seul un état et stable, parmi 10 et 01.

Les expériences de [Gardner *et al.*, 2000] sont basées sur des équations de la génétique qui reposent sur deux paramètres α_1, α_2 qui représentent respectivement la vitesse de synthèse des protéines par A et B.

Pour faire nos comparaisons, l'idée est de diviser le graphe de la figure 15 en trois zones grâce à des formules : en haut à gauche, en haut à droite et en bas à droite. Pour simplifier les notations, appelons a_1 le paramètre lié à la repression de B par A (ie $a_1 = A_{(A,(P,B))}$), et a_2 le paramètre lié à la seconde régulation de A (ie $a_2 = A_{(A,(A,(P,B)))}$). On définit de la même façon b_1 et b_2. Et on note $b = B_B$ ainsi que $a = A_A$.

On définit le haut du graphe par la formule

$$H_a = (a_1 \gtrsim P_B) \wedge (a_2 \gtrsim 1)$$

En effet, le haut du graphe représente les conditions où le gène A est efficace, c'est-à-dire sa répression est utile ($a_1 \gtrsim P_B$) et son activation aussi ($a_2 \gtrsim 1$).

On définit de même la partie droite du graphe

$$H_b = (b_1 \gtrsim P_A) \wedge (b_2 \gtrsim 1)$$

Par symétrie, on définit la partie basse

$$H_{\overline{a}} = (a_1 \lesssim P_B) \wedge (a_2 \lesssim 1)$$

et la partie gauche

$$H_{\overline{b}} = (b_1 \lesssim P_A) \wedge (b_2 \lesssim 1)$$

On peut remarquer que $H_a \not\equiv \neg H_{\overline{a}}$. La raison est que l'on ne cherche pas à partitionner le graphe 15 avec ces zones, mais simplement à définir certaines zones.

On peut maintenant définir la partie supérieure gauche par

$$H_{a\overline{b}} = H_a \wedge H_{\overline{b}}$$

ainsi que la partie inférieure droite par

$$H_{\overline{a}b} = H_{\overline{a}} \wedge H_b$$

et la partie supérieure droite par

$$H_{ab} = H_a \wedge H_b$$

Pour vérifier si nos résultats sont cohérents avec ceux de [Gardner *et al.*, 2000], on distingue trois cas : la "zone de bistabilité" est grande ($E_3 = (a \gg 1) \wedge (b \gg 1)$), la "zone de bistabilité" est petite ($E_2 = (a \lesssim 1) \wedge (b \lesssim 1)$) et la taille de la zone n'est pas spécifiée ($E_1 = \top$). Définissons de plus la formule $H_0 = (P_A \gg 1) \wedge (P_B \gg 1)$ qui signifie que les promoteurs sont naturellement forts (ie si un gène n'est pas réprimé, il est sûr de s'activer), ce qui est une hypothèse dans [Gardner *et al.*, 2000].

Considérons $\varphi_b^{=1}$ la formule produite par notre algorithme à partir de l'ensemble $\{01 \to 01; 10 \to 10\}$ associée à des transitions de probabilités un, $\varphi_b^{>0}$ la formule liée aux mêmes transitions mais avec des probabilités non-nulles. Considérons aussi les formules $\varphi_{ma}^{=1}$ et $\varphi_{ma}^{>0}$ liées à l'ensemble $\{10 \to 10; 01 \to 11\}$ avec respectivement des transitions de probabilités 1 et de probabilités non-nulles. Ce dernier ensemble de transitions correspond à la propriété de monostabilité de A (c'est-à-dire la stabilité de 10, mais pas celle de 01). On peut remarquer que pour interdire la transition $01 \to 01$, il suffit d'ajouter dans l'ensemble des transitions $01 \to 11$, puisque les deux transitions $01 \to 00$, $01 \to 10$ ont une probabilité 0 dès que H_0 est vraie (ie un gène qui n'est pas réprimé doit s'activer). Et donc, dire que $01 \to 01$ a une probabilité 0 revient à dire que $01 \to 11$ a une probabilité 1. De même $01 \to 01$ a une probabilité différente de 1 revient à dire que $01 \to 11$ a une probabilité non nulle.

Finalement, pour $\varphi \in \{\varphi_b, \varphi_{ma}\}$, définissons $\varphi^{=0} = \neg\varphi^{>0}$ et $\varphi^{<1} = \neg\varphi^{=1}$.

Pour chaque $E \in \{E_1, E_2, E_3\}, H \in \{H_{ab}, H_{a\overline{b}}, H_{\overline{a}b}\}, \varphi \in \{\varphi_b, \varphi_{ma}\}$ et $c \in \{=0, >0, <1, =1\}$, on a vérifié la validité de la formule

$$(E \wedge H \wedge H_0) \Rightarrow \varphi^c$$

Les résultats sont résumés dans les tableaux 1 et 2.

Sur le tableau 1, on voit que la bistabilité a toujours une probabilité positive seulement dans l'hypothèse H_{ab} (ie quand on se place dans la partie supérieure droite), qu'elle n'est jamais sûre dans l'hypothèse E_2 (ie quand la zone de bistabilité est petite) quand on se place dans la partie supérieure gauche ($H_{a\overline{b}}$) ou dans la partie inférieure droite ($H_{\overline{a}b}$) du graphe 15, et enfin qu'elle est toujours sûre sous les hypothèses $H_{ab} \wedge E_3$.

c / E,H	$= 0$	> 0	< 1	$= 1$
E_1, $H_{a\bar{b}}$	X	X	X	X
E_1, H_{ab}	X	V	X	X
E_1, $H_{\bar{a}b}$	X	X	X	X
E_2, $H_{a\bar{b}}$	X	X	V	X
E_2, H_{ab}	X	V	X	X
E_2, $H_{\bar{a}b}$	X	X	V	X
E_3, $H_{a\bar{b}}$	X	X	X	X
E_3, H_{ab}	X	V	X	V
E_3, $H_{\bar{a}b}$	X	X	X	X

Tableau 1 – Validité de $(H_0 \wedge H \wedge E \Rightarrow \phi_b^c)$

c / E,H	$= 0$	> 0	< 1	$= 1$
E_1, $H_{a\bar{b}}$	X	X	X	X
E_1, H_{ab}	X	X	V	X
E_1, $H_{\bar{a}b}$	X	X	V	X
E_2, $H_{a\bar{b}}$	X	V	X	X
E_2, H_{ab}	X	X	V	X
E_2, $H_{\bar{a}b}$	X	X	V	X
E_3, $H_{a\bar{b}}$	X	X	X	X
E_3, H_{ab}	V	X	V	X
E_3, $H_{\bar{a}b}$	V	X	V	X

Tableau 2 – Validité de $(H_0 \wedge H \wedge E \Rightarrow \phi_{ma}^c)$

Une chose intéressante à noter dans la première colonne du second tableau est que si on se place dans la parite droite (H_{ab} ou $H_{\bar{a}b}$) avec une grande zone de bistabilité (E_3), alors on ne peut pas avoir la monostabilité de A, puisque sous ces hypothèses on aura la stabilité de B. Si on considère la petite zone de bistabilité (E_2), en revanche, on peut observer la monostabilité de A sous l'hypothèse $H_{a\bar{b}}$.

Ainsi, on voit bien que les formules produites par notre algorithme parviennent à décrire les différentes zones du graphe 15, selon les expériences de [Gardner *et al.*, 2000].

Idée de perspective

On peut améliorer cette modélisation en essayant d'enlever une hypothèse de modélisation, à savoir qu'un gène n'a que deux niveaux d'expressions. C'est-à-dire, soit il est actif et il produit sa quantité maximal de protéines immédiatement, soit il est inactif et ne produit rien. Une méthode possible pour se passer de cette hypothèse est celle expliquée dans cette section. Toutefois elle n'est pas encore bien développée et n'a pas été testée sur beaucoup d'exemples.

On a déjà vu sur les graphes des figures 8 et 9, que cette hypothèse modélise un phénomène qui a un sens, toutefois il arrive aussi que ce ne soit pas le cas, on peut par exemple regarder le graphe de droite de la figure 1 du papier [Elowitz et Leibler, 2000].

L'idée est simplement de considérer un nombre fini de niveaux d'expression pour chaque gène. Le problème qu'il faut régler est de savoir combien de niveaux il faut introduire, et qu'est-ce que ces niveaux représentent. Dans la méthode à laquelle nous avons pensée, il faut supposer que dans le réseau de régulation, chaque gène n'est régulé que par au plus un gène (ce qui est le cas avec le "toggle switch" figure 14 et le "repressilator" figure 13). Il faut aussi fixer la valeur des paramètres autres que les concentrations de protéines produites par les gènes. Décrivons cette méthode et illustrons la avec l'exemple du "toggle switch" de la figure 14.

Reprenons les notations de la section 4.2.3, pour introduire les notations suivantes $b_2 = [B]b'_2$ et $b_1 = [B]b'_1$ (b'_1 et b'_2 sont donc des exponentielles d'énergie de liaison protéines-ADN). Concentrons nous sur les régulations du gène A. $Z_{on}(A)$ et $Z_{off}(A)$ sont des fonctions dont la seule variable est $[B]$ (tous les autres paramètres comme les exponentielles d'énergie de liaison et $[ARNP]$ sont constants).

$$Z_{on}(A) = P_A + P_A[B]b'_2$$

$$Z_{off}(A) = 1 + [B]b'_1 + [B]^2 b'_1 b'_2 b$$

Posons $f([B]) = \frac{Z_{on}(A)}{Z_{off}(A)}$.

Fixons un nouveau paramètre $M > 1$. L'interprétation de ce paramètre est la suivante : si on a $f([B]) > M$ alors on considère que $\mathbb{P}_{on}(A) \to 1$, si $f([B]) < 1/M$ alors $\mathbb{P}_{on}(A) \to 0$, et si $1/M \leq f([B]) \leq M$ alors $0 < \mathbb{P}_{on}(A) < 1$. Cette interprétation a un sens car

$$f([B]) > M \Rightarrow \mathbb{P}_{on}(A) = \frac{1}{1 + 1/f([B])} > \frac{M}{M+1}$$

et

$$f([B]) < \frac{1}{M} \Rightarrow \mathbb{P}_{on}(A) = \frac{1}{1 + 1/f([B])} < \frac{1}{M+1}$$

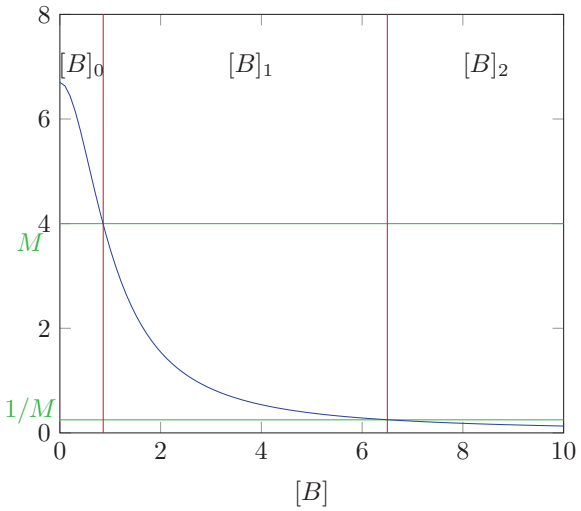


FIGURE 16 – graphe de f

Dans la suite on notera $\mathbb{P}_{on}(A) \to 1$ lorsque $f([B]) > M$, $\mathbb{P}_{on}(A) \to 0$ lorsque $f([B]) < 1/M$, et $0 < P_{on}(A) < 1$ lorsque $1/M \leq f([B]) \leq M$.

Ainsi, il suffit de choisir un M suffisamment grand, de telle sorte que $\frac{M}{M+1}$ soit "proche" de 1 et que $\frac{1}{M+1}$ soit "proche" de 0. Traçons la courbe de f sur la figure 16. Découpons le graphe horizontalement en trois zones : celle au-dessus de la courbe $y = M$, celle en-dessous de la courbe $y = 1/M$, et celle entre ces deux courbes (cf lignes vertes horizontales de la figure 16). Puis, découpons le graphe verticalement en autant de zones que le nombre de fois où la courbe change de zone horizontale (cf les lignes rouges verticales de la figure 16). Ces zones verticales représentent les différents niveaux d'expression des gènes. Ainsi dans l'exemple, il y a trois niveaux d'expression :

— si le niveau d'expression de B est à $[B]_0$, on a $\mathbb{P}_{on}(A) \to 1$ (car $f([B]_0) > M$)

— s'il est à $[B]_1$ on a $0 < \mathbb{P}_{on}(A) < 1$ (car $1/M \leq f([B]_1) < M$)

— s'il est à $[B]_2$ on a $\mathbb{P}_{on}(A) \to 0$.

On peut définir par symétrie les trois niveaux d'expressions de A : $[A]_0, [A]_1$ et $[A]_2$.

L'idée avec ces niveaux d'expression est de définir une chaîne de Markov à temps discret, dont les états sont les éléments de $\{[A]_0, [A]_1, [A]_2\} \times \{[B]_0, [B]_1, [B]_2\}$ (ie un état est la donnée d'un niveau d'expression pour A et d'un niveau pour B). Lorsque le système est dans un certain état, si $P_{on}(A) \to 1$ les états accessibles en une transition ne peuvent être que ceux où le niveau de A est incrémenté (sauf si A est déjà à son niveau maximum, dans ce cas le niveau reste le même), si $P_{on}(A) \to 0$ les états accessibles en une transition sont ceux où le niveau de A est décrementé, et si $0 < P_{on}(A) < 1$ alors le niveau de A peut soit être incrémenté soit être décrémenté. La chaîne de Markov associée au "toggle switch" est dessinée à la figure 17. On ne dessine que les transitions qui ont une probabilité non-nulle, mais on ne représente pas ces probabilités.

Cette chaîne de Markov est intéressante, car d'après le thérorème 1 (cf annexe, classification des états d'une chaîne de Markov), les simulations sur cette chaîne de Markov vont toujours se terminer soit sur l'état $([A]_2, [B]_0)$ soit sur l'état $([A]_0, [B]_2)$.

De plus, on peut comparer la structure de cette chaîne avec les graphes de la figure 8. En effet,

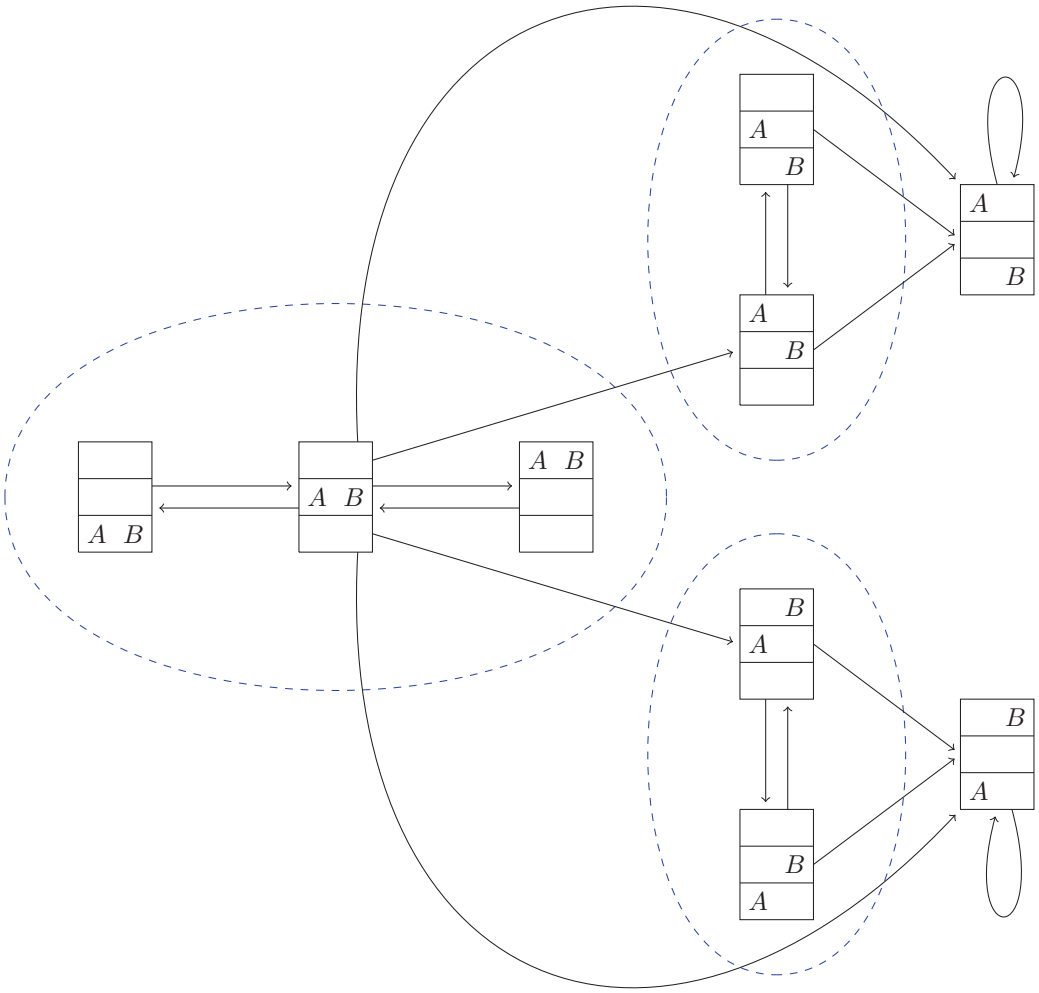


FIGURE 17 – Chaîne de Markov représentant la dynamique du "toggle switch"
Les zones entourées en bleu sont les classes transitoires de la chaîne de Markov.

dans cette chaîne il y a cinq classes d'équivalence (cf annexe, classification des états d'une chaîne de Markov) : la première dans laquelle, les gènes sont au même niveau d'expression, les deux suivantes dans lesquelles un des gènes s'exprime un peu plus que l'autre, et les deux dernières dans lesquelles un gène s'exprime fortement en réprimant complètement l'autre. On peut voir la phase où les deux gènes ont la même expression sur tous les graphes (particulièrement sur $A1, B1, A2$ et $B2$), de plus on voit bien la phase où un des deux gènes prend un ascendant sur le graphe $A2$ de la figure 8. On voit aussi que, comme dans la chaîne de Markov, on peut passer directement de la première phase (où les gènes s'expriment de la même façon) à une phase terminale (où un gène s'exprime fortement en réprimant l'autre) sur le graphe $C2$.

Cette méthode pourrait être un moyen de se passer de l'hypothèse qu'un gène n'a que deux niveaux d'expressions, puisqu'elle a l'air de bien modéliser au moins un réseau. Toutefois cette méthode ne fonctionne que sur peu de réseaux, les réseaux dans lesquels chaque gène régule au plus un gène, et chaque gène est régulé par au plus un gène. De plus, le fait d'avoir à fixer les valeurs des paramètres réduit les applications de la modélisation, mais on pourrait toujours s'en servir pour vérifier le comportement d'un réseau dont on connait les valeurs des paramètres.

Bilan et Perspectives

Ce stage a été intéressant et bénéfique puisqu'il m'a permis d'avoir des connaissances de bases en biologie génomique, ainsi qu'une première approche en bioinformatique. Au cours du stage, j'ai pu travailler sur une modélisation de la régulation de l'expression des gènes, en introduisant notamment une syntaxe générale pour représenter ces régulations. J'ai aussi pu vérifier que cette modélisation était cohérente avec la réalité, en effectuant des simulations avec l'agorithme de Gillespie et en les comparant avec les comportements observés expérimentalement.

J'ai de plus utilisé cette modélisation pour élaborer un algorithme qui, étant donné un réseau de régulation et un comportement dynamique du réseau, génère une formule logique dont chaque modèle représente des contraintes biologiques sous lesquelles ce comportement est valide. J'ai montré que cet algorithme permet à la fois de vérifier et de prédire le comportement des réseaux de régulation. Cet algorithme pourrait permettre de savoir comment instancier un réseau de régulation (ie quels gènes utilisés) pour imposer une certaine dynamique sur ce réseau. Cela pourrait aider à mettre en place des expériences permettant de vérifier les conséquences de certaines régulations sur l'organisme, par exemple.

Toutefois cette méthode dépend d'une hypothèse de modélisation : un gène n'a que deux niveaux d'expression. Cette hypothèse n'est pas invraisemblable, comme le montrent certaines expériences, mais il serait intéressant de développer des méthodes plus générales, comme celle expliquée précédemment. De plus, si j'avais eu plus de temps, il aurait été intéressant de discuter plus en détails de ce projet avec des biologistes qui font des expériences pratiques, afin de trouver des applications directes en biologie. Cependant les personnes avec qui on a parlé du projet, semblaient optimistes quant à ses applications. Une autre perspective de ce projet pourrait consister à représenter les effets d'une mutation sur un réseau de régulation (par exemple, en enlevant ou en ajoutant des régulations, et/ou en imposant des contraintes biologiques sous forme de formules logiques), afin d'examiner les changements dans la dynamique du réseau et de les comparer avec des résultats expérimentaux pour essayer de comprendre, par exemple, le rôle d'un réseau.

Références

[Bintu *et al.*, 2005] Bintu, L., Buchler, N. E., Garcia, H. G., Gerland, U., Hwa, T., Kondev, J. et Phillips, R. (2005). Transcriptional regulation by the numbers : models. *Current opinion in genetics & development*, 15(2):116–124.

[Elowitz et Leibler, 2000] Elowitz, M. B. et Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.

[Gardner *et al.*, 2000] Gardner, T. S., Cantor, C. R. et Collins, J. J. (2000). Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–342.

[Giacobbe *et al.*, 2015] Giacobbe, M., Guet, C. C., Gupta, A., Henzinger, T. A., Paixao, T. et Petrov, T. (2015). Model checking gene regulatory networks. *In Tools and Algorithms for the Construction and Analysis of Systems*, pages 469–483. Springer.

[Hermsen *et al.*, 2006] Hermsen, R., Tans, S. et Ten Wolde, P. R. (2006). Transcriptional regulation by competing transcription factor modules. *PLoS Comput Biol*, 2(12):e164.

[Monteiro, 2011] Monteiro, A. (2011). Gene regulatory networks reused to build novel traits. *Bioessays*, pages 181–186.

[Shea et Ackers, 1985] Shea, M. A. et Ackers, G. K. (1985). The or control system of bacteriophage lambda : A physical-chemical model for gene regulation. *Journal of molecular biology*, 181(2):211–230.

A Annexe : chaîne de Markov

Les chaînes de Markov permettent de simuler la dynamique d'un système stochastique. Il existe deux types de chaînes de Markov, celles à temps discret et celles à temps continue. Une chaîne de Markov (à temps discret ou continu) est une extension d'un graphe orienté, dont les nœuds représentent les états possibles du système et les transitions représentent les changements d'états.

L'idée d'une chaîne de Markov est que l'on a un système qui a tout moment est dans un certain état (ie un nœud du graphe) qui sera actualisé en parcourant le graphe de manière stochastique à partir des probabilités des transitions.

A.1 Chaîne de Markov à temps discret

Une chaîne de Markov à temps discret est une promenade aléatoire sur un graphe orienté dont les arcs sont étiquettés par des probabilités.

Formellement une chaîne de Markov à temps discret est un quadruplet (S, A, P, P_0) tel que :

— S est l'ensemble des états du système

— $A \subseteq S \times S$ est l'ensemble des transitions du système

— $P_0 : S \to [0, 1]$ tel que $\sum_{s \in S} P_0(s) = 1$, est la distribution initiale

— $P : A \to]0, 1]$ est la fonction qui associe à chaque transition la probabilité que le système change d'état en utilisant cette transition

— $\forall s \in S, \sum_{s' \in S} P(s, s') = 1$

On peut trouver un exemple de chaîne de Markov à la figure 18.

Une chaîne de Markov à temps discret définit donc une suite de variables aléatoires $(s_n)_{n \in \mathbb{N}}$ à valeurs dans S. Cette suite représente les états rencontrés lors d'une promenade aléatoire sur ce graphe, et est définie par :

— $\forall s \in S, \mathbb{P}(s_0 = s) = P_0(s)$

— $\forall n \in \mathbb{N}, \forall s, t \in S, \mathbb{P}(s_{n+1} = s | s_n = t) = P(t, s)$

On peut déduire du second point la formule :

$$\forall n \in \mathbb{N}, \forall s \in S, \mathbb{P}(s_{n+1} = s) = \sum_{t \in S} P(t, s)\mathbb{P}(s_n = t)$$

A.2 Chaîne de Markov à temps continu

Une chaîne de Markov à temps continu est aussi une promenade aléatoire sur un graphe orienté, mais la différence est que l'on va s'intéresser au temps passé sur les sommets du graphe.

Formellement une chaîne de Markov à temps continu est un quadruplet (S, A, Q, P_0) tel que :

— S est l'ensemble des états du système

— $A \subseteq S \times S$ est l'ensemble des transitions du système

— $P_0 : S \to [0, 1]$ tel que $\sum_{s \in S} P_0(s) = 1$, est la distribution initiale

— $Q : A \to \mathbb{R}_+^*$ est la fonction qui associe à chaque transition le taux avec lequel le système change d'état en utilisant cette transition

Une chaîne de Markov à temps continu définit deux suites de variables aléatoires $(s_n)_{n \in \mathbb{N}}$ à valeurs dans S et $(T_n)_{n \in \mathbb{N}}$ à valeurs dans $\mathbb{R}_+^*$. La suite $(s_n)_{n \in \mathbb{N}}$ représente les états rencontrés lors d'une promenade aléatoire sur ce graphe et la suite $(T_n)_{n \in \mathbb{N}}$ représente le temps passé sur chacun de ces états. Ces suites sont définies par :

— $\forall s \in S, \mathbb{P}(s_0 = s) = P_0(s)$

— $\forall n \in \mathbb{N}, \forall s, t \in S, \mathbb{P}(s_{n+1} = s | s_n = t) = \frac{Q(t,s)}{\sum_{s' \in S} Q(t,s')}$
— $\forall n \in \mathbb{N}, \forall s \in S, \forall T \geq 0, \mathbb{P}(T_n \leq T | s_n = s) = 1 - exp\left(-\left(\sum_{s' \in S} Q(s,s')\right) T\right)$

A.3 Classification des états d'une Chaîne de Markov

Étant donné une chaîne de Markov à temps discret, introduisons la relation $\rightarrow \subseteq S^2$ défini par : $i \rightarrow j$ si et seulement si j est accessible depuis i dans le graphe (S, A).

On peut alors définir la relation suivante $\leftrightarrow \subseteq S^2$ par :$i \leftrightarrow j$ si et seulement si $i \rightarrow j$ et $j \rightarrow i$. Cette relation est une relation d'équivalence (la preuve est immédiate).

La relation $\rightarrow$ s'étend alors aux classes d'équivalences de la realtion $\leftrightarrow$:

$$C \rightarrow C' \Leftrightarrow \exists (i,j) \in C \times C', i \rightarrow j \Leftrightarrow \forall (i,j) \in C \times C', i \rightarrow j$$

On distingue deux types de classes d'équivalences :
— les classes finales. C est une classe finale si : $\nexists C'$ classe, $C \rightarrow C'$
— les classes transitoires. C est une classe transitoire si : $\exists C'$ classe, $C \rightarrow C'$

Ces classes d'équivalence sont illustrées à la figure 18. Elles permettent d'énoncer un théorème que nous utiliserons :

Théorème 1 *Soit (S, A, P, P_0) une chaîne de Markov à temps discret tel que S est fini. Soit F l'ensemble des états dont la classe d'équivalence est finale. On a alors :*

$$\mathbb{P}(s_n \in F) \underset{n \rightarrow +\infty}{\longrightarrow} 1$$

Ce théorème signifie qu'une promenade aléatoire sur une chaîne de Markov fini à temps discret visitera toujours une classe finale au bout d'un temps fini (et y restera puisque la classe est finale).

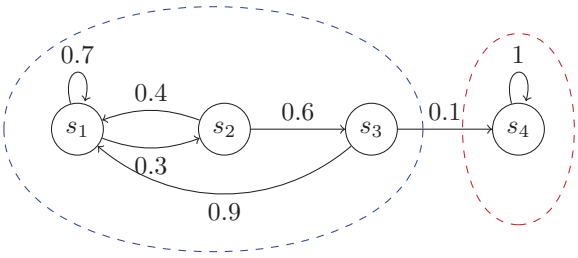


FIGURE 18 – Exemple de chaîne de Markov à temps discret (sans distribution initiale spécifiée) L'ensemble d'états entouré en bleu est une classe transitoire, celui entouré en rouge est une classe finale.

Glossaire

A

activation L'activation d'un gène B par un gène A réfère à l'activation du site promoteur de B par les protéines de A. Cela signifie que les protéines de A peuvent se lier à un site opérateur suffisamment proche du site promoteur de B pour que ces protéines aident l'ARN polymérase à se lier sur ce site promoteur en lui apportant des liaisons supplémentaires.

ARN polymérase Un ARN polymérase est une protéine pouvant se lier au site promoteur d'un gène pour initier le processus de synthèse d'une protéine.

E

expression L'expression d'un gène est la quantité de protéines qu'il produit.

R

régulation Une régulation peut être soit une activation soit une répression.

répression La répression d'un gène B par un gène A réfère à la repression du site promoteur de B par les protéines de A. Cela signifie que les protéines de A peuvent se lier à un site opérateur qui chevauche le site promoteur de B.

S

site opérateur Un site opérateur est une séquence de nucléotides sur l'ADN, à laquelle des protéines peuvent se lier et influencer l'expression d'un gène.

site promoteur Le site promoteur d'un gène est une séquence de nucléotides sur l'ADN. La liaison d'un ARN polymérase sur le site promoteur d'un gène permet d'initier la phase de fabrication d'une protéine.

T

traduction La phase de traduction est une phase de la synthèse des protéines durant laquelle un ribosome produit une protéine à partir d'un ARN messager.

transcription La phase de transcription est une phase de la synthèse des protéines durant laquelle un ARN polymérase lié au site promoteur d'un gène produit un ARN messager.