



MASTER 2 MFA

**Mémoire :**  
**Revisiting and extending probabilistic Boolean networks**



*Réalisé par :*  
Zachary ASSOUMANI

*Encadrante Université :*  
Cécile DARTYGE

*Encadrantes AMU :*  
Elisabeth REMY  
Claudine CHAOUIYA-CHANTEGREL

*Membres du jury :*  
Elisabeth REMY  
Claudine CHAOUIYA-CHANTEGREL  
Pedro MONTEIRO  
Guillaume THEYSSIER

Avril — Août 2023

L'Université de Lorraine n'entend donner ni approbation ni improbation aux opinions émises dans ce rapport, ces opinions devant être considérées comme propres à leur auteur.

## **Remerciements**

Ma reconnaissance va dans un premier temps à mes deux encadrantes, mesdames Claudine CHAOUIYA-CHANTEGREL et Elisabeth REMY, pour m'avoir aiguillé durant ce stage de recherche avec bienveillance et exigence, pour la confiance qu'elles m'ont accordé, mais aussi pour leur exigence qui m'a permis de parfaire mon travail.

Je remercie également messieurs Pedro MONTEIRO et Guillaume THEYSSIER pour leur revue de mon travail, ainsi que leurs questions et remarques pertinentes en tant que jury ; et merci une fois de plus à M. MONTEIRO pour ses ressources informatiques.

Salutations et bonne route aux stagiaires, doctorants, post-doctorants, chercheurs rencontrés à l'I2M : Laurent, José, Nadine, Raha, Jade, Annie, Piyush, Adrien, Laura, Maylis, Félix, M. Gaudillière... Merci pour leur chaleureux accueil et leur inspirante compagnie au cours de ces cinq mois.

Mes remerciements vont également à Mme Cécile DARTYGE, pour m'avoir aimablement accueilli dans sa formation de master ; ainsi qu'aux professeurs dont j'ai assisté aux cours durant cette dernière année à Nancy, dont le contenu des cours et les capacités pédagogiques m'ont grandement aidé à appréhender ce sujet de stage.

Mes sincères sentiments et mille vœux de bonheur aux amis, potes, camarades et autres copains dont le présent paragraphe est trop étroit pour contenir tous les noms, et sans qui toutes ces viles aventures en vaudraient un peu moins la peine. De Nancy, Marseille, Bruxelles, Paris ou d'autres mondes plus irréels ; à chacune de ces rencontres au bon endroit au bon moment, dont la circonstance fut hasardeuse mais féconde d'une ou quinze années de franche camaraderie.

Enfin, immanquablement, évidemment, naturellement, tout mon amour et ma gratitude vont à mes parents et à ma famille. Pour leur éducation ayant façonné les meilleurs aspects de la personne que je suis devenu, pour leur confiance et leur rigueur, et pour tous ces merveilleux moments passés en leur compagnie cette année et toutes celles d'avant.

La bise de loin à mon ami J-A.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 État de l'art</b>	<b>2</b>
1.1 Réseaux booléens de régulation biologique	2
1.1.1 Définitions	2
1.1.2 Propriétés dynamiques	4
1.1.3 Exemple de modèle biologique : différenciation des lymphocytes	5
1.2 Fonctions booléennes et ordre partiel	6
1.3 Chaînes de Markov	8
1.3.1 Définition	8
1.3.2 Comportement à long terme	9
1.3.3 Représentation d'un réseau booléen	10
1.4 Réseaux booléens probabilistes	11
1.4.1 Définition et déroulement d'une itération	11
1.4.2 Chaîne de Markov associée	12
1.4.3 Cas particulier : les PBN indépendants	12
1.4.4 Les attracteurs	13
1.4.5 Passage en revue des classes de modèles	15
1.5 Méthodes d'analyse des BNs et PBNs	16
1.6 Outils logiciels	18
<b>2 Contributions</b>	<b>19</b>
2.1 Interprétation de la loi stationnaire empirique	19
2.2 PBNs asynchrones	25
2.3 Génération de BNs synthétiques	25
2.3.1 Régulations strictement activatrices ou inhibitrices	25
2.3.2 Régulations duales autorisées	26
2.4 Génération de PBNs	27
2.4.1 Définition à l'aide des fonctions voisines	27
2.4.2 Définition à l'aide des tables de vérité aléatoires	28
2.5 Études de cas	28
2.5.1 Modèles synthétiques	28
2.5.2 Modèle biologique : TH <sub>23</sub>	32
<b>Conclusion</b>	<b>40</b>

<b>Bibliographie</b>	<b>43</b>
<b>Déclaration contre le plagiat</b>	<b>44</b>
<b>A Preuve de la formule en section 1.4.2</b>	<b>45</b>
<b>B Irréductibilité de la chaîne de Markov pour un PBN tel que <math>p&gt;0</math></b>	<b>47</b>
B.1 Termes nuls de la matrice de transition . . . . .	47
B.2 Irréductibilité de la chaîne . . . . .	47
<b>C Expression analytique de la probabilité stationnaire d'un attracteur</b>	<b>48</b>
C.1 Pour un BN perturbé synchrone . . . . .	48
C.2 Pour un PBN perturbé synchrone . . . . .	49
<b>D Documentation de l'outil informatique développé</b>	<b>50</b>
D.1 Attributs de la classe PBN . . . . .	50
D.2 Méthodes . . . . .	51
D.3 Générateurs . . . . .	54

# Introduction

En biologie des systèmes, l'étude et la modélisation des phénomènes biologiques à l'aide d'outils et de techniques mathématiques sont des approches privilégiées. Les *niveaux d'expression* des gènes dans une cellule étant dépendants les uns des autres, il convient de les représenter dans un formalisme représentant ces mécanismes de régulation. Une des classes de modèles employées est celles des réseaux booléens [1], aussi appelés "BN" pour *Boolean Networks*. L'expression d'un gène peut prendre deux niveaux (0 ou 1), et sa valeur suivante est liée par une fonction logique aux niveaux d'expression des autres gènes du modèle. Selon les fonctions employées, un réseau booléen permet de rendre compte des mécanismes d'activation et/ou d'inhibition entre les gènes.

Afin de figurer au mieux les processus observés en biologie, cette classe de modèles connaît plusieurs variantes : concernant le nombre et le choix des *fonctions de régulation*, les types de mise à jour à partir des fonctions, l'inclusion de perturbations stochastiques [2, 3, 4], ...

La première extension décrit la classe des réseaux booléens probabilistes (ou "PBN" pour *Probabilistic Boolean Network*). Les fonctions de régulation sont sélectionnées au hasard dans un ensemble fixé, reflétant l'incertitude dans le choix de fonction de régulation d'un gène [5, 6].

Les états stables d'un réseau sont appelés "*attracteurs*", et ils s'interprètent biologiquement comme le comportement des gènes à long terme : différenciation vers un autre type, mort cellulaire... Ainsi l'estimation des attracteurs est une question récurrente dans l'analyse des BN comme des PBN, objets que l'on peut représenter comme des chaînes de Markov sur un espace d'états fini [3, 7, 8].

Dans un PBN, le choix des fonctions pour un gène peut être restreint afin de correspondre à un mécanisme de régulation donné. Par exemple, si un gène  $g$  régule positivement un gène  $g'$ , la présence de  $g$  plutôt que son absence tendra à activer  $g'$ . De telles fonctions sont dites "*consistantes*" avec une régulation, et peuvent être ordonnées dans un PO-Set [9]. Il convient de s'interroger sur les effets de leur sélection sur la dynamique du réseau, en particulier la modification et l'atteignabilité des attracteurs.

La seconde extension concerne la méthode de calcul des successeurs d'un état. En mise à jour *synchrone*, tous les gènes sont actualisés simultanément; en mise à jour *asynchrone*, ils le sont un par un. La stratégie optée a des implications sur les propriétés dynamiques du réseau. Parmi les deux, le cas asynchrone est reconnu comme représentant le plus fidèlement les processus biologiques [10, 11, 12], a été étudié avec la classe de modèles BN, mais assez peu en ce qui concerne les PBN.

Le premier chapitre de ce mémoire est un passage en revue des connaissances actuelles sur les réseaux booléens : la classe de modèles et ses extensions, sa dynamique, ses outils d'analyse. Ensuite, le second chapitre présentera mes diverses contributions à ce domaine d'étude : un résultat de convergence d'un algorithme pour estimer les attracteurs, des méthodes de construction de modèles booléens et de PBN employant les fonctions consistantes, et deux études de cas interprétant l'impact du mode de mise à jour et du choix de fonctions sur le comportement à long terme du réseau booléen.

# Chapitre 1

## État de l'art

### 1.1 Réseaux booléens de régulation biologique

#### 1.1.1 Définitions

Dans la grande entreprise de compréhension du vivant, les processus de vie et de mort des cellules sont parmi ceux dont la modélisation mathématique contribue activement à leur étude. Dans un réseau de gènes, l'évolution de chaque composante est définie par une "**fonction de régulation**", mettant à jour le niveau d'activité du gène en fonction de l'état des autres composantes. La dynamique globale du système s'interprète alors en termes de processus biologiques.

Le travail de représentation des réseaux cellulaires par des modèles logiques est ainsi complété par l'étude théorique de ces modèles. Selon le choix de modélisation, différentes propriétés dynamiques peuvent émerger ; ainsi, l'étude de ces propriétés aide à l'approfondissement des connaissances sur les processus cellulaires initialement étudiés [13].

Considérons un ensemble de gènes, indicés  $\mathcal{V} = \{g_1 \dots g_n\}$ . L'ensemble de leurs interactions, souvent déduites d'expérimentations pratiques, est représentée par un **graphe de régulation** orienté signé  $(\mathcal{V}, \mathcal{E})$ . L'action d'un gène  $g_i$  sur l'état d'activation d'un gène  $g_j$ , sans que la nature biologique de l'action ne soit ici précisée, se traduit par  $(g_i, g_j) \in \mathcal{E}$ .

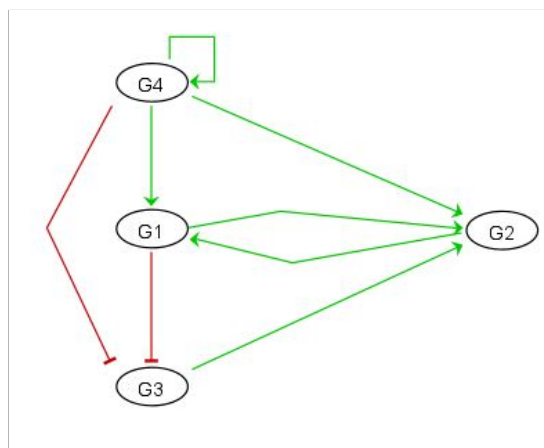


FIGURE 1.1 – Exemple de graphe de régulation à 4 nœuds [14].

On définit  $G(\mathcal{V}, f)$  un **réseau booléen** (ou "BN" pour *Boolean Network* en anglais) par :

- un ensemble de gènes  $\mathcal{V} = \{g_1 \dots g_n\}$ , dont les états d'activation sont contenus dans le vecteur d'activité  $x = (x_1, \dots, x_n) \in \{0, 1\}^n$
- une liste de fonctions  $f = (f_1, \dots, f_n)$ ,  $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$

Pour tout  $1 \leq i \leq n$ ,  $x_i$  correspond à l'état d'activation du gène  $g_i$  : 0 s'il est désactivé, 1 s'il est activé. La fonction  $f_i$  est appelée **fonction de régulation** du gène  $g_i$ , et prend en argument les valeurs des autres composantes de  $x$ . Ensemble, les  $f_i$  permettent de déduire le graphe de régulation, et de prédire le comportement du modèle.

$$\begin{aligned} f_1(x) &= \begin{cases} 1 & \text{si } (x_2 = 1) \vee (x_4 = 1) \\ 0 & \text{sinon} \end{cases} \\ f_2(x) &= \begin{cases} 1 & \text{si } (x_1 = 1 \wedge x_4 = 1) \vee (x_3 = 1) \\ 0 & \text{sinon} \end{cases} \\ f_3(x) &= \begin{cases} 1 & \text{si } (x_1 = 0 \wedge x_4 = 0) \\ 0 & \text{sinon} \end{cases} \\ f_4(x) &= x_4 \end{aligned}$$

FIGURE 1.2 – Fonctions de régulation adéquates avec l'exemple du graphe 1.1.

L'ensemble des **variables régulatrices** du gène  $g_j$  est noté  $Reg_j$ , ce sont les gènes dont la valeur peut influencer sur la sortie de la fonction  $f_j$ . Plus formellement :

$$i \in Reg_j \iff \exists x, f_j(x) \neq f_j(\bar{x}^i), \quad \text{avec } \bar{x}^i = (x_1, \dots, x_{i-1}, \neg x_i, x_{i+1}, \dots, x_n)$$

On dit aussi que  $g_i$  est un *régulateur fonctionnel* de  $g_j$ , ou que  $x_i$  est une *variable essentielle* de  $f_j$ .

Dans le graphe de régulation, elles correspondent aux voisins entrants de  $g_j$ . Le graphe est dit *signé*, c'est-à-dire qu'à chaque arête est associé un signe +/- selon si elle représente une relation d'activation ou d'inhibition. Ainsi pour un gène  $g_j$ , on partitionnera ses variables régulatrices en :

- variables activatrices :  $Reg_j^+ = \{g_i \in Reg_j ; \forall x, f_j(\bar{x}^{i \leftarrow 0}) \leq f_j(\bar{x}^{i \leftarrow 1})\}$  (=  $f_j$  est *positive* en  $x_i$ )
- variables inhibitrices :  $Reg_j^- = \{g_i \in Reg_j ; \forall x, f_j(\bar{x}^{i \leftarrow 0}) \geq f_j(\bar{x}^{i \leftarrow 1})\}$  (=  $f_j$  est *négative* en  $x_i$ )
- variables duales :  $Reg_j^{+-} = Reg_j \setminus (Reg_j^+ \cup Reg_j^-)$

avec  $f_j(\bar{x}^{i \leftarrow b}) = (x_1, \dots, x_{i-1}, b, x_{i+1}, \dots, x_n)$ .

Sur la figure 1.1, toutes les relations activatrices sont tracées en vert, et toutes les relations inhibitrices en rouge.

À un graphe de régulation peuvent correspondre différents choix de fonctions de régulation. Dans l'exemple 1.2, le gène  $g_1$ , de fonction de régulation  $f_1(x) = x_2 \vee x_4$ , possède 2 régulateurs activateurs :  $g_2$  et  $g_4$ . Mais si la fonction était  $f_1(x) = x_2 \wedge x_4$ ,  $g_1$  serait toujours activé par  $g_2$  et  $g_4$ , et le graphe de régulation serait toujours celui de 1.1.

### 1.1.2 Propriétés dynamiques

À un état donné  $x$  du réseau booléen, on peut définir ses possibles états suivants à partir d'un **opérateur de succession**, noté  $Succ : \{0, 1\}^n \rightarrow 2^{\{0, 1\}^n}$ .

Dans le cas dit **synchrone**, toutes les variables sont mises à jour *simultanément*. L'état  $x$  n'a qu'un état successeur, celui obtenu en substituant à chaque composante la sortie de sa fonction de régulation :

$$Succ(x) = \{f(x)\} := \{(f_1(x), \dots, f_n(x))\}$$

Dans le cas dit **asynchrone**, chacune des  $n$  variables peut être mise à jour *séparément*. Selon la variable sélectionnée pour être mise à jour, on a différents états successeurs possibles :

$$Succ(x) = \begin{cases} \{(x_1, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_n); i \in \llbracket 1, n \rrbracket, f_i(x) \neq x_i\} & \text{si } f(x) \neq x \\ \{x\} & \text{sinon} \end{cases}$$

Chaque état  $x$  est garanti d'avoir au moins un successeur : l'ensemble des états (s'il en existe) ne lui diffère que d'une transition sur une variable, ou lui-même. L'aléa dans l'ordre d'actualisation rend l'évolution non-déterministe, puisqu'il existe au plus  $n$  états successeurs à tout état. Le cas asynchrone représente plus fidèlement la réalité en rendant compte de la variabilité dans les durées des processus biologiques [10], mais son absence de déterminisme complexifie l'analyse de ses propriétés dynamiques.

Les possibles évolutions du système, étant donnés une liste de fonctions de régulation et un choix de mise à jour synchrone ou asynchrone, sont représentées par un **graphe de transition d'états**  $(\mathcal{V}', \mathcal{E}')$ , aussi appelé "STG" pour *State Transition Graph* :

- $\mathcal{V}' = \{0, 1\}^n$
- $\mathcal{E}' = \{(x_1, x_2) \in \mathcal{V}'^2 ; x_2 \in Succ(x_1)\}$

Le nombre d'états ( $= 2^n$ ) étant fini, certains seront visités une infinité de fois par la trajectoire du système. Il convient alors de s'intéresser aux comportements asymptotiques.

On définit  $x \in \{0, 1\}^n$  comme **état stable** s'il ne peut plus évoluer, c'est-à-dire si  $Succ(x) = \{x\}$ .

Plus généralement, un **attracteur** est une composante fortement connexe<sup>1</sup>  $A \subset \{0, 1\}^n$  du graphe de transition d'états, vérifiant :

$$\forall x \in A, \bigcup_{k \geq 1} Succ^k(x) = A$$

où l'on définit récursivement l'itérée de l'opérateur  $Succ$  :

$$Succ^1 := Succ, \quad \forall k \geq 1, \forall x \in \{0, 1\}^n, Succ^{k+1}(x) = \bigcup_{y \in Succ^k(x)} Succ(y)$$

Enfin, pour un attracteur  $A$  donné, son **bassin d'attraction** (*strict*), noté  $B_A(\bar{B}_A)$ , est l'ensemble des états dont certaines (*toutes les*) trajectoires mènent à cet attracteur [15, sect. 3C].

$$\begin{aligned} B_A &= \{x \in \{0, 1\}^n ; \exists k \geq 1, Succ^k(x) \cap A \neq \emptyset\} \\ \bar{B}_A &= \{x \in \{0, 1\}^n ; \forall k \geq 1, \forall A' \neq A \text{ attracteur}, Succ^k(x) \cap A' = \emptyset\} \\ &= \{x \in \{0, 1\}^n ; \forall A' \neq A \text{ attracteur}, x \notin B_{A'}\} \end{aligned}$$

1. Pour toute paire d'états appartenant à  $A$ , ces états sont mutuellement atteignables dans le STG.  $A$  est maximal pour cette propriété, c'est-à-dire que tout ajout d'état supplémentaire ne la ferait plus respecter.



*Remarque : les attracteurs sont les sous-ensembles minimaux stables par l'opérateur Succ.*

*Remarque : une liste de fonctions de régulation définit son graphe de régulation et son graphe de transition d'états.*

*Remarque : pour tout attracteur  $A$ ,  $\bar{B}_A \subseteq B_A$ .*

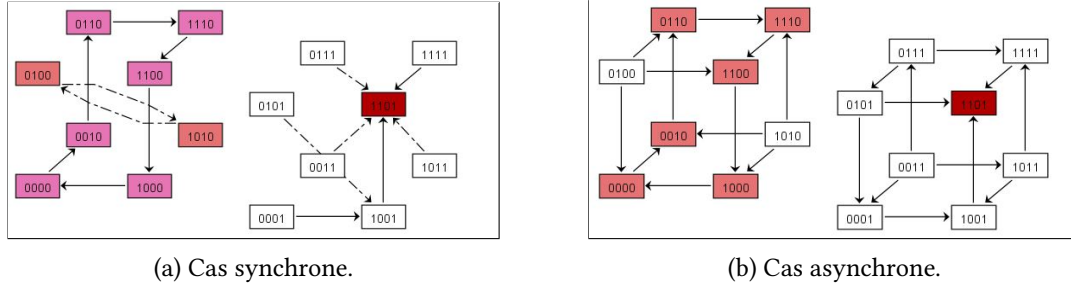


FIGURE 1.3 – Graphes de transition d'états, les attracteurs y sont surlignés.  
L'état stable 1101 possède une autoboucle, non-indiquée sur les figures.

Intéressons-nous à cet exemple en figure 1.3 à  $n = 4$  variables, déduit des fonctions de régulation de la figure 1.2. Chaque noeud du STG est noté comme un mot binaire à 4 caractères — par exemple, '0101' est l'état où  $x_2$  et  $x_4$  sont activés, mais pas  $x_1$  et  $x_3$ . Dans le cas synchrone, on constate l'existence d'un attracteur de taille 2  $\{0100, 1010\}$ , un attracteur de taille 6  $\{0000, 0010, 0110, 1110, 1100, 1000\}$ , et d'un état stable  $\{1101\}$  dont le bassin d'attraction est tous les états pour lesquels  $x_4 = 1$ . Pour le même réseau asynchrone, on remarque le même attracteur de taille 6 et l'état stable, bien que les arêtes du STG soient différentes.

### 1.1.3 Exemple de modèle biologique : différenciation des lymphocytes

Lorsque l'on représente des comportements biologiques par un BN, l'existence et l'atteignabilité des attracteurs sont des propriétés dynamiques de première importance. Les mécanismes biologiques de la cellule (mort cellulaire, différenciation...) dépendent de son comportement asymptotique.

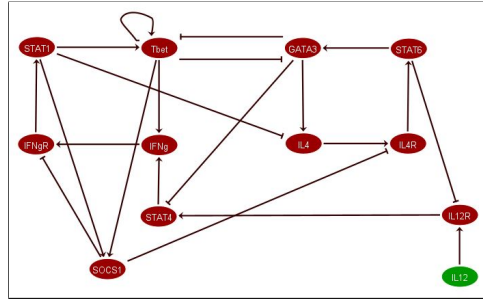
Parmi les cellules du système immunitaire, les lymphocytes T auxiliaires (ou 'T-helper') peuvent se différencier en deux phénotypes : T-helper 1 (dit *Th1*, chargé de l'activation d'autres cellules) ou T-helper 2 (dit *Th2*, ayant un rôle dans la production d'anticorps) [16, sect. 4.1].

Le réseau THBOOLEAN [16] modélise les régulations entre les molécules et macromolécules sécrétées par les lymphocytes T auxiliaires, participant à la différenciation de ces dernières. Présenté en figures 1.4, il est composé de douze variables booléennes (facteurs de transcription, récepteurs, autres protéines...), dont les relations d'activation et d'inhibition sont représentées en figure 1.4a. Dans les graphes de transition figs. 1.4b et 1.4c, les  $2^{12} = 4096$  états-nœuds sont agrégés par bassins d'attraction. La mise à jour asynchrone engendre trois états stables, marqués en rouge vif :

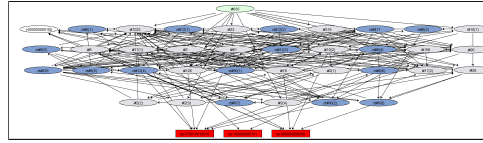
- '000000000000' (aucun gène activé)
- '010010010010' (activation de  $[IL4, IL4R, STAT6, GATA3]$ )
- '100100100001' (activation de  $[IFN\gamma, SOCS1, Tbet]$ )

Le cas synchrone contient ces trois états stables, mais également trois attracteurs en rouge pâle de taille 4, et un attracteur de taille 2.

L'interprétation biologique faite dans l'article [16] est la suivante : ces trois états stables correspondent respectivement à la *différenciation* vers un lymphocyte naïf, de type Th2, et de type Th1. Pour chacun de ces types de cellules, les gènes activés dans l'état stable correspondant en sont les *marqueurs*.



(a) Graphe de régulation à 12 nœuds.



(b) Graphe des transitions, asynchrone.



(c) Graphe des transitions, synchrone.

FIGURE 1.4 – Modèle THBOOLEAN, simulant la différenciation de lymphocytes T auxiliaires.

## 1.2 Fonctions booléennes et ordre partiel

Dans le graphe de régulation d'un réseau booléen, le noeud associé au gène  $g_i$  compte  $p := |Reg_i|$  voisins entrants. Une fonction de régulation pour  $g_i$ , que l'on peut considérer de  $\{0, 1\}^p$  dans  $\{0, 1\}$ , est **monotone** si elle est positive ou négative en chacune de ses variables. Elle est dite **non-dégénérée** si toutes ses variables lui sont essentielles.

*Exemples :*  $f(x_1, x_2, x_3) = x_1 \vee (x_2 \wedge x_3)$  est monotone, mais  $f(x_1, x_2, x_3) = (x_1 \wedge \neg x_3) \vee (x_2 \wedge x_3)$  ne l'est pas.  $f(x_1, x_2) = x_1 \wedge x_2$  est non-dégénérée, mais  $f(x_1, x_2) = x_1$  est dégénérée.

$\mathcal{F}_i$  est l'ensemble des fonctions de régulation **consistantes** avec le signe des régulateurs de  $g_i$ . Ce sont les fonctions de  $\{0, 1\}^p \leftarrow \{0, 1\}$  monotones non-dégénérées, positives (resp. négatives) en chaque activateur (resp. inhibiteur) de  $g_i$  [9].

Pour toute fonction booléenne sur  $\{0, 1\}^p$ , on note la liste de ses états acceptés  $\mathbb{T}(f) = \{x \in \{0, 1\}^p \mid f(x) = 1\}$ . On structure alors  $\mathcal{F}_i$  en un ensemble ordonné par la relation suivante :

$$\forall f, f' \in \mathcal{F}_i, \quad f \preceq f' \iff \mathbb{T}(f) \subseteq \mathbb{T}(f')$$

$(\mathcal{F}_i, \preceq)$  est un PO-Set<sup>2</sup>, que l'on peut représenter sous la forme d'un diagramme de Hasse (cf. figure 1.5).

Toute fonction de  $\mathcal{F}_i$  est exprimable sous une forme normale disjonctive, appelée *forme normale disjonctive complète* [17], sur les  $p$  littéraux  $u_1 \dots u_p$ , avec  $u_k = x_k$  (resp.  $u_k = \neg x_k$ ) si  $g_k$  active (resp. inhibe)  $g_i$ . On l'écrit ainsi :

2. Partially-Ordered Set.

$$f = \bigvee_{j=1}^m C_j, \quad \text{avec } C_j = \bigwedge_{k \in E_j} u_k$$

avec  $E_j \subseteq \{1, \dots, p\}$  les indices des littéraux apparaissant dans la clause  $C_j$ .

Dans un ensemble ordonné  $(P, \leq)$ , une *antichaine* de  $P$  est un sous-ensemble  $A \subseteq P$ , telle que pour toute paire  $x, y \in A$ ,  $x$  et  $y$  sont incomparables — c'est-à-dire que ni  $x \leq y$  ni  $y \leq x$  ne sont vérifiés. Soit à présent  $B \subseteq 2^S$ , ses éléments sont des parties de  $S$ . On dira que  $B$  *couvre* l'ensemble  $S$  si  $\bigcup_{X \in B} X = S$ .

*Exemples :*  $\{\{1, 2\}, \{1, 4\}\}$  est une antichaine de  $(2^{\{1,2,3,4\}}, \subseteq)$ , mais  $\{\{1\}, \{1, 4\}\}$  n'est pas une antichaine.  $\{\{1, 2\}, \{1, 3, 4\}\}$  couvre  $\{1, 2, 3, 4\}$ , mais  $\{\{1, 2\}, \{1, 3\}\}$  ne le couvre pas.

Il existe alors une bijection entre  $(\mathcal{F}_i, \preceq)$  et l'ensemble des antichaines couvrantes de  $(2^{\{1 \dots p\}}, \subseteq)$  :

$$\begin{aligned} \mathcal{F}_i &\longleftrightarrow 2^{2^{\{1 \dots p\}}} \\ \bigvee_{j=1}^m \bigwedge_{k \in E_j} u_k &\longleftrightarrow \{E_j\}_{j=1}^m \end{aligned}$$

Dans le PO-Set de fonctions consistantes de  $g_i$ ,  $f'$  est voisin direct de  $f$  si l'on se trouve dans un des deux cas suivants :

- $f < f'$ , et  $\nexists f'' \in \mathcal{F}_i$ ;  $f < f'' < f'$  ( $f'$  est un *parent* de  $f$ )
- $f' < f$ , et  $\nexists f'' \in \mathcal{F}_i$ ;  $f' < f'' < f$  ( $f'$  est un *enfant* de  $f$ ).

Lorsque l'on cherche à étudier la dynamique de fonctions de régulation proches de la fonction de référence d'un gène, l'intérêt des voisins dans le PO-Set des fonctions consistantes est double : il conserve le graphe de régulation, et ne modifie pas radicalement l'ensemble des états acceptés.

L'article [9] présente un algorithme explorant localement ces voisins, faisant appel de la représentation en antichaine de la fonction booléenne. Cette méthode d'exploration sera utilisée en section 2.4.1 pour définir une certaine classe de réseaux booléens non-déterministes.

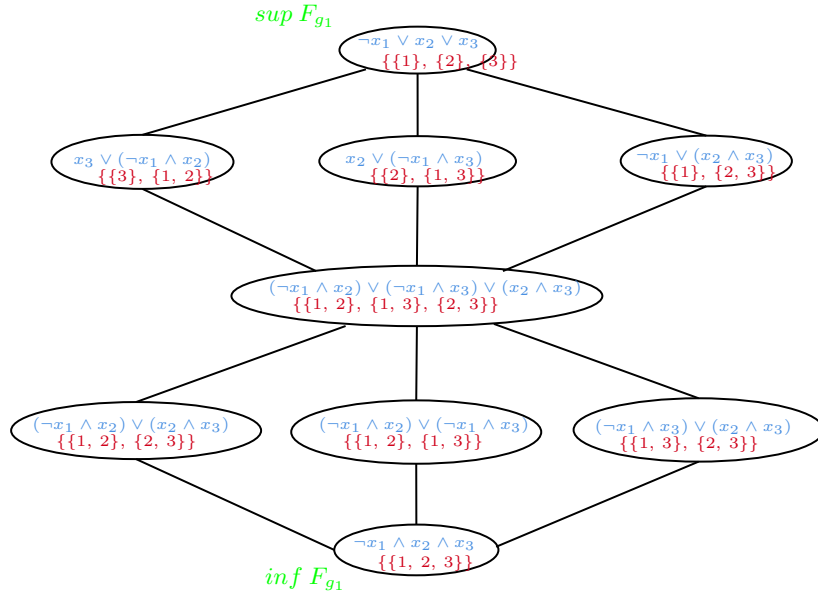


FIGURE 1.5 – Exemple de diagramme de Hasse du PO-Set pour un gène dont les 3 régulateurs sont  $x_1$  inhibiteur,  $x_2$  et  $x_3$  activateurs. Les fonctions sont en bleu, les antichaînes correspondantes en rouge.

## 1.3 Chaînes de Markov

### 1.3.1 Définition

Une **chaîne de Markov**  $(X_i)_{i \in \mathbb{N}}$  est un processus stochastique à temps discret, vérifiant la propriété suivante appelée **propriété de Markov** :

$$\forall n \in \mathbb{N}, \mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

avec  $x_0, \dots, x_{n+1}$  appartenant à  $E$  l'espace des états de la chaîne. La distribution de l'état futur, conditionnée aux états passés et présents, ne dépend que de l'état présent.

On étudiera ici des chaînes de Markov *homogènes*, c'est-à-dire dont les probabilités de transition sont indépendantes de  $n$ . La **matrice de transition**  $P$  d'une chaîne de Markov définit ces probabilités comme suit :

$$\forall n \in \mathbb{N}, \forall i, j \in E, P_{i,j} = \mathbb{P}(X_{n+1} = j | X_n = i)$$

Le **graphe associé** à la chaîne de Markov est le graphe orienté pondéré dont la matrice d'adjacence est la matrice de transition susdéfinie. Ses sommets sont les éléments de  $E$ , et ses arêtes sont les couples  $(i, j)$  tels que  $P_{ij} > 0$ , pondérées par les  $P_{ij}$ .

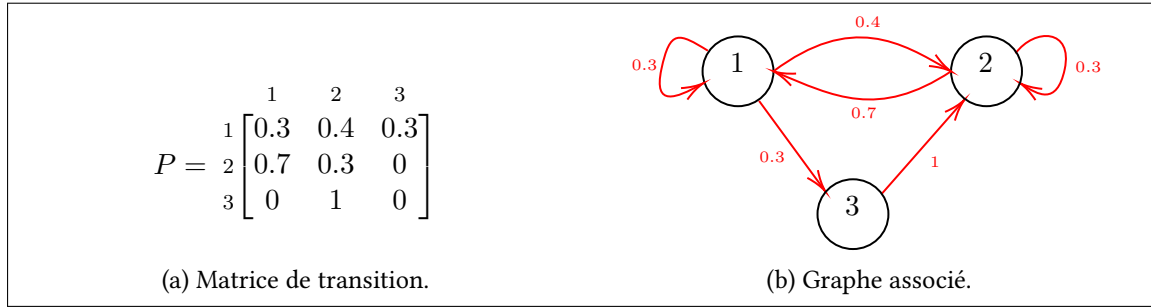


FIGURE 1.6 – Exemple de chaîne de Markov à 3 états.

La matrice de transition est dite *stochastique* :  $\forall i, \sum_j P_{i,j} = 1$ .

Ses puissances successives prédisent la transition après  $k$  étapes :  $(P^k)_{i,j} = \mathbb{P}(X_{n+k} = j | X_n = i)$ .

### 1.3.2 Comportement à long terme

Deux états  $i$  et  $j$  sont dits *communicants* s'ils sont mutuellement accessibles l'un à l'autre avec une probabilité non-nulle :  $\exists k, k' \geq 1, (P^k)_{ij} > 0$  et  $(P^{k'})_{ji} > 0$ . La relation de communication est une relation d'équivalence sur  $E$ , dont les classes sont appelées **classes communicantes**. Celles-ci correspondent aux composantes fortement connexes du graphe associé. Notons qu'en réannotant les états, on peut réarranger  $P$  en une matrice diagonale par blocs dont les blocs sont les classes communicantes. La chaîne est dite **irréductible** si elle ne possède qu'une classe communicante, c'est-à-dire si tous les états de  $E$  sont accessibles l'un l'autre.

L'état  $i$  est de période  $k$  si toute transition de  $i$  vers lui-même doit s'accomplir en un multiple de  $k$  étapes. La périodicité étant invariante au sein d'une classe communicante, on peut parler de classe **apériodique** si ses états sont de période 1.

Un état  $i \in E$  est dit **récurrent** si le nombre de passages de la chaîne en  $i$  est infini presque sûrement, c'est-à-dire :  $\mathbb{P}(\sum_{k \leq 0} \mathbb{1}_{X_k=i} = +\infty) = 1$ . Dans une chaîne irréductible à espace d'états fini, tous les états sont récurrents.

Une chaîne de Markov homogène finie possède une **loi stationnaire**  $\pi = (\pi_1, \dots, \pi_n)$  sur  $E$  si  $\pi P = \pi$  et  $\sum_{i \in E} \pi_i = 1$ , c'est-à-dire si  $\pi$  est un vecteur propre normalisé de  $P$  associé à la valeur propre 1.  $\pi_i$  s'interprète comme la proportion de temps passé à l'état  $i$  à long terme. Si la chaîne est irréductible et apériodique, elle est dite **ergodique** et possède une unique loi stationnaire, vérifiant :  $\lim_{k \rightarrow \infty} (P^k)_{ij} = \pi_j$ . [18].

$$\lim_{k \rightarrow \infty} P^k = \begin{bmatrix} 10/23 & 10/23 & 3/23 \\ 10/23 & 10/23 & 3/23 \\ 10/23 & 10/23 & 3/23 \end{bmatrix}$$

FIGURE 1.7 – Puissance limite de la matrice de la chaîne de Markov ergodique présentée en fig. 1.6. La loi stationnaire est  $[10/23, 10/23, 3/23]$ .

### 1.3.3 Représentation d'un réseau booléen

Les chaînes de Markov ainsi définies peuvent se révéler utiles pour étudier les réseaux booléens. Le **STG** d'un réseau booléen (défini en section 1.1.2) peut être vu comme le **graphe d'une chaîne de Markov**, dont les états sont ceux du STG, et la matrice de transition  $P$  est telle que  $y \notin Succ(x) \Rightarrow P_{x,y} = 0$ .

Pour un réseau à mise à jour synchrone, puisque  $\forall x Succ(x)$  ne contient qu'un élément, la matrice  $P$  est nécessairement binaire. Quant aux cas où  $\#Succ(x) \geq 2$  en mise à jour asynchrone, il convient de faire un choix pour les probabilités  $P_{x,y}$ . Toutes les transitions parmi les successeurs sont généralement assumées équiprobables [12], la matrice de transition de la chaîne de Markov vaut alors :

$$\forall x, y \in \{0, 1\}^n, P_{x,y} = \frac{1}{\#Succ(x)} \mathbb{1}_{Succ(x)}(y)$$

Revenons à l'exemple de réseau booléen à  $n = 4$  gènes présenté en figures 1.1-1.2-1.3. Voici les matrices de transition de leurs chaînes de Markov, avec la partition délimitée entre  $\{x_4 = 0\}$  et  $\{x_4 = 1\}$  :

	0000	0010	0100	0110	1000	1010	1100	1110	0001	0011	0101	0111	1001	1011	1101	1111
$P_{sync}$	0000	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
	0010	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.
	0100	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
	0110	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.
	1000	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1010	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.
	1100	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
	1110	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.
	0001	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.
	0011	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	0101	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.
	0111	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1001	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1011	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1101	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1111	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.

(a) Dans le cas synchrone.

	0000	0010	0100	0110	1000	1010	1100	1110	0001	0011	0101	0111	1001	1011	1101	1111
$P_{async}$	0000	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
	0010	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.
	0100	1/3	.	.	1/3	.	.	1/3	.	.	.	.	.	.	.	.
	0110	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.
	1000	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1010	.	1/3	.	.	1/3	.	1/3	.	.	.	.	.	.	.	.
	1100	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
	1110	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.
	0001	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.
	0011	.	.	.	.	.	.	.	1/3	.	.	1/3	.	1/3	.	.
	0101	.	.	.	.	.	.	.	1/2	.	.	.	.	.	1/2	.
	0111	.	.	.	.	.	.	.	.	.	1/2	.	.	.	.	1/2
	1001	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1011	.	.	.	.	.	.	.	.	.	.	.	1/2	.	.	1/2
	1101	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.
	1111	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.

(b) Dans le cas asynchrone.

FIGURE 1.8 – Matrices de transition des chaînes de Markov associées à l'exemple des figures 1.1-1.2-1.3.

Les figures 1.3 nous indiquent l'existence d'au moins deux classes communicantes dans le cas synchrone comme asynchrone, ici bien délimitées par la forme "diagonale par blocs" des matrices : seuls

les blocs sur la diagonale principale (supérieur-gauche et inférieur-droit) sont non-nuls. La chaîne de Markov n'est pas irréductible, donc la loi stationnaire n'est pas unique.

En notant  $\delta_a$  la mesure de Dirac au point  $a$ , on pourra vérifier que les distributions de probabilité :

- $\pi^1 = \delta_{1101}$
- $\pi^2 = \frac{1}{6}(\delta_{0000} + \delta_{0010} + \delta_{0110} + \delta_{1110} + \delta_{1100} + \delta_{1000})$
- ou toute combinaison  $\pi = t\pi^1 + (1-t)\pi^2$  avec  $t \in [0, 1]$

sont des lois stationnaires pour le cas synchrone comme asynchrone.

## 1.4 Réseaux booléens probabilistes

Dans le cas d'un BN synchrone, la dynamique est déterministe : chaque état n'a qu'un seul successeur possible, et il n'existe qu'une seule liste de fonctions de régulation.

Il peut arriver que le mécanisme de régulation biologique puisse être descriptible non par un, mais plusieurs ensembles de fonctions de régulation. Tout en conservant sa synchronicité, il est envisageable de changer aléatoirement de fonctions de régulation au cours de la simulation du réseau [6].

### 1.4.1 Définition et déroulement d'une itération

Un **réseau booléen probabiliste** (ou "PBN" pour *Probabilistic Boolean Network* en anglais)  $A^{p,q}(\mathcal{V}, F, C)$  est le processus stochastique défini par les composantes suivantes :

- un ensemble de gènes  $\mathcal{V} = \{g_1 \dots g_n\}$ , dont les états d'activation sont contenus dans le vecteur d'activité  $x = (x_1, \dots, x_n) \in \{0, 1\}^n$
- les contextes (=listes de fonctions)  $F = (f^{(1)}, \dots, f^{(m)})$
- la liste des poids de chaque contexte,  $C = (c_1, \dots, c_m)$ , telle que  $\sum_{l=1}^m c_l = 1$
- le facteur de changement de contexte  $q \in [0, 1]$
- et le facteur de perturbation  $p \in [0, 1]$ .

Une itération se déroule en plusieurs étapes :

#### 1) Contexte

Chaque contexte  $f^{(l)} \in F$  ( $0 \leq l \leq m$ ) est la liste de fonctions de régulation d'un BN comme définie précédemment. En notant  $f^{(l)} = (f_1^{(l)}, \dots, f_n^{(l)})$ , la fonction  $f_i^{(l)} : \{0, 1\}^n \rightarrow \{0, 1\}$  prédit le gène  $i$  lorsque le réseau booléen  $l$  est sélectionné.

Le PBN change ou non de contexte en tirant une variable  $\xi \sim \text{Bernouilli}(q)$  :

- si  $\xi = 0$ , on ne change pas de contexte,
- si  $\xi = 1$ , on tire un contexte suivant la probabilité discrète sur  $F : \sum_{l=1}^m c_l \delta_{f^{(l)}}$ , où  $\delta_a$  est la mesure de Dirac en  $a$ . Autrement dit  $\forall l, \mathbb{P}(f^{(l)} \text{ est sélectionné} \mid \xi = 1) = c_l$ .

## 2) Perturbation ou appel de la fonction [2]

On tire ensuite le vecteur de perturbation  $\gamma = (\gamma_i)_{1 \leq i \leq n}$  à valeurs dans  $\{0, 1\}^n : \forall i, \gamma_i \sim \text{Bernoulli}(p)$ .

- Si  $\gamma \neq 0$ , on met à jour les  $x_i$  perturbés :

$$\text{Succ}(x) = \{(x_1^{\gamma_1}, \dots, x_n^{\gamma_n})\}, \text{ avec } x_i^{\gamma_i} = \begin{cases} x_i, & \text{si } \gamma_i = 0 \\ 1 - x_i, & \text{sinon.} \end{cases}$$

- Si  $\gamma = 0$ , il n'y a pas de perturbation. L'état est mis à jour avec le contexte  $f$  sélectionné, comme dans un BN synchrone :

$$\text{Succ}(x) = \{f(x)\}$$

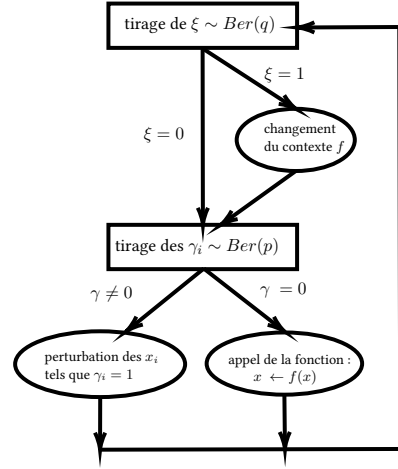


FIGURE 1.9 – Organigramme des étapes d'une itération pour un PBN.

### 1.4.2 Chaîne de Markov associée

Contrairement au BN classique, une configuration d'un PBN peut être représentée comme un couple (contexte, état), évoluant dans l'espace  $F \times \{0, 1\}^n$ . La dynamique d'un PBN peut être simulée par une chaîne de Markov sur cet espace [6].

Pour tous contextes  $\mathbf{f}_1, \mathbf{f}_2 \in F$  et vecteurs d'activité  $\mathbf{x}_1, \mathbf{x}_2 \in \{0, 1\}^n$  :

$$\mathbb{P}((\mathbf{f}_2, \mathbf{x}_2) | (\mathbf{f}_1, \mathbf{x}_1)) = [(1 - q)\mathbb{1}_{\mathbf{f}_1 = \mathbf{f}_2} + qc_2] \times [(1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1) = \mathbf{x}_2} + (1 - p)^{n - \eta(\mathbf{x}_1, \mathbf{x}_2)} p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2}]$$

où  $\eta(x, y)$  désigne la distance de Hamming entre  $x$  et  $y$ . Une preuve de cette formule est détaillée en annexe A.

La matrice de transition est alors de taille  $m2^n$ , avec  $m$  le nombre de contextes et  $n$  le nombre de gènes.

*Remarque : si  $p = q = 0$ , le comportement est identique à un BN classique. Si  $p > 0$  et  $q = 1$ , on dit que le PBN est "instantanément aléatoire".*

*Remarque : la probabilité, lors d'une itération, de faire appel à la fonction vaut  $\mathbb{P}(\xi = 0) = (1 - p)^n$ . Lorsque l'on simule un grand réseau ( $n \gg 1$ ), on devra choisir un  $p \ll 1/n$  pour ne pas se retrouver quasi-sûrement dans le cas d'une perturbation. On a alors  $\mathbb{P}(\xi \neq 0) \approx np$ .*

### 1.4.3 Cas particulier : les PBN indépendants

Un PBN est dit **indépendant** si les tirages des fonctions de régulation de chaque gène  $f_1, \dots, f_n$  sont indépendants [6].

C'est notamment le cas lorsque l'on associe à chaque gène  $g_i$  un ensemble  $F_i = \{f_{i,1}, \dots, f_{i,l(i)}\}$ , et que  $F = F_1 \times \dots \times F_n$ , c'est-à-dire : un contexte est une liste dont le  $i$ -ème élément est une fonction appartenant à  $F_i$ . Les tirages sont alors mutuellement indépendants, on pourrait qualifier cette sous-classe de '**PBN mutuellement indépendants**'. Mais dans la suite de ce mémoire, en particulier en section 2.4.2, j'emploierai le terme '**PBN indépendants**' pour la qualifier, par opposition au terme '**PBN non-indépendants**' qui désignera la classe des PBN en général.



On définit le PBN **indépendant**  $A'^{p,q}(\mathcal{V}, F', C')$  avec les composantes suivantes :

- un ensemble de gènes  $\mathcal{V} = \{g_1 \dots g_n\}$
- $F' = (F_1, \dots, F_n)$ , où chaque  $F_i = \{f_{i,1}, \dots, f_{i,m_i}\}$  est un ensemble de fonctions de régulation
- $C' = (c_1, \dots, c_n)$ , où chaque  $c_i = \{c_{i,1}, \dots, c_{i,m_i}\}$  est une liste de coefficients de probabilité, telle que  $\sum_{j=1}^{m_i} c_{i,j} = 1$

Son exécution se déroule comme celle d'un PBN, excepté lorsque l'on tire un contexte.

$$\forall 1 \leq i \leq n, \forall 1 \leq j \leq m_i,$$

$$\mathbb{P}(f_i = f_{i,j} \mid \xi = 1) = c_{i,j}$$

On constate qu'il peut effectivement être considéré comme un PBN au sens général, possédant  $N = \prod_{i=1}^n |F_i|$  contextes.  $A'^{p,q}(\mathcal{V}, F', C') = A^{p,q}(\mathcal{V}, F, C)$  si :

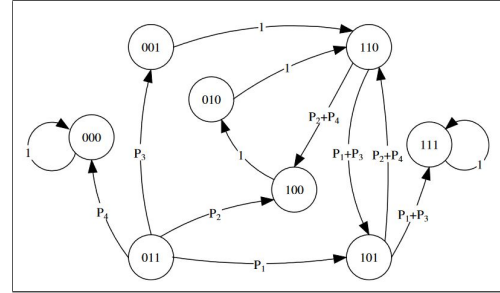
$$F = \prod_{F_i \in F'} F_i, \quad \text{et } C = (c_{j_1, \dots, j_n})_{1 \leq i \leq n, 1 \leq j_i \leq m_i}$$

$$\text{avec } c_{j_1, \dots, j_n} := \mathbb{P}\left(\bigcap_{i=1}^n f_i = f_{i,j_i} \mid \xi = 1\right) = \prod_{i=1}^n c_{i,j_i}$$

La définition originelle d'un PBN dans l'article [5] décrivait ce type de PBN à tirages mutuellement indépendants, il n'a été étendu à sa définition actuelle qu'au cours des articles suivants [3].

$x_1 x_2 x_3$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

(a) Choix de fonctions.



(b) STG.

FIGURE 1.10 – Exemple de PBN indépendant issu de [5].

$F_1 = \{f_1^{(1)}, f_2^{(1)}\}$ ,  $F_2 = \{f_1^{(2)}\}$ ,  $F_3 = \{f_1^{(3)}, f_2^{(3)}\}$ , il possède  $2 \times 1 \times 2 = 4$  contextes possibles.

Remarque : Le STG du PBN, comme vu en 1.10b, est obtenu en pondérant les STG des contextes par leurs poids :

$$\text{pour } w_f(x, y) = 1 \text{ ssi } y = f(x), \quad w_{PBN}(x, y) = \sum_{i=1}^N c_i \cdot w_{f_i}(x, y)$$

Remarque : il existe des PBN indépendants qui ne sont pas mutuellement indépendants. Avec  $f_1', f_1'', f_2', f_2'', f_3', f_3''$  des fonctions de régulation,  $F = ((f_1', f_2', f_3'), (f_1', f_2'', f_3'), (f_1'', f_2', f_3'), (f_1'', f_2'', f_3'))$ , et  $C = (1/4, 1/4, 1/4, 1/4)$ .

*Indépendant* :  $\mathbb{P}(f_i = f_i', f_j = f_j') = 1/4 = \mathbb{P}(f_i = f_i') \mathbb{P}(f_j = f_j') \forall i \neq j$ .

*Non-mutuellement indépendant* :  $\mathbb{P}(f_1 = f_1') \mathbb{P}(f_2 = f_2') \mathbb{P}(f_3 = f_3') = 1/8$ , mais  $\mathbb{P}(f_1 = f_1', f_2 = f_2', f_3 = f_3') = 0$ .

#### 1.4.4 Les attracteurs

Bien que la définition originelle d'un PBN [5] corresponde au cas  $q = 1$  (le contexte est modifié à chaque tour), un article plus récent des mêmes auteurs [3, pp. 1996-1997] justifie de fixer  $q \ll 1$  afin

que le contexte soit inchangé sur de longues périodes de temps, car un changement de contexte n'a biologiquement lieu que lorsqu'un stimulus extérieur vient le causer. Ainsi, tant que le contexte n'est pas modifié, le PBN se comporte comme un BN avec perturbation. En suivant ces indications, on se retrouve avec les inégalités suivantes entre les ordres de grandeur :

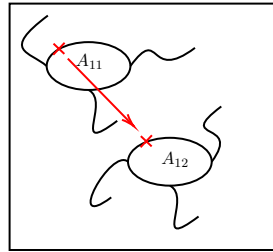
$$\underbrace{q}_{\text{proba}^\circ \text{ d'un changement de contexte}} \ll \underbrace{np}_{\text{proba}^\circ \text{ d'une perturbation}} \ll 1$$

Si le facteur de perturbation  $p$  est strictement positif, tous les termes de la matrice de transition sont non-nuls (exceptées les transitions  $(f_1, \mathbf{x}) \rightarrow (f_2, \mathbf{x})$  tels que  $f_2(\mathbf{x}) \neq \mathbf{x}$ ), donc la chaîne est irréductible. Les preuves de ces deux affirmations sont détaillées en annexe B. De ce fait, si l'on transpose la définition précédente d'un attracteur à l'espace d'états  $F \times \{0, 1\}^n$ , le seul attracteur sera  $F \times \{0, 1\}^n$  tout entier. On définit donc un attracteur dans un PBN comme relatif à un de ses contextes [3] :

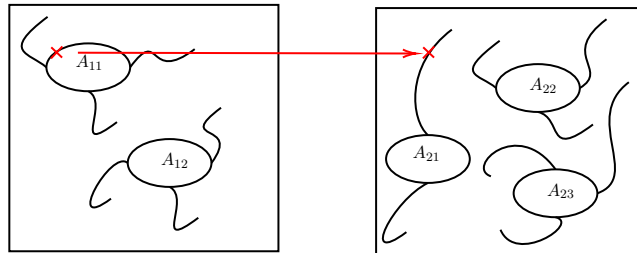
Soit un PBN de contextes  $F = (f^{(1)}, \dots, f^{(m)})$ . Pour chaque contexte  $f^{(k)}$ , on considère ses attracteurs  $A_{k,1}, \dots, A_{k,r_k}$ . L'ensemble des attracteurs du PBN est alors celui des  $A_{k,i}$ .

Contrairement à un BN sans perturbation, les attracteurs du PBN ainsi définis ne sont pas nécessairement stables, au sens où le système peut n'y rester que pour une durée finie. Il peut s'y échapper dans deux cas illustrés par la figure 1.11. Chaque sous-figure représente un STG : les ovales sont les attracteurs cycliques, les branches sont leurs bassins associés, et la croix rouge est l'emplacement de la configuration  $x \in \{0, 1\}^n$ .

- 1.11a : une perturbation modifie la configuration, et la déplace vers un autre bassin d'attraction
- 1.11b : une fonction de régulation différente est sélectionnée, partitionnant l'espace en de nouveaux bassins d'attraction



(a) Par perturbation.



(b) Par changement de contexte.

FIGURE 1.11 – Deux cas de sortie de l'attracteur.

Dans un BN avec perturbation, soumis à des sorties d'attracteurs de la forme 1.11a, la **stabilité re-**

**lative** entre deux attracteurs  $U$  et  $V$  désigne informellement la "facilité" à pouvoir transitionner d'un attracteur vers un autre. Plusieurs définitions formelles de ces indicateurs existent [19, 20]. En notant  $P$  la matrice de transition,  $\pi$  la loi stationnaire, et  $W$  la matrice dont chaque colonne est  $\pi$  :

- par la taille des bassins d'attraction :

$$RS_{bassin1}(U, V) = \log \left( \frac{|B_U|}{|B_V|} \right)$$

- par le taux de transition entre les bassins :

$$RS_{bassin2}(U, V) = \log \left( \frac{b_{UV}}{b_{VU}} \right) \quad , \text{ avec } b_{UV} = \sum_{i \in B_U} \sum_{j \in B_V} \frac{P_{ij}}{|B_V|}$$

- par la loi stationnaire :

$$RS_{stat}(U, V) = \log \left( \frac{\pi_U}{\pi_V} \right)$$

- par le temps moyen de premier passage [21, Th° 11.16] :

$$RS_{passage}(U, V) = \frac{1}{M_{uv}} - \frac{1}{M_{vu}}$$

$$\text{avec } u \in U, v \in V, \quad M_{ij} = \frac{Z_{ii} - Z_{ij}}{\pi_i} \quad \forall i, j \quad \text{et } Z = (Id - P + W)^{-1}.$$

Si  $RS_X(U, V) > 0$  pour un des  $RS_X$  présentés plus haut, alors l'attracteur  $U$  sera dit *plus stable que* l'attracteur  $V$ .

Lorsqu'ils sont évalués sur des réseaux inférés de données biologiques, ces indicateurs sont très fortement corrélés entre eux. Dans le cas où chaque attracteur du réseau correspond à une différenciation cellulaire (comme dans l'exemple biologique de la section 1.1.3), l'ordre partiel de stabilité permet d'inférer une généalogie entre les types cellulaires : cela se justifie par le fait que la transition d'un état naïf vers un état différencié est plus probable que la transition inverse. [22, sect. 3C et 3D]

La relation d'ordre partiel de stabilité défini par  $RS_{passage}$  est très peu sensible au facteur de perturbation  $p$ . Pour un réseau comptant un nombre de gènes  $n$  trop important, le calcul analytique de  $RS_{passage}$  n'est plus envisageable puisqu'il demande de calculer puis inverser une matrice  $M$  de taille  $2^n \times 2^n$ . On peut toutefois approcher stochastiquement une valeur  $M_{ij}$  en échantillonnant, sur la chaîne de Markov partant de l'état initial  $i$ , le nombre moyen d'itérations pour atteindre l'état  $j$ . [22, sect. 3E]

#### 1.4.5 Passage en revue des classes de modèles

Nous avons vu que la classe des PBN étaient une extension de la classe des BN synchrones. Résumons les classes possibles pour un PBN  $A = A^{p,q}(V, F, C)$ , selon les signes des paramètres  $q$  et  $p$  :

	$p = 0$	$p > 0$
$q = 0$	BN synchrone non-perturbé	BN synchrone perturbé
$q > 0$	PBN non-perturbé	PBN perturbé

TABLE 1.1 – Classes de BN synchrones.

La définition d'un PBN, et donc les cas de figures dans le tableau ci-dessus, ne prend en compte qu'une mise à jour *synchrone*. Pour les mêmes motivations de réalisme de la modélisation biologique, nous nous intéresserons dans le chapitre suivant à étendre la classe des PBN vers une mise à jour *asynchrone*.

## 1.5 Méthodes d'analyse des BNs et PBNs

Le modèle biologique présenté plus haut nous a fourni un exemple de réseau booléen dont les attracteurs s'interprètent comme des types vers lesquels une cellule peut se différencier. Dans un BN comme dans un PBN, il est alors important de pouvoir identifier les attracteurs ainsi que leur prévalence à long terme — notamment par la loi stationnaire de la chaîne de Markov associée.

Afin d'approcher la loi stationnaire d'une chaîne de Markov ergodique, une approche classique consiste à **itérer sa matrice de transition** [23, pp. 121-122] :

---

**Algorithme 1** : Approximation de la loi stationnaire par la *power method*

---

**Entrée** : Matrice de transition ergodique  $P$ , nombre d'itérations  $N$

**Sortie** :  $\pi$ , approximation du vecteur propre de  $P$  normalisé associé à la valeur propre 1

Initialiser aléatoirement la distribution  $\pi^0$  sur l'espace d'états  $\{0, 1\}^n$  ;

**pour chaque**  $k = 0$  à  $N - 1$  **faire**

$\pi^{(k+1)} = \pi^{(k)} P$ ;

**fin**

**retourner**  $\pi^N$  ;

---

Mais puisque chaque itération a un coût algorithmique en  $O(2^{2n})$ , l'approximation devient difficilement calculable même pour des valeurs faibles de  $n$ .

Une autre méthode s'appuie sur un corollaire de la loi des grands nombres appliquée aux chaînes de Markov, appelé théorème ergodique. Pour  $P$  une chaîne de Markov irréductible, pour tout état  $x$ , en notant  $N_x(n) = \sum_{k=0}^{n-1} \mathbb{1}_{X_k=x}$  le nombre de visites en  $x$  durant les  $n$  premières itérations de la chaîne et  $\pi_x$  sa probabilité stationnaire, on a la convergence suivante [24, Sect. 1.10] :

$$\frac{N_x(n)}{n} \xrightarrow{n \rightarrow \infty} \pi_x$$

La probabilité à long terme de se trouver dans un état  $x$  peut donc s'approximer empiriquement, en **échantillonnant les états observés en simulant la chaîne de Markov** [7, sect. 2.4] :

---

**Algorithme 2** : Approximation de la loi stationnaire

---

**Entrée** : Une chaîne de Markov, des seuils  $T, N, R \gg 1$  <sup>a</sup>

**pour chaque**  $i = 0$  à  $R$  **faire**

    Initialiser aléatoirement la configuration  $S^{(0)} \in \{0, 1\}^n$  ;

**pour chaque**  $t = 0$  à  $T - 1$  **faire**

        Transitionner la chaîne de Markov de  $S^{(t)}$  à  $S^{(t+1)}$  ;

**fin**

    Échantillonner de  $S^{(T)}$  à  $S^{(T+N)}$  ;

**fin**

**retourner** la moyenne des  $R$  histogrammes ;

---

a. La simulation de [7, sect. 3.2], illustrée en 1.12, fixe les paramètres à  $T = 2,000,000$ ,  $N = 6,000,000$ , et  $R = 500$ .

La présence d'un facteur de perturbation  $p > 0$  garantit l'ergodicité de la chaîne, et donc l'existence d'une unique loi stationnaire. Puisque la chaîne est irréductible et à espace d'états fini, la chaîne est récurrente positive, c'est-à-dire que la probabilité stationnaire de tout état est strictement positive. Néanmoins, les configurations de gènes observées expérimentalement comme prévalentes à long terme sont celles ayant une probabilité stationnaire significativement élevée dans la chaîne de Markov [7, sect. 3.2].

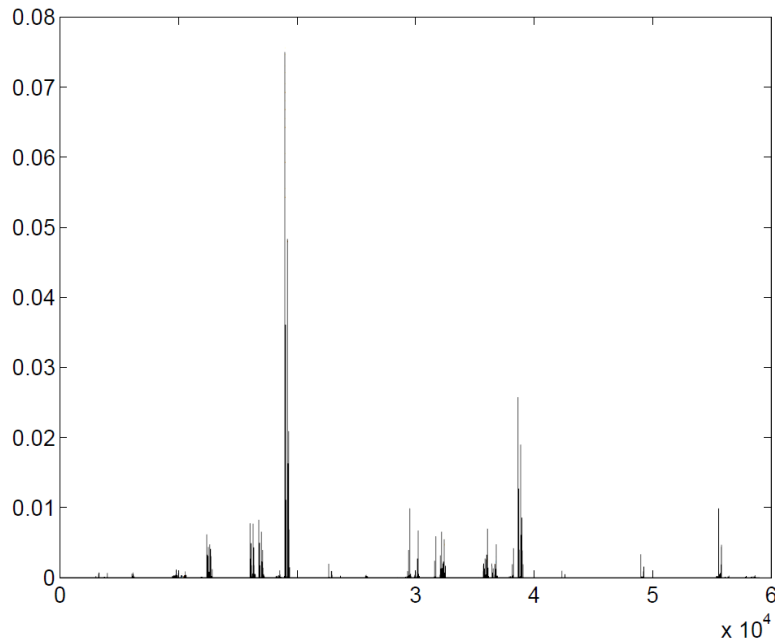


FIGURE 1.12 – Exemple de distribution stationnaire présenté dans [7, fig. 4a].

Le PBN étudié possède  $n = 10$  gènes, et n'admet pas deux mais trois niveaux d'expression  $(-1, 0, 1)$ , donc  $3^{10} = 59049$  états. Les états sont indicés en abscisse, et leur fréquence de visite en ordonnée.

Pour un BN perturbé synchrone ou un PBN synchrone, il existe une **expression analytique** [3] de la probabilité à long terme  $\pi(A)$  de se trouver dans un attracteur  $A$ . Le calcul de cette expression est détaillée en annexe C. Cependant, elle requiert la connaissance préalable de tous les attracteurs et de leurs bassins.

## 1.6 Outils logiciels

Des **outils informatiques** existants permettent de construire et étudier des modèles booléens ou des modèles booléens probabilistes.

GINSIM<sup>3</sup> se présente sous la forme d’une interface graphique : elle permet de définir un réseau de régulation génétique (cf. figs 1.1, 1.4a), définir les niveaux d’expression comme booléens ou multivalués, et définir les fonctions de régulation du réseau. L’interface de simulation génère le STG (cf. fig 1.3, 1.4b, 1.4c), en mise à jour synchrone comme asynchrone, avec possibilité de contraindre le niveau d’expression d’un ou plusieurs gènes. Le logiciel permet aussi d’exporter un modèle sous un format de fichier compatible avec d’autres logiciels ou modules (MABOSS, GRAPHVIZ, BOOLNET...)

BOOLNET<sup>4</sup> est un package du langage R rassemblant des méthodes de construction, simulation, et analyse de réseaux booléens. Il manipule des réseaux booléens synchrones comme asynchrones, ainsi que probabilistes — dans ce dernier cas, le modèle est fixé à  $q = 1$  et est indépendant. Il permet également d’obtenir des informations sur les attracteurs du réseau, ses bassins d’attraction, et de simuler sa chaîne de Markov. Une méthode génère synthétiquement des réseaux booléens, en réglant le nombre de nœuds et de voisins dans le graphe de régulation.

---

3. <http://ginsim.org/>

4. <https://cran.r-project.org/web/packages/BoolNet/>

## Chapitre 2

# Contributions

Ce chapitre présente mes ajouts personnels au domaine d'étude des réseaux booléens.

En première partie, j'y présente une méthode, dont je démontre la convergence, pour identifier les attracteurs et les bassins d'attraction à partir de la loi stationnaire empirique.

Dans les parties suivantes, j'y présente plusieurs algorithmes générant aléatoirement des réseaux booléens, et un autre formant des PBN à partir des fonctions voisines dans l'ordre partiel défini en 1.2.

Il se termine sur une section expérimentale, consacrée aux études de cas. Sur un modèle synthétique et un modèle biologique, j'emploie l'algorithme de loi stationnaire pour estimer la prévalence des attracteurs, et analyse l'effet des fonctions voisines sur le comportement à long terme.

Sauf mention contraire, les images dans les figures de ce chapitre proviennent d'un programme informatique que j'ai développé en langage PYTHON. À l'instar des logiciels présentés en 1.6, il permet de construire, simuler, analyser différentes classes de réseaux booléens, mais inclut aussi des paramètres et algorithmes originaux présentés dans ce mémoire. Sa documentation est disponible en annexe D, et son code source au lien suivant : <https://github.com/K4RI/pbn-simulation/>.

### 2.1 Interprétation de la loi stationnaire empirique

Revenons sur l'algorithme 2, et définissons formellement les barres de l'histogramme de distribution empirique obtenu par sa simulation.

Soit  $\mathcal{A} = G(\mathcal{V}, f)$  un réseau booléen.  $\forall x \in \{0, 1\}^n$ , on définit

$$S_{T,N,R}(x) = \frac{1}{NR} \sum_{i=1}^R \sum_{t=T+1}^{T+N} \mathbb{1}_{X_{it}=x}$$

comme la fréquence de visite de l'état  $x$  lors de l'échantillonnage, où  $\forall i \in \llbracket 1, R \rrbracket$ , chaque  $(X_{it})_{t \geq 0}$  est une réalisation du processus de Markov défini par  $\mathcal{A}$ , de point de départ  $X_{i0}$  tiré uniformément dans  $\{0, 1\}^n$ .

L'exemple à  $n = 4$  gènes illustré au chapitre précédent possède, d'après son STG en figure 1.3a, trois attracteurs en mise à jour synchrone : un cycle de taille 2  $\{0100, 1010\}$ , un cycle de taille 6  $\{0000, 0010, 0110, 1110, 1100, 1000\}$ , et un état stable  $\{1101\}$ . Lorsque l'on lui applique l'algorithme 2 (figure 2.1), deux constats se

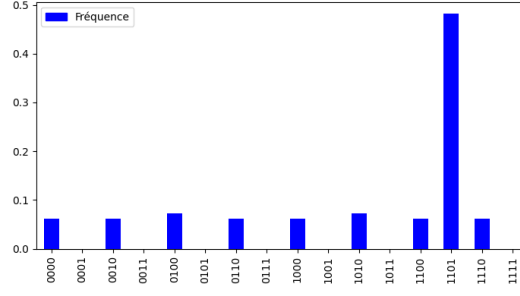


FIGURE 2.1 – Distribution stationnaire pour l'exemple synchrone des figures 1.1-1.2-1.3, avec  $T = 100$ ,  $N = 2000$ ,  $R = 2000$ .

font :

- Les états aux barres non-nulles sont exactement l'union des attracteurs
- Deux états appartenant au même attracteur ont des barres de même hauteur.

Dans cette section, nous allons énoncer et démontrer certains résultats de convergence de la distribution empirique pour un réseau booléen synchrone, afin d'estimer ses attracteurs et bassins d'attraction. Pour tout attracteur  $A$  et  $y \in B_A$ , notons  $\delta(y, A) = \min\{t \geq 0 \mid \text{Succ}^t(y) \cap A \neq \emptyset\}$  le temps d'atteinte de l'attracteur  $A$  depuis l'état  $y$ .

**Proposition 1** (Attracteurs et distribution stationnaire - synchrone).

Soit  $x \in \{0, 1\}^n$ . Si  $T > \max_{A \text{ attracteur}} \max_{y \in B_A} \delta(y, A)$ , alors

$$S_{T,N,R}(x) > 0 \implies x \in \bigcup_{A \text{ attracteur}} A$$

$$\lim_{N,R \rightarrow \infty} S_{T,N,R}(x) > 0 \iff x \in \bigcup_{A \text{ attracteur}} A$$

*Preuve.*

$\Rightarrow$

Pour  $y \in \{0, 1\}^n$ , notons  $A_y$  l'attracteur vers lequel converge  $y$ .

$$\begin{aligned} \lim_{N,R \rightarrow \infty} S_{T,N,R}(x) > 0 &\Rightarrow \exists i, t > T ; X_{it} = x \\ &\Rightarrow \exists x_0, t > T ; x \in \text{Succ}^t(x_0) \\ &\Rightarrow \exists x_0, t > T ; x \in \text{Succ}^{t-\delta(x_0, A_{x_0})}(\text{Succ}^{\delta(x_0, A_{x_0})}(x_0)) \\ &\Rightarrow \exists x_0, t > T ; x \in \text{Succ}^{t-\delta(x_0, A_{x_0})}(A_{x_0}) \\ &\Rightarrow \exists x_0, t > T ; x \in A_{x_0} \end{aligned}$$

Quelque soit l'état initial de la simulation où  $x$  a été échantillonné,  $x$  a été visité après  $T$  étapes, donc après le temps d'atteinte de l'attracteur de l'état initial.  $x$  appartient donc à l'attracteur de l'état



initial, et donc à  $\bigcup_{A \text{ attracteur}} A$ .

◀

Soit  $A$  un attracteur du réseau booléen, et  $x \in A$ . Nous allons montrer un résultat plus fort, duquel se déduit  $\Leftarrow$ .

**Proposition 2** (Convergence de la distribution stationnaire - synchrone).

Sur la même hypothèse,  $\forall A$  attracteur,  $\forall x \in A$ ,

$$\lim_{N, R \rightarrow \infty} S_{T, N, R}(x) = \frac{|B_A|}{|A| \cdot 2^n} \quad (2.1)$$

*Preuve.* On montre cette proposition à l'aide d'un lemme intermédiaire.

**Lemme.**  $\forall A$  attracteur,  $\forall x \in A$ , et  $i \geq 0$

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=T+1}^{T+N} \mathbb{1}_{X_{it}=x} = \begin{cases} \frac{1}{|A|} + \mathcal{O}(1/N) & \text{si } X_{i0} \in B_A \\ 0 & \text{sinon} \end{cases} \quad (2.2)$$

*Preuve.*

· Si  $X_{i0} \notin B_A$ , la chaîne de Markov  $(X_{it})_{t \geq 0}$  n'atteint aucun état de l'attracteur  $A$ , et a fortiori n'atteint jamais  $x$ .

· Si  $X_{i0} \in B_A$  :

À l'étape  $t = T + 1$ , puisque  $T > \delta(X_{i0}, A)$ , la chaîne se trouve déjà dans un état  $y \in A$ .

Soit  $\delta(y, x) = \min\{t \geq 0 \mid x \in \text{Succ}^t(y)\}$  le temps d'atteinte de  $x$  depuis  $y$ . L'attracteur  $A$  est cyclique :  $\delta(y, x)$  est compris entre 0 et  $|A| - 1$ , et de plus  $X_{it} = x$  si et seulement s'il existe un entier  $k$  tel que  $t = T + 1 + \delta(y, x) + k \cdot |A|$ .

La simulation est échantillonnée entre  $T + 1$  et  $T + N$ , le nombre de visites de l'état  $x$  est donc :

$$\sum_{t=T+1}^{T+N} \mathbb{1}_{X_{it}=x} = 1 + \left\lfloor \frac{N - 1 - t_{yx}}{|A|} \right\rfloor$$

Ainsi,

$$\frac{1}{N} \sum_{t=T+1}^{T+N} \mathbb{1}_{X_{it}=x} = \frac{1}{|A|} + \frac{\epsilon_i(x)}{N}$$

où  $\epsilon_i(x) = 1 - \frac{1 + \delta(y, x)}{|A|} - \left\{ \frac{N - 1 - \delta(y, x)}{|A|} \right\}$ <sup>1</sup> est compris entre  $-1$  et  $1$ . ■

Avec ce résultat, on peut réécrire :

---

1.  $\{x\} = x - \lfloor x \rfloor \in [0, 1[$  est la partie fractionnaire de  $x$ .

$$S_{T,N,R}(x) = \frac{1}{R} \sum_{i=1}^R \left[ \left( \frac{1}{|A|} + \frac{\epsilon_i(x)}{N} \right) \mathbb{1}_{X_{i0} \in B_A} \right] = \frac{1}{|A|} \cdot p_{AR} + \epsilon'_N(x) \quad (2.3)$$

avec  $p_{AR} = \frac{1}{R} \sum_{i=1}^R \mathbb{1}_{X_{i0} \in B_A}$ , et  $\epsilon'_N(x) := \sum_{i=1}^R \frac{\epsilon_i(x)}{N} \mathbb{1}_{X_{i0} \in B_A}$ , on a alors  $|\epsilon'_N(x)| \leq \frac{1}{N}$ .

À chaque simulation  $i$  dans l'algorithme 2, l'état de départ  $X_{i0}$  est tiré uniformément dans  $\{0, 1\}^n$ . Les  $\mathbb{1}_{X_{i0} \in B_A}$  est donc des variables aléatoires indépendantes suivant une même loi de Bernoulli de paramètre et d'espérance  $\frac{|B_A|}{2^n}$ . Par application de la loi des grands nombres,  $p_{AR}$  converge presque sûrement vers  $\mathbb{E}(\mathbb{1}_{X_{10} \in B_A}) = \frac{|B_A|}{2^n}$ .

$$S_{T,N,R}(x) = \underbrace{\frac{1}{|A|} \cdot p_{AR}}_{\xrightarrow{R \rightarrow \infty} \frac{|B_A|}{|A| \cdot 2^n}} + \underbrace{\epsilon'_N(x)}_{\xrightarrow{N \rightarrow \infty} 0} \xrightarrow{N, R \rightarrow \infty} \frac{|B_A|}{|A| \cdot 2^n}$$

■

La limite  $\lim_{N,R \rightarrow \infty} S_{T,N,R}(x) = \frac{|B_A|}{|A| \cdot 2^n}$  étant strictement positive pour tout attracteur  $A$  et  $x \in A$ , on en déduit le sens indirect de la proposition 1. ■

La proposition 1 nous confirme qu'avec des paramètres  $T, N, R$  suffisamment hauts, l'algorithme de distribution stationnaire permet de détecter les attracteurs du réseau booléen synchrone comme ceux dont la probabilité est non-nulle dans la distribution.

La proposition 2, quant à elle, fournit deux informations intéressantes :

- Les états d'un même attracteur ont asymptotiquement la même fréquence stationnaire
- Si l'on connaît la taille  $|A|$  de l'attracteur dont fait partie un état  $x$ , on peut approcher la **proportion du bassin d'attraction**  $p_A := \frac{|B_A|}{2^n}$  par  $|A| \cdot S_{T,N,R}(x)$ .

Dans l'hypothèse où il n'existe pas deux attracteurs  $|A_1|$  et  $|A_2|$  distincts tels que  $\frac{|B_{A_1}|}{|A_1|} = \frac{|B_{A_2}|}{|A_2|}$ , on a  $\forall x, y \in \bigcup_{A \text{ attracteur}} A$  :

$$A_x \neq A_y \iff \lim_{N,R \rightarrow \infty} S_{T,N,R}(x) \neq \lim_{N,R \rightarrow \infty} S_{T,N,R}(y)$$

Il suffit alors de partitionner par hauteur de barres les états visités pour trouver les attracteurs.

Pour prévenir le cas où cette hypothèse n'est pas vérifiée, il convient, pour chaque classe  $C \in \mathcal{C}$  d'états regroupés car de hauteur proche, de simuler le réseau à partir d'un  $x \in C$  jusqu'à revenir à  $x$  — c'est-à-dire parcourir l'attracteur cyclique  $A_x$ .

- si  $A_x = C$ ,  $C$  est bien un attracteur.
- sinon,  $A_x \subsetneq C$ . On consigne  $A_x$  comme attracteur, et on relance l'opération sur un  $y \in C \setminus A_x$ .

De cette manière on identifie l'ensemble  $\mathcal{C}'$  des attracteurs du réseau booléen. Il vérifie  $\bigcup_{A' \in \mathcal{C}'} A' = \bigcup_{A \in \mathcal{C}} A$ , et  $\forall A' \in \mathcal{C}', \exists A \in \mathcal{C} \mid A' \subset A$ .

On peut donc identifier les attracteurs en groupant les états par hauteur des barres, puis en partitionnant ces groupes si besoin. Identifier un attracteur  $A$  nous informe de sa taille  $|A|$ . À partir de la fréquence stationnaire de n'importe quel  $x \in A$ , on approche la proportion  $p_A := \frac{|B_A|}{2^n}$  du bassin d'attraction avec  $\widehat{p_{ANR}} = |A| \cdot S_{T,N,R}(x)$ .

**Proposition 3** (Intervalle de confiance sur la proportion du bassin d'attraction - synchrone).

Sur la même hypothèse,  $p_A := \frac{|B_A|}{2^n}$ , et  $\widehat{p_{ANR}} := |A| \cdot S_{T,N,R}(x)$ .

Pour  $N$  et  $R$  suffisamment élevés, l'intervalle de confiance à  $100(1 - \alpha)\%$  de  $p_A$  est :

$$p_A = \widehat{p_{ANR}} \pm \left[ \left( q_{1-\alpha/2} \sqrt{\widehat{p_{ANR}}(1 - \widehat{p_{ANR}})} \right) \cdot \frac{1}{\sqrt{R}} + \left( |A| + \frac{|A| \cdot q_{1-\alpha/2}}{\sqrt{R}} \right) \cdot \frac{1}{N} \right]$$

où  $q_{1-\alpha/2}$  est le  $(1 - \alpha/2)$ -quantile de la distribution normale centrée réduite.

Remarque : Pour 90%,  $q_{1-\alpha/2} \approx 1.65$ ; pour 95%,  $q_{1-\alpha/2} \approx 1.96$ ; pour 99%,  $q_{1-\alpha/2} \approx 2.58$ .

*Preuve.*

Nous avons vu au cours de la démonstration de la proposition 2, l'équation 2.3 :

$$\widehat{p_{ANR}} = p_{AR} + |A| \cdot \epsilon'_N(x)$$

$p_{AR}$  est la proportion de succès d'une distribution binomiale  $B(R, p_A)$ . Sous réserve d'un nombre d'essais  $R$  assez haut, l'intervalle de confiance de Wald [25] du paramètre  $p_A$  à  $100(1 - \alpha)\%$  est :

$$p_{AR} \pm q_{1-\alpha/2} \sqrt{p_{AR}(1 - p_{AR})} \cdot \frac{1}{\sqrt{R}}$$

autrement écrit,  $\mathbb{P} \left( |p_{AR} - p_A| < q_{1-\alpha/2} \sqrt{p_{AR}(1 - p_{AR})} \cdot \frac{1}{\sqrt{R}} \right) \geq 1 - \alpha$ .

Il s'agit à présent d'élargir l'intervalle, afin d'avoir une expression sur  $\widehat{p_{ANR}}$  plutôt que  $p_{AR}$ .

Par l'inégalité triangulaire  $|\widehat{p_{ANR}} - p_A| \leq |p_{AR} - p_A| + |\widehat{p_{ANR}} - p_{AR}| \leq |p_{AR} - p_A| + \frac{|A|}{N}$ , on a :

$$\mathbb{P} \left( |\widehat{p_{ANR}} - p_A| < q_{1-\alpha/2} \sqrt{p_{AR}(1 - p_{AR})} \cdot \frac{1}{\sqrt{R}} + \frac{|A|}{N} \right) \geq 1 - \alpha$$

Calculons ensuite la proximité entre  $\sqrt{p_{AR}(1 - p_{AR})}$  et  $\sqrt{\widehat{p_{ANR}}(1 - \widehat{p_{ANR}})}$ .

La fonction  $t \mapsto t(1 - t)$  étant de dérivée  $1 - 2t$  :

$$\begin{aligned} \sqrt{p_{AR}(1 - p_{AR})} - \sqrt{\widehat{p_{ANR}}(1 - \widehat{p_{ANR}})} &\leq \int_{\widehat{p_{ANR}}}^{p_{AR}} (1 - 2t) dt \leq \left| \int_{\widehat{p_{ANR}}}^{p_{AR}} 1 dt \right| \\ &= |p_{AR} - \widehat{p_{ANR}}| = |A| \cdot |\epsilon'_N(x)| \leq \frac{|A|}{N} \end{aligned}$$

On trouve donc :

$$\mathbb{P} \left( |\widehat{p_{ANR}} - p_A| < q_{1-\alpha/2} \left[ \sqrt{\widehat{p_{ANR}}(1 - \widehat{p_{ANR}})} + \frac{|A|}{N} \right] \cdot \frac{1}{\sqrt{R}} + \frac{|A|}{N} \right) \geq 1 - \alpha$$

ce qui nous permet de conclure. ■

Ainsi, cette méthode permet d'estimer la *proportion relative* de l'espace occupée par le bassin d'attraction. L'erreur absolue sur  $|B_A|$  correspond à cette erreur relative multipliée par  $2^n$ , d'ordre  $\mathcal{O}(2^n \cdot (\frac{1}{\sqrt{R}} + \frac{1}{N}))$ . Mais lorsque l'on étudie la différenciation de cellules vers d'autres phénotypes biologiques, l'information recherchée concerne in fine la prévalence relative de ces types (cf. l'étude de cas en section 2.5). La proportion relative des bassins d'attraction est un bon indicateur dans ce sens.

---

**Algorithme 3** : Détection des attracteurs et des bassins d'attraction d'un BN synchrone

---

**Entrée** : Un réseau booléen  $bn$ , des paramètres  $T, N, R$

**Sortie** : Ensemble des attracteurs avec proportion des bassins d'attraction et intervalle de confiance à 95 %

$S :=$  histogramme de l'algorithme 2 appliqué à  $bn$  ;

Partitionner en un ensemble  $\mathcal{C}$  de classes les états selon leurs valeurs dans  $S$  ;

Partitionner chaque  $C \in \mathcal{C}$  en attracteurs, et les inscrire dans  $\mathcal{C}'$  ;

$\mathcal{C}_+ := \emptyset$ ;

**pour chaque**  $A \in \mathcal{C}'$  **faire**

Sélectionner aléatoirement un  $x \in A$  ;

$p_A = |A| \cdot S(x)$  ;

$err_A = 1.96 \cdot \sqrt{p_A(1 - p_A)} \cdot \frac{1}{\sqrt{R}} + \frac{|A| + |A|q_{1-\alpha/2}/\sqrt{R}}{N}$  ;

Ajouter  $[A, p_A, err_A]$  à  $\mathcal{C}_+$  ;

**fin**

**retourner**  $\mathcal{C}_+$  ;

---

Cet algorithme est, comme le 2, de complexité  $\mathcal{O}((T + N) \cdot R)$ , et fournit un résultat avec une marge d'erreur en  $\mathcal{O}(\frac{1}{\sqrt{R}} + \frac{1}{N})$ .

Revenons à l'exemple repris dans la figure 2.1 en début de section. Dans cette distribution stationnaire, on peut effectivement partitionner les états visités selon la hauteur des barres, puis identifier les attracteurs  $\{0100, 1010\}$ ,  $\{0000, 0010, 0110, 1110, 1100, 1000\}$ , et  $\{1101\}$ . Estimons la proportion de leurs bassins d'attraction avec les instructions dans la boucle de l'algorithme 3 :

Attracteur	Proportion du bassin calculée par l'algo 3	Proportion réelle (cf. fig. 1.3a)
$\{0100, 1010\}$	12.2% $\pm$ 1.6 %	$2/16 = 12.5\%$
$\{0000, 0010, 0110, 1110, 1100, 1000\}$	38.3% $\pm$ 2.7%	$6/16 = 37.5\%$
$\{1101\}$	49.5% $\pm$ 2.3%	$8/16 = 50\%$

TABLE 2.1 – Table des résultats de l'algorithme 3 sur l'exemple.

Les résultats obtenus en table 2.1 correspondent à la vraie proportion des bassins d'attraction observée dans le STG, ce qui illustre que l'on peut correctement identifier *et* estimer la prévalence des attracteurs d'un réseau en mise à jour synchrone.

En mise à jour **asynchrone**, l'existence d'un seuil pour  $T$  au-delà duquel on est certain de n'échantillonner exclusivement des attracteurs n'est garantie que par *l'absence de cycles non-terminaux*.

En effet, si le réseau n'en possède pas, le sens direct de la proposition 1 se vérifie pour  $T >$

$\max_{x \in \{0,1\}^n} \delta'(x)$ , avec  $\delta'(x) = \min\{t \geq 0 \mid \text{Succ}^t(y) \subset \bigcup_{A \text{ attracteur}} A\}$  le temps garanti d'atteinte des attracteurs. Il est envisageable<sup>2</sup> d'adapter en inégalités le lemme de la proposition 2, et donc la proposition 2 elle-même :

**Conjecture 2bis** (Encadrement de la distribution stationnaire - asynchrone).

Pour un réseau asynchrone sans cycles non-terminaux,  $T > \max_{x \in \{0,1\}^n} \delta'(x)$ ,  
 $\forall A \text{ attracteur}$ ,

$$\frac{|\bar{B}_A|}{2^n} \leq \lim_{N,R \rightarrow \infty} \sum_{x \in A} S_{T,N,R}(x) \leq \frac{|B_A|}{2^n}$$

Dans ce cas, l'union des attracteurs reste identifiable par les barres non-nulles de la distribution stationnaire. Dans les sous-cas particuliers où tous les bassins larges sont aussi stricts (c'est le cas dans notre exemple en mise à jour asynchrone, fig. 1.3b), les inégalités de la conjecture 2bis deviendraient des égalités, ce qui reviendrait à pouvoir approcher  $p_A$  par  $\sum_{x \in A} S_{T,N,R}(x)$  en adaptant l'algorithme 3.

## 2.2 PBNs asynchrones

À partir des classes de modèles définies en section 1.4.5 et de la définition d'une mise à jour asynchrone, on peut définir la classe des PBN asynchrones comme les processus stochastiques de mêmes paramètres et étapes d'itération que présentés en section 1.4.1, mais dont les successeurs d'un état par appel de la fonction seraient calculés comme pour un BN asynchrone.

Cette notion sera notamment employée dans les algorithmes de la section 2.4 et les études de cas de la section 2.5.

## 2.3 Génération de BNs synthétiques

Afin de généraliser mes observations, j'ai également travaillé à la génération de réseaux booléens synthétiques aléatoires. On génère d'abord le graphe de régulation, puis des fonctions conformes à ce graphe, selon deux méthodes. Les paramètres du modèle sont le nombre  $n$  de gènes (entier), et le nombre  $k$  de régulateurs (entier, ou liste d'entiers s'il n'est pas le même pour tous les gènes).

### 2.3.1 Régulations strictement activatrices ou inhibitrices

Pour chaque  $g_i$  ( $1 \leq i \leq n$ ), on tire  $k_i$  voisins entrants. Chaque voisin a une probabilité  $p_{neg}$  ( $= 1/2$  par défaut) d'être inhibiteur de  $g_i$ , et  $1 - p_{neg}$  d'être son activateur.

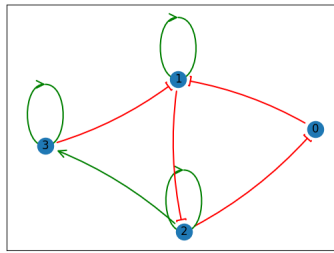
L'annexe de [26] propose, à partir d'un graphe de régulation, de décrire un réseau booléen par les fonctions suivantes. Pour  $1 \leq i \leq n$  :

---

2. Merci à M. Guillaume THEYSSIER pour cette suggestion.

$$f_i(x) = \begin{cases} \left( \bigvee_{j \in Reg_i^+} x_j \right) \wedge \neg \left( \bigvee_{j \in Reg_i^-} x_j \right) & , \text{ si } g_i \text{ a des activateurs et des inhibiteurs} \\ \bigvee_{j \in Reg_i^+} x_j & , \text{ si } g_i \text{ n'a que des activateurs} \\ \neg \left( \bigvee_{j \in Reg_i^-} x_j \right) & , \text{ si } g_i \text{ n'a que des inhibiteurs.} \end{cases} \quad (2.4)$$

Un gène s'active à la prochaine étape lorsqu'aucun de ses inhibiteurs n'est activé, et au moins un de ses activateurs (s'il en possède) est activé. À partir du graphe de régulation généré plus haut, on calcule alors, univoquement avec l'équation 2.4, les fonctions de régulation de chaque gène.



(a) Graphe de régulation généré.

$$\begin{aligned} f_0(x) &= \neg x_2 \\ f_1(x) &= x_1 \wedge \neg(x_0 \vee x_3) \\ f_2(x) &= x_2 \wedge \neg x_1 \\ f_3(x) &= x_2 \vee x_3 \end{aligned}$$

(b) Fonctions correspondantes.

FIGURE 2.2 – Exemple de génération de réseau booléen à régulation signée.

Le graphe indique que le gène  $g_1$  est activé par  $g_1$  et inhibé par  $g_0$  et  $g_3$ , d'où la fonction  $f_1(x) = x_1 \wedge \neg(x_0 \vee x_3)$ .

### 2.3.2 Régulations duales autorisées<sup>3</sup>

Contrairement à la méthode précédente contraignant les fonctions à satisfaire des régulations strictement activatrices ou inhibitrices, celle-ci ne précise pas le signe de régulation.

Pour chaque  $g_i$  ( $1 \leq i \leq n$ ), on tire  $k_i$  voisins entrants pour former le graphe de régulation. Ici, la nature de la régulation d'un gène à un autre (activation, inhibition...) n'est pas précisée.

Le gène  $g_i$  possède  $k_i$  régulateurs dans le graphe, il existe donc  $2^{k_i}$  fonctions de régulation potentielles. Chacune de ces fonctions est identifiable par une table de vérité de taille  $2^{k_i}$ . Pour en piocher une uniformément parmi les fonctions potentielles, on génère donc un mot binaire de taille  $2^{k_i}$ ; il est inséré comme dernière colonne de la table de vérité, laquelle représente alors la fonction  $f_i$  sélectionnée du réseau booléen.

Dans la figure 2.3, le gène  $g_i$  est régulé via la fonction  $f_i$  par 3 gènes, de variables  $x_2$ ,  $x_3$ , et  $x_5$ . Il possède donc  $2^3 = 8$  combinaisons d'entrées, inscrites sur la gauche du tableau. La sortie de chaque combinaison est notée en rouge sur la dernière colonne du tableau, c'est le mot binaire de taille 8 généré aléatoirement. Prenons le cas de la ligne en gras : un état pour lequel les gènes ( $g_2$ ,  $g_3$ ,  $g_5$ ) ont pour niveaux d'activation respectifs ( $x_2 = 0$ ,  $x_3 = 1$ ,  $x_5 = 1$ ), l'étape suivante verra le gène  $i$  activé, puisque  $f_i(x) = 1$ .

3. Analogue à la méthode `generateRandomNKNetwork()` de BoolNet : <https://cran.r-project.org/web/packages/BoolNet/BoolNet.pdf#page.13>

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$\longrightarrow$	$f_i(x)$
*	0	0	*	0	$\longrightarrow$	1
*	0	0	*	1	$\longrightarrow$	0
*	0	1	*	0	$\longrightarrow$	0
*	0	1	*	1	$\longrightarrow$	1
*	1	0	*	0	$\longrightarrow$	1
*	1	0	*	1	$\longrightarrow$	0
*	1	1	*	0	$\longrightarrow$	0
*	1	1	*	1	$\longrightarrow$	1

FIGURE 2.3 – Exemple de génération de table de vérité, pour un gène  $i$  régulé par les variables  $x_2$ ,  $x_3$  et  $x_5$ .

Notons que contrairement à la méthode précédente, celle-ci peut engendrer des régulations duales ou non-fonctionnelles. Dans l'exemple 2.3, on observe que  $f_i(000) = 1$  et  $f_i(001) = 0$ , donc  $g_5$  n'est pas activateur du gène  $g_i$ , or  $f_i(010) = 0$  et  $f_i(011) = 1$ , donc  $g_5$  n'est pas inhibiteur du gène  $g_i$ . Ainsi,  $g_5$  est *régulateur dual* de  $g_i$ . De même, on constate que  $\forall x, f_i(x_1, x_2, x_3, x_4, x_5) = f(x_1, \neg x_2, x_3, x_4, x_5)$ . Ainsi, le gène  $g_2$  n'est pas un *régulateur fonctionnel* de  $g_i$ .

## 2.4 Génération de PBNs

### 2.4.1 Définition à l'aide des fonctions voisines

Considérons un BN  $\mathcal{A} = G(V, f)$ , dans lequel on souhaite introduire un niveau de stochasticité en associant à chaque gène  $i$  non seulement sa fonction de référence, mais aussi les voisins de cette fonction dans le PO-Set de ses fonctions consistantes comme présenté en section 1.2.

Les paramètres de construction sont :

- $I \subseteq V$  la liste des gènes dont les fonctions sont à étendre
- $d \in \mathbb{N}$  la distance à laquelle rechercher les voisins dans les diagrammes de Hasse des PO-Sets
- $p_{ref} \in [0, 1]$  la probabilité associée à la fonction de référence
- $p \in [0, 1]$  et  $q \in [0, 1]$  les facteurs de perturbation et de changement de contexte.

Le PBN des fonctions voisines  $\mathcal{A}' = A^{p,q}(V, (F_1, \dots, F_n), C)$  est indépendant et se construit ainsi. Pour chaque  $i \in I$  :

1. On explore  $F_i$  l'ensemble des fonctions voisines de  $f_i$  à distance inférieure ou égale à  $d$ .
2. La distribution  $c_i$  associe  $c_i(f_i) = p_{ref}$  à la fonction de référence, et répartit le reste  $\sum_{f \in F_i, f \neq f_i} c_i(f) = 1 - p_{ref}$  entre les autres fonctions.

Pour  $i \in V \setminus I$ , la fonction reste inchangée :  $F_i = \{f_i\}$ , et  $c_i(f_i) = 1$ .

Notons que si  $p_{ref} = 1$ , le modèle obtenu se comporte comme le BN initial. Si  $p_{ref} = 0$ , le PBN obtenu n'emploie plus les fonctions de référence initiales, mais seulement leurs fonctions voisines explorées.

\*L'exploration locale du diagramme de Hasse se fait à l'aide du package Java FUNCTIONHOOD<sup>4</sup>, dont les détails des algorithmes sont présentés dans [9]. Les méthodes `getFormulaParents()` (resp. `getFormulaChildren()`) prennent en argument une antichaine correspondant à une formule, et renvoient la liste

4. <https://github.com/ptgm/functionhood/>

de ses parents (resp. de ses enfants) dans le diagramme de Hasse. Dans mon programme, je l'emploie dans cet ordre :

1. Conversion de la formule en antichaîne.
2. Avec un module de communication avec Java (ici Py4J<sup>5</sup>), appels successifs des méthodes *getFormulaParents()* et/ou *getFormulaChildren()* selon la distance voulue et avec l'antichaîne en entrée, puis collecte de la liste des antichaînes voisines en sortie.
3. Conversion des antichaînes en formules, qui sont les formules voisines.

## 2.4.2 Définition à l'aide des tables de vérité aléatoires

Dans un PBN, il est important que les différents contextes représentent les mêmes phénomènes de régulation. Ainsi dans les procédés présentés dans cette section, on génère dans un premier temps un graphe de régulation non-signé comme en section 2.3.2. Puis dans un second temps, on génère les fonctions de régulation selon l'une des deux méthodes suivantes. Les paramètres du modèle sont le nombre  $n$  de gènes (entier), et le nombre  $k$  de régulateurs (entier, ou liste d'entiers si le nombre de fonctions par classe n'est pas le même pour tous les gènes), et le nombre  $m$  de contextes en cas non-indépendant/de fonctions par classe en cas indépendant (entier, ou liste d'entiers si le nombre de fonctions par classe n'est pas le même pour tous les gènes).

### PBN non-indépendant

Il s'agit ici de générer un nombre  $m$  de contextes, c'est-à-dire de vecteurs  $f^{(j)} = (f_{j1}, \dots, f_{jn})$  de fonctions de régulation. Un contexte définissant un BN, on génère chacun d'entre eux à partir du graphe de régulation selon le procédé décrit en 2.3.2.

### PBN indépendant

Les fonctions de régulation d'un PBN indépendant sont organisées en classes :  $F_1, \dots, F_n$ , avec  $F_i$  l'ensemble des fonctions de régulation potentielles du gène  $g_i$ .

Ainsi pour chaque gène  $g_i$  dont la classe est de taille  $m_i = |F_i|$ , on génère  $m_i$  fonctions via des tables de vérité sur les régulateurs de  $g_i$ .

## 2.5 Études de cas

Dans des exemples de régulation biologiques ou synthétiques, en mise à jour synchrone ou asynchrone, on étudie comment cette altération des fonctions de régulation affecte la dynamique du réseau.

### 2.5.1 Modèles synthétiques

À l'aide des méthodes décrites plus haut, il est possible de générer des modèles booléens à partir de paramètres donnés. Dans cette section, on comparera les propriétés dynamiques (nombre, type, taille des attracteurs) de BN générés synthétiquement avec celles de leurs PBN des fonctions voisines, dont la construction est étudiée en section 2.4.1. On précisera ensuite, en analysant ces exemples, comment l'ajout de nouvelles fonctions mène à des altérations sur la dynamique du réseau.

---

5. <https://www.py4j.org/index.html>



Soit  $n$  le nombre de gènes, et  $\mathcal{V}' = \{0, 1\}^n$  l'espace d'états. On s'intéresse aux propriétés sur des ensembles d'attracteurs, c'est-à-dire aux fonctions prenant en entrée un ensemble d'ensembles d'états et renvoyant Vrai ou Faux.

Exemples : ici un  $\mathcal{C}_i$  est une propriété, et l'argument  $a$  est un ensemble d'attracteurs.

$\mathcal{C}_1(a) = "a \text{ contient un attracteur de taille } 2"$ .  $\mathcal{C}_1(\{\{000\}, \{010, 101\}\})$  est vrai, mais  $\mathcal{C}_1(\{\{000\}, \{010\}\})$  est faux.  
 $\mathcal{C}_2(a) = "a \text{ ne contient que des points fixes}"$ .  $\mathcal{C}_2(\{\{000\}, \{010\}\})$  est vrai, mais  $\mathcal{C}_2(\{\{000\}, \{010, 101\}\})$  est faux.

Les exemples recherchés de BN sont ceux tels qu'un PBN de leurs fonctions voisines altère ses attracteurs. Mon algorithme qui suit prend en argument deux propriétés  $\mathcal{C}_1$  et  $\mathcal{C}_2$ , et génère des réseaux aléatoires synthétiques jusqu'à trouver un cas, s'il existe, tel que les attracteurs du BN satisfont  $\mathcal{C}_1$ , et les attracteurs du PBN des fonctions voisines satisfont  $\mathcal{C}_2$ .

---

**Algorithme 4 :** Génération de modèles booléens vérifiant certaines propriétés dynamiques

---

**Entrée :** Paramètres de génération de BN, paramètres d'exploration de fonctions voisines, deux propriétés  $\mathcal{C}_1$  et  $\mathcal{C}_2$  sur des ensembles d'attracteurs, un seuil d'itération  $t_{max}$

**Sortie :** Un réseau booléen et son PBN étendu aux fonctions voisines, dont les attracteurs satisfont respectivement  $\mathcal{C}_1$  et  $\mathcal{C}_2$

**pour chaque**  $t = 0$  à  $t_{max}$  **faire**

$bn$  = BN aléatoire à régulations activatrices ou inhibitrices généré synthétiquement ;

$pbn\_ext$  = PBN des fonctions voisines de  $bn$  ;

$a_1$  = ensemble des attracteurs de  $bn$  ;

$a_2$  = ensemble des attracteurs de  $pbn\_ext$  ;

**si**  $\mathcal{C}_1(a_1) \wedge \mathcal{C}_2(a_2)$  **alors**

**retourner**  $bn, pbn\_ext$

**fin**

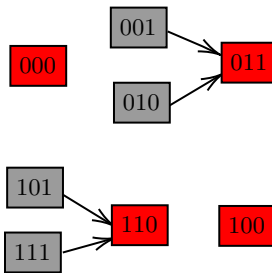
**fin**

**retourner** 0

---

J'ai ainsi recherché un modèle synchrone à  $n = 3$  gènes tel que les attracteurs vérifient  $\mathcal{C}_1 =$  "tous les attracteurs sont des points fixes et il n'en existe que 4", et tel que ceux du PBN des fonctions *enfants* vérifient  $\mathcal{C}_2 =$  "tous les attracteurs sont des points fixes et il n'en existe que 3". Un résultat est le BN défini par les fonctions suivantes :

$$\begin{aligned} f_0(x) &= x_0 \\ f_1(x) &= x_1 \vee x_2 \\ f_2(x) &= (\neg x_0 \wedge x_1) \vee (\neg x_0 \wedge x_2) \end{aligned}$$



En effet, le STG de ce réseau est :

FIGURE 2.4 – STG du BN initial.

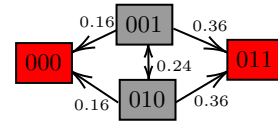
Il possède bien quatre états stables coloriés en rouge (les autoboucles sont masquées pour la lisibilité),

les états transients sont coloriés en gris.

L'étape suivante recherche les enfants de  $f_0, f_1, f_2$  dans leurs diagrammes de Hasse respectifs. Avec les fonctions enfants écrites en gras, les classes  $F_i = \{f_i + \textbf{enfants de } f_i\}$  sont :

$$F_0 = \{x_0\} \quad F_1 = \{x_1 \vee x_2, \mathbf{x_1 \wedge x_2}\} \quad F_2 = \{(\neg x_0 \wedge x_1) \vee (\neg x_0 \wedge x_2), \neg \mathbf{x_0 \wedge x_1 \wedge x_2}\}$$

La distribution de probabilité dans chaque classe est la suivante : la fonction de référence est sélectionnée avec une probabilité  $p_{ref}$ , et l'autre avec une probabilité  $1 - p_{ref}$ . Dans les expériences de cette section,  $p_{ref} = 0.6$ .



Ce PBN indépendant possède  $1 \times 2 \times 2 = 4$  contextes. Son STG est :

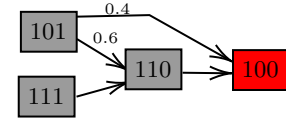


FIGURE 2.5 – STG du PBN étendu aux enfants.

Les auto-boucles ne sont pas indiquées pour des raisons de lisibilité, mais elles sont bien présentes aux états stables (rouge) et aux états dont les poids sortants affichés ne somment pas à 1.

Le modèle étant devenu non-déterministe, la présence de plusieurs arcs sortants d'un même état rend compte de ses multiples successeurs potentiels. On observe deux changements dans la dynamique :

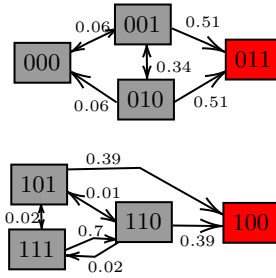
- 110 n'est plus un point fixe, puisqu'il admet 100 comme successeur
- 001 et 010 ne sont plus dans le bassin d'attraction strict de 011, mais dans son bassin large et dans celui de 000.

Même sans le graphe de transition d'états, les nouveaux arcs peuvent se déterminer à partir des nouvelles fonctions :

- dans le cas de base, le successeur de  $x = 110$  est lui-même. Mais si  $x_1 \wedge x_2$  est sélectionnée comme fonction de régulation de  $g_1$ , sa sortie en  $x = 110$  vaut 0, transitionnant de 110 vers 100. Ainsi 110 n'est plus un point fixe.
- si les deux fonctions enfants sont respectivement choisies pour réguler  $g_1$  et  $g_2$  (à lieu avec une probabilité  $(1 - p_{ref})^2 = 0.16$ ), elles auront pour sortie 0 à partir des états 001 ou 010, les faisant transitionner vers le point fixe 000. Ainsi, 001 ou 010 se trouvent dans le bassin d'attraction large de 000.

Les fonctions voisines de  $f_0$  et  $f_1$ , à respectivement zéro et un régulateur, ont été toutes explorées. Mais en étendant la classe  $F_2$  jusqu'aux parents (n°3), frères (n°4-5) et grands-parents (n°6-7-8) de la fonction de référence  $f_2$  dans son diagramme de Hasse, elle devient :

$$\begin{aligned}
F_2 = \{ & (\neg x_0 \wedge x_1) \vee (\neg x_0 \wedge x_2), \\
& \neg x_0 \wedge x_1 \wedge x_2, \\
& (\neg x_0 \wedge x_1) \vee (\neg x_0 \wedge x_2) \vee (x_1 \wedge x_2), \\
& (\neg x_0 \wedge x_2) \vee (x_1 \wedge x_2), \\
& (\neg x_0 \wedge x_1) \vee (x_1 \wedge x_2), \\
& x_2 \vee (\neg x_0 \wedge x_1), \\
& x_1 \vee (\neg x_0 \wedge x_2) \\
& \neg x_0 \vee (x_1 \wedge x_2) \}
\end{aligned}$$



Le STG du PBN obtenu est :

FIGURE 2.6 – STG du PBN étendu aux grands-parents.

Cette extension a une fois de plus retiré un point fixe : 000. En effet, si  $\neg x_0 \vee (x_1 \wedge x_2)$  est choisie comme fonction de régulation de  $g_2$ , elle aura une sortie 1 à partir de l'entrée  $x = 000$ , faisant transitionner 000 vers 001, ce qui en fait un état non-terminal donc transient.

La prévalence des attracteurs est une question primordiale dans les applications biologiques des réseaux booléens, puisqu'elle traite de la tendance d'une cellule à se différencier, à long terme, vers un tel ou un autre type cellulaire [9, 27]. Il est donc pertinent d'estimer et de comparer les distributions stationnaires du BN et de ses deux extensions.

À l'instar de [9, sect. 6.2], le résultat des  $R = 1000$  simulations est affiché dans une grille de dimensions 10x100 : chacune des cases est coloriée d'une couleur selon le point fixe atteint en fin de simulation. Cette méthode est une variante de l'algorithme 2 étudié en section 2.1.

Le STG en figure 2.4 indiquait des proportions de bassins d'attraction pour  $\{000\}$ ,  $\{011\}$ ,  $\{100\}$ ,  $\{110\}$  à respectivement  $\frac{1}{8}$ ,  $\frac{3}{8}$ ,  $\frac{1}{8}$ ,  $\frac{3}{8}$ . Les simulations du BN en figure 2.7a indiquent des fréquences d'atteinte similaire ; ce lien s'explique par ma proposition 2 démontrée en section 2.1.

Le comportement à long terme du modèle en fig. 2.7b écarte totalement l'attracteur 110 au profit de 100, et augmente la fréquence d'atteinte de 000 suite à l'élargissement de son bassin d'attraction large.

Quant à la dernière extension illustrée en fig. 2.7c, on constate que la moitié des états de départ sont dans le bassin d'attraction de 011, et l'autre moitié dans celui de 100, ce qui correspond bien au partitionnement de l'espace d'états en deux bassins d'attraction stricts et de même taille.



'00100000000000100000000'<sup>6</sup> est l'état où seuls les gènes IFNg (position 3) et STAT1 (position 15) sont exprimés, il sera noté [IFNg, STAT1].

Il a été constaté l'existence de 3 états stables pour ce réseau booléen [26] :

- $Th0$  : '0000000000000000000000' (aucun gène activé)
- $Th1$  : '00110000000001000010000' (activation de [IFNg, IFNgR, SOCS1, Tbet])
- $Th2$  : '10001100110000010100000' (activation de [GATA3, IL-10, IL-10R, IL-4, IL-4R, STAT3, STAT6])

On a une partition de l'espace d'états :

- $\mathcal{U}_0 = \{x \in \{0, 1\}^n \mid x_{Tbet} = 0 \wedge x_{GATA3} = 0\}$ , contenant l'état stable  $Th0$
- $\mathcal{U}_1 = \{x \in \{0, 1\}^n \mid x_{Tbet} = 1 \wedge x_{GATA3} = 0\}$ , contenant l'état stable  $Th1$
- $\mathcal{U}_2 = \{x \in \{0, 1\}^n \mid x_{Tbet} = 0 \wedge x_{GATA3} = 1\}$ , contenant l'état stable  $Th2$ .
- $\mathcal{U}_+ = \{x \in \{0, 1\}^n \mid x_{Tbet} = 1 \wedge x_{GATA3} = 1\}$ , dont chaque état est instable, puisque l'inhibition mutuelle de Tbet et GATA3 les empêche de cohabiter.

Les gènes Tbet et GATA3 sont considérés comme des *marqueurs* de la différenciation vers  $Th1$  et  $Th2$  respectivement. De plus, il a été observé [26] que l'expression des gènes IL4 et IL4R étaient cruciales dans la différenciation vers  $Th2$ .

Ce sont donc leurs fonctions de régulation que l'on va ici étendre, formant 4 PBN des fonctions voisines :

- pour GATA3 :  $(\neg Tbet \wedge GATA3) \vee (\neg Tbet \wedge STAT6)$   
vers son enfant  $\neg Tbet \wedge GATA3 \wedge STAT6$   
et son parent  $(\neg Tbet \wedge STAT6) \vee (\neg Tbet \wedge GATA3) \vee (GATA3 \wedge STAT6)$
- pour Tbet :  $\neg GATA3 \wedge (Tbet \vee STAT1)$   
vers son enfant  $\neg GATA3 \wedge Tbet \wedge STAT1$   
et son parent  $(\neg GATA3 \wedge Tbet) \vee (\neg GATA3 \wedge STAT1) \vee (Tbet \wedge STAT1)$
- pour IL4 :  $GATA3 \wedge \neg STAT1$   
vers son parent  $GATA3 \vee \neg STAT1$
- pour IL4R :  $IL4 \wedge \neg SOCS1$   
vers son parent  $IL4 \vee \neg SOCS1$

Les observations de cette sous-section porteront sur les probabilités d'activation de Tbet et GATA3 à long terme — c'est-à-dire d'atteinte de  $Th0$ ,  $Th1$ , ou  $Th2$  — dans chacun de ces réseaux (BN ou PBN des fonctions voisines), avec une mise à jour synchrone ou asynchrone.

Pour toutes les simulations de cette section, conformément aux expériences de [9, sect. 6.2], l'état initial  $x_0$  est toujours fixé sur [IFNg].

### Cas initial : le BN

Dans le cas du BN **synchrone**, la dynamique est déterministe. À partir de l'état [IFNg], on atteint [IFNgR], puis [JAK1], puis [STAT1], puis [SOCS1, Tbet]. Dès lors qu'on se trouve dans un état où Tbet est activé mais pas GATA3, Tbet s'autoactivera et bloquera l'activation de GATA3, le seul gène

6. L'ordre des gènes dans le vecteur d'activité est le suivant : GATA3, IFNbR, IFNg, IFNgR, IL10, IL10R, IL12R, IL18R, IL4, IL4R, IRAK, JAK1, NFAT, SOCS1, STAT1, STAT3, STAT4, STAT6, Tbet, IFNb, IL12, IL18, TCR.

pouvant inhiber Tbet. L'état stable atteint est Th1.

Pour étudier GATA3 et Tbet dans ce BN en mise à jour **asynchrone**, il est pertinent de borner le sous-espace des états accessibles. Les gènes autres que IFNg ne sont pas activés dans l'état initial, et pour qu'un le soit à une étape donnée, l'équation 2.4 indique qu'il est nécessaire qu'au moins un de ses activateurs le soit à l'étape précédente. Ainsi, seuls les gènes tels qu'il existe un chemin d'arcs activateurs depuis IFNg sont activables. Ils sont au nombre de six :  $\check{\mathcal{V}} = \{IFNg, IFNgR, JAK1, STAT1, SOCS1, Tbet\}$ .

Considérons le sous-espace des états  $\check{\mathcal{U}} = \{x \in \{0, 1\}^n \mid \forall i, x_i = 1 \Rightarrow g_i \in \check{\mathcal{V}}\}$ . Il inclut l'ensemble des états accessibles depuis  $x_0$ , et il est partitionnable en trois composantes :  $\check{\mathcal{U}}_0 = \{0\} \subset \mathcal{U}_0$ ,  $\check{\mathcal{U}}_1 = \{x \in \check{\mathcal{U}} \mid x_{Tbet} = 1\} \subset \mathcal{U}_1$ , et les autres états dans  $\check{\mathcal{U}}_* \subset \mathcal{U}_0$ . Les deux premières sont des composantes *absorbantes*, c'est-à-dire dont il n'y a pas de chemin pour en sortir une fois dedans. Pour  $\check{\mathcal{U}}_0$  : si aucun gène n'a d'activateur exprimé, aucun gène ne pourra s'activer au tour suivant (on reste dans  $\check{\mathcal{U}}_0$ ). Pour  $\check{\mathcal{U}}_1$  : l'autoactivation de Tbet et la non-expression de GATA3 suffisent à maintenir Tbet perpétuellement activé (on reste dans  $\check{\mathcal{U}}_1$ ).

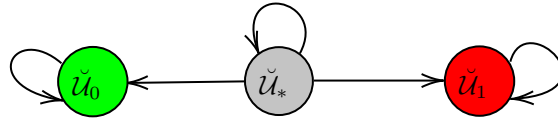


FIGURE 2.9 – Représentation des transitions possibles entre les ensembles d'états  $\check{\mathcal{U}}_0$ ,  $\check{\mathcal{U}}_1$ , et  $\check{\mathcal{U}}_*$ , dans le BN asynchrone du modèle Th\_23.

Dans le STG asynchrone restreint à  $\check{\mathcal{U}}_*$ , voici la composante des états accessibles depuis  $x_0$  :

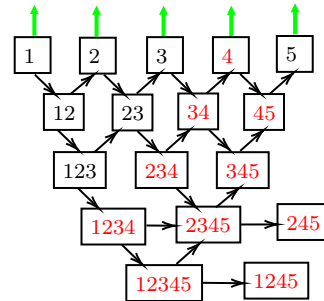


FIGURE 2.10 – STG asynchrone restreint.

Les états  $y$  sont notés selon leurs gènes activés : 1=IFNg, 2=IFNgR, 3=JAK1, 4=STAT1, 5=SOCS1. Toutes les transitions marquées par les flèches sont équiprobables ; les flèches vertes dirigent vers des états dans  $\check{\mathcal{U}}_0$ , car les états à un seul gène activé peuvent voir ce gène désactivé ; et les états notés en rouge ont une transition supplémentaire vers un état de  $\check{\mathcal{U}}_1$  (non-représentée pour des soucis de lisibilité), car STAT1 est un activateur de Tbet.

L'état initial  $x_0$  est noté "1" dans ce graphe, celui où seul IFNg est exprimé. Si une simulation termine dans  $\check{\mathcal{U}}_0$ , l'attracteur Th0 est atteint ; si elle termine dans  $\check{\mathcal{U}}_1$ , l'attracteur Th1 est atteint.

Par le même procédé qu'employé en figure 2.7, estimons empiriquement la prévalence de chaque attracteur, en mise à jour synchrone puis asynchrone :

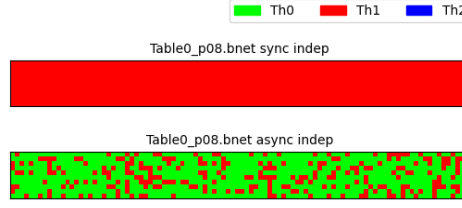


FIGURE 2.11 – Attracteurs atteints lors de 1000 simulations, pour le BN initial du modèle Th<sub>23</sub>.

0% / 100% / 0% - 74.3% / 25.7% / 0%

Les résultats sont cohérents avec ceux envisagés : la simulation synchrone termine toujours dans Th1, et la simulation asynchrone s'achève dans Th0 ou dans Th1. La première bifurcation à l'état initial "1" suffit à expliquer pourquoi la fréquence est plus élevée pour Th0 que pour Th1 : il y a une probabilité 1/2 de transiter directement vers l'état nul Th0 (flèche verte), et 1/2 de poursuivre dans  $\check{\mathcal{U}}_*$  ... Sur un grand nombre de simulations, environ la moitié finissent dans Th0 dès la première étape, et l'autre moitié a encore un risque de s'y rendre.

### PBN des fonctions voisines : GATA3

La fonction de référence de GATA3 est  $\neg Tbet \wedge (GATA3 \vee STAT6)$ , et ses voisins directs dans le diagramme de Hasse sont  $\neg Tbet \wedge GATA3 \wedge STAT6$ , et  $(\neg Tbet \wedge STAT6) \vee (\neg Tbet \wedge GATA3) \vee (GATA3 \wedge STAT6)$ .

Nous l'avons vu dans le cas précédent, STAT6 ne peut être activé en partant de  $x_0$  puisque n'appartient pas à  $\check{\mathcal{V}}$ . Or, tant que GATA3 et STAT6 sont désactivés, les trois fonctions de régulation de GATA3 renvoient une sortie 0. En termes d'états atteignables et de comportement à long terme, on se retrouve dans le même cas de figure que le BN initial.

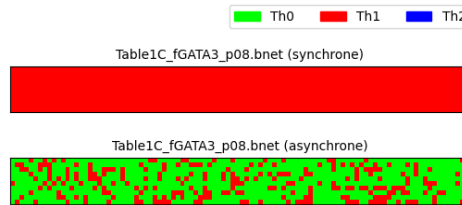


FIGURE 2.12 – Attracteurs atteints lors de 1000 simulations, pour le PBN des fonctions voisines de GATA3.

0% / 100% / 0% - 74.8% / 25.2% / 0%

Les résultats de simulation sont sensiblement identiques à la sous-section précédente.

### PBN des fonctions voisines : Tbet

La fonction de référence de Tbet est  $\neg GATA3 \wedge (Tbet \vee STAT1)$ , et ses voisins directs dans le diagramme de Hasse sont  $\neg GATA3 \wedge Tbet \wedge STAT1$ , et  $(\neg GATA3 \wedge Tbet) \vee (\neg GATA3 \wedge STAT1) \vee (Tbet \wedge STAT1)$ . Le gène GATA3 étant systématiquement désactivé pour des raisons mentionnées plus haut, ces trois fonctions sont respectivement équivalentes à  $Tbet \vee STAT1$ ,  $Tbet \vee STAT1$ , et  $Tbet \wedge STAT1$ .

Pour reprendre les notations entrevues en section 1.2 :  $\mathbb{T}(Tbet \wedge STAT1) \subseteq \mathbb{T}(Tbet \vee STAT1)$ . L'ajout des fonctions voisines augmente strictement la probabilité d'employer la fonction  $Tbet \wedge STAT1$ , moins prompte à activer le gène Tbet que la fonction de référence  $Tbet \vee STAT1$ .

Mais la nouvelle fonction  $Tbet \wedge STAT1$  a surtout une autre propriété cruciale : l'état Th1 (= [IFNg, IFNgR, SOCS1, Tbet]) n'est pas son point fixe, puisqu'il n'exprime pas STAT1. Si le réseau atteint Th1 et que la fonction  $Tbet \wedge STAT1$  est sélectionnée, alors Tbet est désactivé et on transite vers l'état nul [IFNg, IFNgR, SOCS1].

En **synchrone**, pour l'une ou l'autre fonction de Tbet, les états suivants sont [IFNgR], puis [JAK1], puis [STAT1]. Si  $Tbet \vee STAT1$  est sélectionnée, on revient sur [SOCS1, Tbet] pour ensuite revenir sur Th1 ; mais si  $Tbet \wedge STAT1$  est sélectionnée, on transitionne vers [SOCS1] puis vers l'état nul Th0, qui lui est bien stable par toutes les fonctions.

En **asynchrone**, comme on a pu le voir dans l'étude du BN asynchrone, il y a une probabilité assez importante d'avoir atteint Th0 avant Th1. Supposons que l'on ait atteint Th1 : la fonction  $Tbet \wedge STAT1$  est nécessairement appelée à une certaine étape, et désactive Tbet. On retourne dans le sous-espace  $\check{\mathcal{U}}_*$ , avec à nouveau deux issues : l'attracteur Th0, ou revenir à Th1.

Quelque soit le mode de mise à jour, dès lors que l'on se trouve dans Th1, on a une probabilité non-nulle d'atteindre l'attracteur stable Th0. Réadaptons l'illustration de la figure 2.9 :

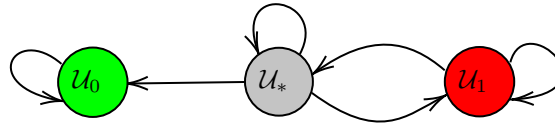


FIGURE 2.13 – Représentation des transitions possibles entre les ensembles d'états  $\check{\mathcal{U}}_0$ ,  $\check{\mathcal{U}}_1$ , et  $\check{\mathcal{U}}_*$ , dans le PBN des fonctions voisines de Tbet.

Il est attendu qu'à long terme, toutes les simulations finissent par rejoindre l'attracteur Th0.

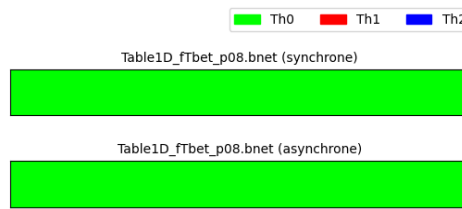


FIGURE 2.14 – Attracteurs atteints lors de 1000 simulations, pour le PBN des fonctions voisines de Tbet.  
100% / 0% / 0% - 100% / 0% / 0%

En incluant la possibilité de modifier une fonction de régulation légèrement différente pour Tbet, la dynamique du réseau a complètement changé : il est impossible de rejoindre l'attracteur Th1.



## PBN des fonctions voisines : IL4

La fonction de référence de IL4 est  $GATA3 \wedge \neg STAT1$ , et son voisin direct dans le diagramme de Hasse est  $GATA3 \vee \neg STAT1$ . L'activation du gène IL4 peut avoir lieu si elle est régulée par  $GATA3 \vee \neg STAT1$  et que le gène STAT1 est désactivé. Il pourra éventuellement s'en suivre une boucle d'activation sur les gènes  $\{IL4, IL4R, STAT6, GATA3, IL10, IL10R, STAT3\}$ .

En mise à jour **synchrone**, nous avons vu plus haut qu'il suffit de 4 étapes à partir de l'initiation (activation de IFNgR, puis JAK1, puis STAT1, puis Tbet) pour activer le gène Tbet, inhiber définitivement le gène GATA3, et ainsi stabiliser le système dans l'attracteur Th1. L'exception à ce scénario nécessite une activation de GATA3 dans les toutes premières étapes de la simulation.

Supposons que IL4 soit désactivé à l'étape 1 par  $GATA3 \wedge \neg STAT1$ , puis que la fonction  $GATA3 \vee \neg STAT1$  l'active à l'étape 2, avant que STAT1 ne soit activé. Le signal activateur n'arrivera à STAT6 (l'activateur direct de GATA3) qu'à l'étape 4, au même instant que l'activation de Tbet. La sortie de  $f_{GATA3} := \neg Tbet \wedge (GATA3 \vee STAT6)$  sera nulle par inhibition par Tbet, c'est-à-dire que GATA3 ne pourra être activée. Il en est de même si la première activation de IL4 a lieu aux étapes 3 ou plus tard.

Dans le cas contraire, supposons que IL4 soit activé dès la 1<sup>ère</sup> étape. Tbet et GATA3 sont alors activés en même temps à l'étape 4, respectivement par STAT1 et STAT6. Parce qu'ils sont mutuellement inhibiteurs l'un de l'autre, ils sont ensuite tous deux désactivés à l'issue de l'étape 5. Or dans la même étape 5, IFNg est réactivée par le signal de Tbet. Sauf inhibition extérieure, ce signal réactive Tbet quatre étapes plus tard, à l'étape 9. Pour éviter à l'étape 10 un second scénario comme celui-ci ou une inhibition définitive comme au paragraphe précédent, il faut que GATA3 soit réactivé à l'étape 8 ou moins, c'est-à-dire que IL4 soit réactivé par  $GATA3 \vee \neg STAT1$  à l'étape 5 ou moins.

Si IL4 se réactive à l'étape 5, la régulation du gène IL4R (le suivant dans la boucle) sera inhibée par le SOC5 activé par Tbet. Si IL4 tente de se réactiver à l'étape 4, la sortie de la fonction  $GATA3 \vee \neg STAT1$  est nulle, puisque le gène STAT1 est exprimé à ce moment-là.

Pour stabiliser vers l'attracteur Th2, il est donc nécessaire et suffisant que le gène IL4 soit activé dès l'étape 1 pour annuler Tbet, puis à l'étape 2 ou 3 pour réactiver GATA3 avant Tbet, et ainsi inhiber définitivement Tbet.

Soit  $p_{ref}$  la probabilité de tirer la fonction de référence ; celle de tirer  $GATA3 \vee \neg STAT1$  est de  $1 - p_{ref}$  à chaque tour. Notons  $A_k$  l'événement " $GATA3 \vee \neg STAT1$  est tirée à l'étape  $k$ ".

$$\begin{aligned}
 & P(A_1 \wedge (A_2 \vee A_3)) \\
 &= P(A_1 \wedge (A_2 \vee (\neg A_2 \wedge A_3))) \\
 &= P(A_1) \cdot (P(A_2) + P(\neg A_2 \wedge A_3)) \\
 &= (1 - p_{ref}) \cdot (1 - p_{ref} + p_{ref}(1 - p_{ref})) \\
 &= (1 - p_{ref})(1 - p_{ref}^2)
 \end{aligned}$$

Mes simulations de cette section utilisent un facteur  $p_{ref} = 0.8$ , cette probabilité vaut 7.2%. On s'attend à une telle fréquence de Th2, et le reste de Th1.

Lors des premières étapes de la mise à jour **asynchrone**, deux boucles de régulation fonctionnent en parallèle : celui induit par  $\tilde{\mathcal{V}} = \{IFNg, IFNgR, JAK1, STAT1, SOCS1, Tbet\}$ , et celui induit par  $\mathcal{V}' = \{IL4, IL4R, STAT6, GATA3\}$ .

Négligeons dans ce paragraphe les inhibitions d'une boucle vers l'autre. La première boucle (celle de  $\check{V}$ ), déjà étudiée avec le BN asynchrone (cf. fig. 2.11), peut s'achever de 2 manières : une désactivation de tous les gènes de  $\check{V}$ , ou une activation définitive de Tbet. La seconde boucle (celle de  $\check{V}'$ ), parce qu'IL4 peut se faire réactiver par  $GATA3 \vee \neg STAT1$  à toute étape, termine presque sûrement par une activation de GATA3.

Prenons maintenant en compte les inhibitions d'une boucle vers l'autre. De  $\check{V}$  vers  $\check{V}'$ , l'expression de STAT1 inhibe IL4, et l'expression de SOCS1 inhibe IL4R. Mais les inhibitions à principalement considérer sont celles-ci : une activation de Tbet stabilise le réseau vers Th1, et inhibe tous les gènes de  $\check{V}'$  ; une activation de GATA3 stabilise le réseau vers Th2, et inhibe tous les gènes de  $\check{V}$ .

L'attracteur atteint dépend donc de l'événement ayant lieu en premier parmi ces trois-là :

- *désactivation des gènes de  $\check{V}$*  : les gènes de  $\check{V}'$  n'ont plus de risques d'être inhibés. IL4 se réactive occasionnellement jusqu'à ce que GATA3 soit activé  $\Rightarrow$  Th2.
- *activation de Tbet* : les gènes de  $\check{V}'$  sont inhibés  $\Rightarrow$  Th1.
- *activation de GATA3* : les gènes de  $\check{V}$  sont inhibés  $\Rightarrow$  Th2.

Parce qu'il a généralement lieu plus tard que les deux premiers, on néglige le troisième scénario. Les deux cas restants sont ceux rencontrés lors de l'étude du BN asynchrone. Les fréquences de l'un et l'autre avaient été respectivement estimées proches de  $3/4$  et  $1/4$  ; ce sont donc les fréquences attendues ici pour Th2 et Th1.

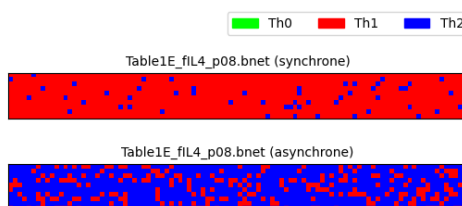


FIGURE 2.15 – Attracteurs atteints lors de 1000 simulations, pour le PBN des fonctions voisines de IL4.

0% / 94.7% / 5.3% - 0% / 24.2% / 75.8%

### PBN des fonctions voisines : IL4R

La fonction de référence de IL4R est  $IL4 \wedge \neg SOCS1$ , et son voisin direct dans le diagramme de Hasse est  $IL4 \vee \neg SOCS1$ . Le raisonnement de cette section est grandement analogue à celui consacré à IL4.

En mise à jour **synchrone**, l'indicateur clé est le temps de première activation du gène IL4R par  $IL4 \vee \neg SOCS1$ .

Si elle a lieu à l'étape 1, GATA3 est activée à l'étape 3 c'est-à-dire une étape avant Tbet, inhibe Tbet, et stabilise le réseau à l'attracteur Th2. Si elle a lieu à l'étape 3 ou plus, GATA3 reçoit simultanément une activation depuis STAT6 et une inhibition depuis Tbet, il ne peut être activé et le réseau se stabilise à l'attracteur Th1.

Si elle a lieu à l'étape 2, GATA3 et Tbet s'inhibent mutuellement à l'étape 5, comme vu dans le cas synchrone consacré à IL4. Parce que Tbet est réactivé en étape 9, GATA3 doit être réactivé à l'étape 8 ou moins pour ne pas être définitivement inhibé ; cela revient à une réactivation de IL4R à l'étape 6 ou moins. La fonction  $IL4 \vee \neg SOCS1$  est de sortie nulle à l'étape 6, car SOCS1 y est exprimé. En revanche,

une sélection de cette fonction aux étapes 3, 4 ou 5 permet bien d'activer IL4R puis STAT6 puis GATA3 à temps.

Pour stabiliser vers l'attracteur Th2, il est donc nécessaire et suffisant que le gène IL4R soit activé à l'étape 1 ; ou bien une première fois l'étape 2 pour annuler Tbet, et une seconde fois à l'étape 3 4 ou 5 pour réactiver GATA3 avant Tbet, et ainsi inhiber définitivement Tbet.

Notons  $B_k$  l'événement " $IL4 \vee \neg SOCS1$  est tirée à l'étape  $k$ ".

$$\begin{aligned}
& P(B_1 \vee (B_2 \wedge (B_3 \vee B_4 \vee B_5))) \\
&= P(B_1 \vee ([\neg B_1 \wedge B_2] \wedge [B_3 \vee (\neg B_3 \wedge B_4) \vee (\neg B_3 \wedge \neg B_4 \wedge B_5)]))) \\
&= (1 - p_{ref}) + p_{ref}(1 - p_{ref})^2(1 + p_{ref} + p_{ref}^2) \\
&= (1 - p_{ref}) + p_{ref}(1 - p_{ref})(1 - p_{ref}^3) \\
&= (1 - p_{ref})(1 + p_{ref} - p_{ref}^4)
\end{aligned}$$

Mes simulations de cette section utilisent un facteur  $p_{ref} = 0.8$ , cette probabilité vaut 27.8%. On s'attend à une proche fréquence de Th2, et le reste de Th1.

Le comportement **asynchrone** de ce réseau est très proche du réseau asynchrone associé à IL4 vu précédemment. On peut lui appliquer les mêmes justifications, et supposer de même une fréquence de  $1/4$  de Th1 et  $3/4$  de Th2.

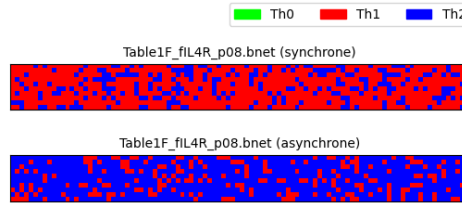


FIGURE 2.16 – Attracteurs atteints lors de 1000 simulations, pour le PBN des fonctions voisines de IL4R.

0% / 74.0% / 26.0% - 0% / 22.8% / 72.2%

Dans cette dernière sous-section, j'ai articulé et détaillé le lien de cause à effet entre un changement de paramètre (la fonction) et une altération des propriétés dynamiques (les attracteurs atteints).

On a pu constater que le choix de mise à jour, tout comme la modification des fonctions de régulation d'un gène, peuvent avoir un impact fort sur l'atteignabilité de certains attracteurs, c'est-à-dire ici sur l'activation à long terme de certains gènes marqueurs (ici GATA3 et Tbet). Parmi les résultats les plus contre-intuitifs : dans les exemples étudiés, l'expression à long terme de GATA3 est inchangée par une modification des fonctions de régulation de GATA3, mais l'est (Th2 devient atteignable) lorsque l'on étend celles d'IL4 ou IL4R.

# Conclusion

Mon travail s'est réparti tant sur des domaines théoriques qu'expérimentaux. L'algorithme calculant la distribution stationnaire d'un réseau était préalablement admis comme identifiant les attracteurs ; j'ai établi de nouveaux résultats le concernant afin d'estimer la prévalence des attracteurs sans nécessiter le calcul de toutes les transitions.

De plus, à l'aide d'algorithmes de génération aléatoire et de construction de modèles booléens incluant les fonctions consistantes, j'ai constaté et analysé plusieurs types d'altération sur la dynamique à long terme que pouvait engendrer une perturbation des fonctions de régulation, dans un exemple synthétique et dans un exemple biologique.

L'étude du modèle synthétique montrait ici un cas où l'extension vers les fonctions de régulation voisines menait à une suppression d'un point fixe. Il reste à s'intéresser à d'autres cas possibles : plusieurs points fixes (ou plusieurs attracteurs cycliques) vers un attracteur cyclique, un point fixe s'étendant à un cycle... De plus, certains paramètres de simulation comme le facteur de perturbation  $p$  et le facteur de changement de contexte  $q$  n'ont été que brièvement manipulés au cours des expériences, leur variation engendrerait possiblement des résultats hybrides entre le cas de base (BN) et le cas probabiliste (PBN), voire inenvisagés.

L'élaboration de BNs à partir d'un graphe de régulation signé s'est jusqu'ici faite par une seule méthode de construction, garantissant la même fonction monotone dans un PO-Set donné. L'étude des fonctions booléennes monotones, dont le champ de recherche a bénéficié encore cette année de grandes avancées, pourrait permettre de générer des BN comportant d'autres fonctions monotones.

Enfin, un prolongement plus appliqué de ce travail pourrait s'orienter vers l'analyse de données biologiques à l'aide des outils développés, ou vers le raffinement de classes de modèles existantes afin de figurer des régulations biologiques sous-jacentes jusqu'ici peu considérées.

# Bibliographie

- [1] Stuart Alan Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3) :437–467, 1969. doi:[10.1016/0022-5193\(69\)90015-0](https://doi.org/10.1016/0022-5193(69)90015-0).
- [2] Ilya Shmulevich, Edward R. Dougherty, and Wei Zhang. Gene perturbation and intervention in Probabilistic Boolean Networks. *Bioinformatics*, 18(04) :1319–31, 11 2002. doi:[10.1093/bioinformatics/18.10.1319](https://doi.org/10.1093/bioinformatics/18.10.1319).
- [3] Marcel Brun, Edward R. Dougherty, and Ilya Shmulevich. Steady-state probabilities for attractors in probabilistic boolean networks. *Signal Processing*, 85(10) :1993–2013, 2005. doi:[10.1016/j.sigpro.2005.02.016](https://doi.org/10.1016/j.sigpro.2005.02.016).
- [4] Yufei Xiao and Edward R. Dougherty. The impact of function perturbations in boolean networks. *Bioinformatics*, 23(10) :1265–1273, 2007. doi:[10.1093/bioinformatics/btm093](https://doi.org/10.1093/bioinformatics/btm093).
- [5] Ilya Shmulevich, Edward R. Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks : A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2) : 261–274, 02 2002. doi:[10.1093/bioinformatics/18.2.261](https://doi.org/10.1093/bioinformatics/18.2.261).
- [6] Ilya Shmulevich and Edward R. Dougherty. *Probabilistic Boolean Networks - The Modeling and Control of Gene Regulatory Networks*. SIAM, 2010. ISBN 978-0-89871-692-4. doi:[10.5555/1734075](https://doi.org/10.5555/1734075).
- [7] Seungchan Kim, Edward Dougherty, Yidong Chen, Michael Bittner, and Edward Suh. Can Markov Chain Models Mimic Biological Regulation? *Journal of Biological Systems*, 10(4) :337–357, 2002. doi:[10.1142/S0218339002000676](https://doi.org/10.1142/S0218339002000676).
- [8] Ilya Shmulevich, Ilya Gluhovsky, Ronaldo F. Hashimoto, Edward R. Dougherty, and Wei Zhang. Steady-state analysis of genetic regulatory networks modelled by probabilistic boolean networks. *Comparative and Functional Genomics*, 4(6) :601–608, 2003. doi:[10.1002/cfg.342](https://doi.org/10.1002/cfg.342).
- [9] José E. R. Cury, Pedro T. Monteiro, and Claudine Chaouiya. Partial Order on the set of Boolean Regulatory Functions. *arXiv*, 2019. doi:[10.48550/arXiv.1901.07623](https://doi.org/10.48550/arXiv.1901.07623).
- [10] Madalena Chaves, Réka Albert, and Eduardo D. Sontag. Robustness and fragility of Boolean models for genetic regulatory networks. *Journal of Theoretical Biology*, 235(3) :431–449, 2005. doi:[10.1016/j.jtbi.2005.01.023](https://doi.org/10.1016/j.jtbi.2005.01.023).
- [11] Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14) :124–131, 2006. doi:[10.1093/bioinformatics/btl210](https://doi.org/10.1093/bioinformatics/btl210).

- [12] Nuno D. Mendes, Rui Henriques, Elisabeth Remy, Jorge Carneiro, Pedro T. Monteiro, and Claudine Chaouiya. Estimating Attractor Reachability in Asynchronous Logical Models. *Frontiers in Physiology*, 9, 2018. doi:[10.3389/fphys.2018.01161](https://doi.org/10.3389/fphys.2018.01161).
- [13] E. Remy and C. Chaouiya. Logical Modelling of Regulatory Networks, methods and applications. *Bulletin of Mathematical Biology*, 75(6) :891–895, 2013. doi:[10.1007/s11538-013-9863-0](https://doi.org/10.1007/s11538-013-9863-0).
- [14] W. Abou-Jaoudé, P. Traynard, P.T. Monteiro, J. Saez-Rodriguez, T. Helikar, D. Thieffry, and C. Chaouiya. Logical Modeling and Dynamical Analysis of Cellular Networks. *Frontiers in Genetics*, 7(94), 2016. doi:[10.3389/fgene.2016.00094](https://doi.org/10.3389/fgene.2016.00094).
- [15] D. Bérenguier, C. Chaouiya, P. T. Monteiro, A. Naldi, and E. Remy. Dynamical Modeling and Analysis of Large Cellular Regulatory Networks. *Chaos*, 23(2) :891–895, 2013. doi:[10.1063/1.4809783](https://doi.org/10.1063/1.4809783).
- [16] E. Remy, P. Ruet, L. Mendoza, D. Thieffry, and C. Chaouiya. From Logical Regulatory Graphs to Standard Petri Nets : Dynamical Roles and Functionality of Feedback Circuits. *Transactions on Computational Systems Biology - TCSB*, 4230 :56–72, 2006. doi:[10.1007/11905455\\_3](https://doi.org/10.1007/11905455_3).
- [17] Archie Blake. *Canonical expressions in Boolean algebra*. PhD thesis, The University of Chicago, 1937. URL <https://www.proquest.com/openview/2f92469c3a02402fbb717e0ee3b5d569/>.
- [18] Erhan Çinlar. *Introduction to stochastic processes*. Pearson College, 1997.
- [19] Joseph Xu Zhou, Areejit Samal, Aymeric Fouquier d’Hérouël, Nathan D. Price, and Sui Huang. Relative stability of network states in Boolean Network models of gene regulation in development. *BioSystems*, 142-143 :15–24, 2016. doi:[10.1016/j.biosystems.2016.03.002](https://doi.org/10.1016/j.biosystems.2016.03.002).
- [20] Jae Il Joo, Joseph X. Zhou, Sui Huang, and Kwang-Hyun Cho. Determining Relative Dynamic Stability of Cell States Using Boolean Network Model. *Scientific Reports*, 8, 2018. doi:[10.1038/s41598-018-30544-0](https://doi.org/10.1038/s41598-018-30544-0).
- [21] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Society, 1997.
- [22] Ajay Subbaroyan, Priyotosh Sil, Olivier C. Martin, and Areejit Samal. Leveraging Developmental Landscapes for Model Selection in Boolean Gene Regulatory Networks. *Briefings in Bioinformatics*, 24(3), 2023. doi:[10.1093/bib/bbad160](https://doi.org/10.1093/bib/bbad160).
- [23] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*, pages 121–122. Princeton University Press, 1994. doi:[10.2307/j.ctv182jsw5](https://doi.org/10.2307/j.ctv182jsw5).
- [24] J.R. Norris. *Markov Chains*. Cambridge University Press, 1997. ISBN 978-0521633963. doi:[10.1017/CBO9780511810633](https://doi.org/10.1017/CBO9780511810633).
- [25] Stein Emil Vollset. Confidence intervals for a binomial proportion. *Statistics in Medicine*, 12(9) : 809–824, 1993. doi:[10.1002/sim.4780120902](https://doi.org/10.1002/sim.4780120902).
- [26] Luis Mendoza and Ioannis Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3, 2006. doi:[10.1186/1742-4682-3-13](https://doi.org/10.1186/1742-4682-3-13).

- [27] Abhishek Garg, Kartik Mohanram, Alessandro Di Cara, Giovanni De Micheli, and Ioannis Xenarios. Modeling stochasticity and robustness in gene regulatory networks. *Bioinformatics*, 25(12) : i101–i109, 2009. doi:[10.1093/bioinformatics/btp214](https://doi.org/10.1093/bioinformatics/btp214).

## **Déclaration contre le plagiat**

Je soussigné ASSOUMANI Zachary, régulièrement inscrit à l'Université de Lorraine en année universitaire 2022-2023 en 2ème année de Master de Mathématiques, certifie qu'il s'agit d'un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité. Je certifie, de surcroît, que je n'ai ni recopié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Conformément à la loi, le non-respect de ces dispositions me rend passible de poursuites devant la commission disciplinaire et les tribunaux de la République Française.

Fait à Marseille, le 15 septembre 2023.



## Annexe A

### Preuve de la formule en section 1.4.2

La formule est la suivante, pour un PBN  $A^{p,q}(V, F, C)$ . Pour tous contextes  $\mathbf{f}_1, \mathbf{f}_2 \in F$  et vecteurs d'activité  $\mathbf{x}_1, \mathbf{x}_2 \in \{0, 1\}^n$ , la probabilité de transition entre  $(\mathbf{f}_1, \mathbf{x}_1)$  et  $(\mathbf{f}_2, \mathbf{x}_2)$  est :

$$\mathbb{P}((\mathbf{f}_2, \mathbf{x}_2)|(\mathbf{f}_1, \mathbf{x}_1)) = [(1 - q)\mathbb{1}_{\mathbf{f}_1=\mathbf{f}_2} + qc_2] \times [(1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2} + (1 - p)^{n-\eta(\mathbf{x}_1, \mathbf{x}_2)} p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2}]$$

où  $\eta(x, y)$  désigne la distance de Hamming entre  $x$  et  $y$ .

---

*Preuve.* Remarquons, dans la figure 1.4.1, que la modification du contexte  $\mathbf{f}$  et de la configuration  $\mathbf{x}$  se font successivement. On a donc :

$$\mathbb{P}((\mathbf{f}_2, \mathbf{x}_2)|(\mathbf{f}_1, \mathbf{x}_1)) = \mathbb{P}(\mathbf{f}_2|\mathbf{f}_1, \mathbf{x}_1) \times \mathbb{P}(\mathbf{x}_2|\mathbf{f}_2, \mathbf{x}_1) \quad (\text{A.1})$$

$\xi$  est une variable de Bernoulli de paramètre  $q$ . De la nullité de  $\xi$  résulte le non-changement de contexte, et de la non-nullité de  $\xi$  résulte la sélection d'un nouveau contexte  $\mathbf{f}_i$  avec une probabilité  $c_i, \forall \mathbf{f}_i \in F$ . Par la formule des probabilités totales, on obtient :

$$\begin{aligned} \mathbb{P}(\mathbf{f}_2|\mathbf{f}_1, \mathbf{x}_1) &= \mathbb{P}(\xi = 0)\mathbb{P}(\mathbf{f}_2|\mathbf{f}_1, \mathbf{x}_1, \xi = 0) + \mathbb{P}(\xi = 1)\mathbb{P}(\mathbf{f}_2|\mathbf{f}_1, \mathbf{x}_1, \xi = 1) \\ &= (1 - q)\mathbb{1}_{\mathbf{f}_1=\mathbf{f}_2} + qc_2 \end{aligned} \quad (\text{A.2})$$

L'atteinte de  $\mathbf{x}_2$  à partir de  $(\mathbf{f}_2, \mathbf{x}_1)$  peut se faire via deux événements disjoints : un appel de la fonction  $\mathbf{f}_2$ , ou une perturbation. On se retrouve dans l'un ou l'autre événement selon la nullité ou non du vecteur de perturbation  $\gamma$ , variable de loi de Bernoulli  $n$ -dimensionnelle.

D'après la formule des probabilités totales,

$$\mathbb{P}(\mathbf{x}_2|\mathbf{f}_2, \mathbf{x}_1) = \mathbb{P}(\mathbf{x}_2, \gamma = 0|\mathbf{f}_2, \mathbf{x}_1) + \mathbb{P}(\mathbf{x}_2, \gamma \neq 0|\mathbf{f}_2, \mathbf{x}_1) \quad (\text{A.3})$$

Dans le cas appel de la fonction :

$$\begin{aligned} \mathbb{P}(\mathbf{x}_2, \gamma = 0|\mathbf{f}_2, \mathbf{x}_1) &= \mathbb{P}(\gamma = 0)\mathbb{P}(\mathbf{x}_2|\mathbf{f}_2, \mathbf{x}_1, \gamma = 0) \\ &= (1 - p)^n \mathbb{P}(\mathbf{x}_2|\mathbf{f}_2, \mathbf{x}_1, \text{appel de la fonction}) \\ &= (1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2}, \end{aligned} \quad (\text{A.4})$$

puisque'un appel de la fonction  $\mathbf{f}_2$  sur la configuration  $\mathbf{x}_1$ , en mise à jour synchrone, envoie nécessairement la configuration sur  $\mathbf{f}_2(\mathbf{x}_1)$ .

Dans le cas perturbation :

$$\mathbb{P}(\mathbf{x}_2, \gamma \neq 0|\mathbf{f}_2, \mathbf{x}_1) = \mathbb{P}(\mathbf{x}_2, \gamma \neq 0|\mathbf{f}_2, \mathbf{x}_1)(\mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} + \mathbb{1}_{\mathbf{x}_1 = \mathbf{x}_2})$$

Si  $\gamma \neq 0$ , on a nécessairement  $\mathbf{x}_1^\gamma \neq \mathbf{x}_1$ , d'où  $\mathbb{P}(\mathbf{x}_2, \gamma \neq 0 | \mathbf{f}_2, \mathbf{x}_1) \mathbb{1}_{\mathbf{x}_1 = \mathbf{x}_2} = 0$ . Ainsi :

$$\mathbb{P}(\mathbf{x}_2, \gamma \neq 0 | \mathbf{f}_2, \mathbf{x}_1) = \mathbb{P}(\mathbf{x}_2, \gamma \neq 0 | \mathbf{f}_2, \mathbf{x}_1) \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2}$$

Avec  $\mathbf{x}_1 \neq \mathbf{x}_2$ , les composantes à perturber sont exactement celles différant entre  $\mathbf{x}_1$  et  $\mathbf{x}_2$ , qui sont au nombre de  $\eta(\mathbf{x}_1, \mathbf{x}_2)$ .

$$\begin{aligned} \mathbb{P}(\mathbf{x}_2, \gamma \neq 0 | \mathbf{f}_2, \mathbf{x}_1) \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} &= \left( \bigcap_{\substack{1 \leq i \leq n \\ \mathbf{x}_{1,i} = \mathbf{x}_{2,i}}} \mathbb{P}(\gamma_i = 0) \times \bigcap_{\substack{1 \leq i \leq n \\ \mathbf{x}_{1,i} \neq \mathbf{x}_{2,i}}} \mathbb{P}(\gamma_i = 1) \right) \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} \\ &= p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} (1 - p)^{n - \eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2}. \end{aligned} \tag{A.5}$$

L'inclusion de A.4 et A.5 dans A.3 donne :

$$\mathbb{P}(\mathbf{x}_2 | \mathbf{f}_2, \mathbf{x}_1) = (1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1) = \mathbf{x}_2} + p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} (1 - p)^{n - \eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} \tag{A.6}$$

Cette dernière équation, inclus avec A.2 dans A.1, nous fournit le résultat attendu. ■

## Annexe B

# Irréductibilité de la chaîne de Markov pour un PBN tel que $p > 0$

### B.1 Termes nuls de la matrice de transition

Pour tous contextes  $\mathbf{f}_1, \mathbf{f}_2 \in F$  et vecteurs d'activité  $\mathbf{x}_1, \mathbf{x}_2 \in \{0, 1\}^n$ , la probabilité de transition entre  $(\mathbf{f}_1, \mathbf{x}_1)$  et  $(\mathbf{f}_2, \mathbf{x}_2)$  est :

$$\mathbb{P}((\mathbf{f}_2, \mathbf{x}_2)|(\mathbf{f}_1, \mathbf{x}_1)) = [(1 - q)\mathbb{1}_{\mathbf{f}_1=\mathbf{f}_2} + qc_2] \times [(1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2} + (1 - p)^{n-\eta(\mathbf{x}_1, \mathbf{x}_2)} p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2}]$$

Ce terme est celui de la matrice de transition d'une chaîne de Markov représentant un PBN, sur l'espace d'états  $F \times \{0, 1\}^n$ . On cherche à identifier les cas où ce terme est nul.

Il est produit de deux expressions. Pour la première,  $(1 - q)\mathbb{1}_{\mathbf{f}_1=\mathbf{f}_2} + qc_2 \geq qc_2 > 0$ . C'est donc de la seconde dont il va falloir déterminer la nullité.

$$\begin{aligned} & (1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2} + (1 - p)^{n-\eta(\mathbf{x}_1, \mathbf{x}_2)} p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} = 0 \\ \Leftrightarrow & \begin{cases} (1 - p)^n \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2} = 0 \\ (1 - p)^{n-\eta(\mathbf{x}_1, \mathbf{x}_2)} p^{\eta(\mathbf{x}_1, \mathbf{x}_2)} \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} = 0 \end{cases} \Leftrightarrow \begin{cases} \mathbb{1}_{\mathbf{f}_2(\mathbf{x}_1)=\mathbf{x}_2} = 0 \\ \mathbb{1}_{\mathbf{x}_1 \neq \mathbf{x}_2} = 0 \end{cases} \Leftrightarrow \begin{cases} \mathbf{f}_2(\mathbf{x}_1) \neq \mathbf{x}_2 \\ \mathbf{x}_1 = \mathbf{x}_2 \end{cases} \end{aligned}$$

La probabilité de transition  $\mathbb{P}((\mathbf{f}_2, \mathbf{x}_2)|(\mathbf{f}_1, \mathbf{x}_1))$  est donc nulle ssi  $\mathbf{x}_1 = \mathbf{x}_2$  et  $\mathbf{f}_2(\mathbf{x}_1) \neq \mathbf{x}_1$ . ■

### B.2 Irréductibilité de la chaîne

Nous allons montrer que dans la chaîne de Markov définie par un PBN  $A^{p,q}(V, F, C)$ , pour tous états  $(f, x)$  et  $(g, y) \in \{0, 1\}^n \times F$ , l'état  $(g, y)$  est accessible depuis  $(f, x)$ .

· Cas 1 :  $x \neq y$  ou  $g(x) = x$

D'après le résultat de la section précédente,  $\mathbb{P}((g, y)|(f, x)) > 0$ , donc  $(g, y)$  est accessible depuis  $(f, x)$  en une seule étape.

· Cas 2 :  $x = y$  et  $g(x) \neq x$

Puisque  $n \geq 1$ , il existe  $z \in \{0, 1\}^n$  différent de  $x$ . Puisque  $z \neq x$ , on a  $\mathbb{P}((g, z)|(f, x)) > 0$ . De plus,  $\mathbb{P}((g, x)|(g, z)) > 0$ . Donc la probabilité  $P_2$  de rejoindre en deux itérations  $(g, x)$  à partir de  $(f, x)$  vérifie

$$P_2 \geq \mathbb{P}((g, z)|(f, x)) \cdot \mathbb{P}((g, x)|(g, z)) > 0$$

Donc il y a une probabilité non-nulle d'accéder à  $(g, x)$  depuis  $(f, x)$ . ■

## Annexe C

# Expression analytique de la probabilité stationnaire d'un attracteur [3]

### C.1 Pour un BN perturbé synchrone

Pour  $x, y \in \{0, 1\}^n$ , on note  $P_y^*(x)$  la probabilité qu'une perturbation déplace la configuration  $x$  vers la configuration  $y$ . Explicitement,

$$P_y^*(x) := P(y|x, \text{perturbation}) = \mathbb{1}_{x \neq y} \frac{p^{\eta(x,y)} (1-p)^{n-\eta(x,y)}}{1 - (1-p)^n}$$

On suppose connus les attracteurs  $A_1, \dots, A_r$  et leurs bassins d'attraction  $B_1, \dots, B_r$ . On définit la matrice  $C \in \mathcal{M}_{r,2}(\mathbb{R})$  :

$$\forall 1 \leq i, k \leq r, \quad C_{ik} = \frac{1}{|A_i|} \sum_{x \in B_k} \sum_{y \in A_i} P_y^*(x)$$

L'hypothèse faite pour l'approximation est d'avoir  $p \ll 1$ , de manière que le temps entre deux perturbations soit assez long pour laisser le système atteindre un attracteur avant chaque perturbation. Les probabilités stationnaires des bassins approximent alors le vecteur stationnaire de la matrice stochastique  $C$ , c'est-à-dire

$$\pi(B_k) \approx \sum_{i=1}^r C_{ik} \pi(B_i)$$

Enfin, en notant  $\delta(x, A_k)$  le nombre d'itérations par  $f$  nécessaires pour atteindre  $A_k$  depuis un état  $x \in B_k$ , et  $b = (1-p)^n$ , la probabilité stationnaire d'un attracteur  $A_k$  est approchée par

$$\pi(A_k) \approx \sum_{i=1}^r \frac{1}{|A_i|} \left[ \sum_{x \in B_k} \sum_{y \in A_i} P_y^*(x) b^{\delta(x, A_k)} \right] \pi(B_i).$$

## C.2 Pour un PBN perturbé synchrone

Pour  $x, y \in \{0, 1\}^n$  et  $f, g \in F$ , on note  $P_{y,g}^*(x, f)$  la probabilité de se "réinitialiser", à partir de l'état  $y$  et du contexte  $g$ , vers l'état  $x$  et le contexte  $f$ . Une réinitialisation est l'occurrence d'un changement de contexte ( $\xi = 1$ ), d'une perturbation ( $\gamma \neq 0$ ), ou des deux simultanément.

$$P_{y,g}^*(x, f) := P(x, f | y, g, \text{réinitialisation}) = \sum_{A \in \{\gamma=0 \wedge \xi=1, \gamma \neq 0 \wedge \xi=0, \gamma \neq 0 \wedge \xi=1\}} \mathbb{P}(A) P_{y,g,A}^*(x, f)$$

Événement $A$	$\mathbb{P}(A)$	$P_{y,g,A}^*(x, f)$
$\gamma = 0 \wedge \xi = 1$	$(1-p)^n q$	$c_g \cdot \mathbb{1}_{f(y)=x}$
$\gamma \neq 0 \wedge \xi = 0$	$(1 - (1-p)^n)(1-q)$	$\frac{p^{eta(x,y)}(1-p)^{n-\eta(x,y)}}{1-(1-p)^n} \cdot \mathbb{1}_{x \neq y} \cdot \mathbb{1}_{f=g}$
$\gamma \neq 0 \wedge \xi = 1$	$(1 - (1-p)^n)q$	$\frac{p^{eta(x,y)}(1-p)^{n-\eta(x,y)}}{1-(1-p)^n} \cdot c_g \cdot \mathbb{1}_{x \neq y}$

TABLE C.1 – Table des probabilités conditionnelles de transition, pour les trois cas de la formule ci-dessus.

Pour chaque contexte  $f_k \in F$ , on suppose connus ses attracteurs  $A_{k1}, \dots, A_{kr_k}$  et leurs bassins d'attraction  $B_{k1}, \dots, B_{kr_k}$ . Soit  $N = \sum_k r_k$  le nombre total d'attracteurs sur tous les contextes, on définit la matrice  $C \in \mathcal{M}_{N^2}(\mathbb{R})$  :

$$\forall 1 \leq k, l \leq m, 1 \leq i \leq r_k, 1 \leq v \leq r_l, \quad C_{ki,lv} = \frac{1}{|A_{lv}|} \sum_{x \in B_{ki}} \sum_{y \in B_{lv}} P_{y,fl}^*(x, f_k)$$

Avec la même approximation  $p \ll 1$ , les probabilités stationnaires des couples bassins-contextes approximent alors le vecteur stationnaire de la matrice stochastique  $C$ , c'est-à-dire

$$\pi(B_{ki}, f_k) \approx \sum_{l=1}^m \sum_{v=1}^{r_l} C_{ki,lv} \pi(B_{lv}, f_l)$$

En notant  $\delta(x, A_{ki})$  le nombre d'itérations par  $f_k$  nécessaires pour atteindre  $A_{ki}$  depuis un état  $x \in B_{ki}$ , et  $b = (1-q)(1-p)^n$ , la probabilité stationnaire d'un couple attracteur-contexte  $(A_{ki}, f_k)$  est approchée par

$$\pi(A_{ki}, f_k) \approx \sum_{l=1}^m \sum_{v=1}^{r_l} \frac{1}{|A_{lv}|} \left[ \sum_{x \in B_{ki}} \sum_{y \in B_{lv}} P_{y,fl}^*(x, f_k) b^{\delta(x, A_{ki})} \right] \pi(B_{lv}, f_l)$$

On peut finalement approcher

$$\pi(A_{ki}) \approx \sum_{l=1}^m \sum_{j=1}^{r_l} \frac{|A_{ki} \cap A_{lj}|}{|A_{lj}|} \pi(A_{lj}, f_l).$$

## Annexe D

# Documentation de l'outil informatique développé

Les outils logiciels présentés plus haut permettent de construire, simuler, analyser différentes classes de réseaux booléens, mais restent limités en ce qui concerne les réseaux booléens probabilistes et les perturbations. Au cours de mon stage, j'ai ainsi développé un programme informatique en langage PYTHON, afin de construire des réseaux booléens selon des paramètres déjà considérés dans la littérature, mais aussi selon de nouveaux paramètres comme décrits dans ce rapport.

Le code source est disponible au lien suivant : <https://github.com/K4RI/pbn-simulation/>.

### D.1 Attributs de la classe PBN

ATTRIBUTS - MODÈLE :

- *title* (str) : nom attribué au PBN.
- *n* (int) : nombre de gènes
- *F* (list) : fonctions de régulation
- *c* (list) : distribution de probabilité sur les différentes fonctions
- *indep* (bool) : indépendance du PBN.

Si True, *F* est une liste de contextes et *c* est une liste de coefficients de probabilité.

Si False, *F* est de la forme  $F_1, \dots, F_n$  avec  $F_i$  les fonctions de régulation du  $i$ -ème gène. *c* est une liste dont le  $i$ -ème élément est la liste des probabilités des fonctions de  $F_i$ . On tire alors le contexte comme une  $n$ -liste dont la  $i$ -ème composante est piochée dans  $F_i$ .

- *varnames* (list) : nom des variables associées aux gènes
- *regulation* (tuple) : s'il est préalablement défini, le graphe de régulation du réseau. Si sa régulation n'est pas signée, son seul élément est la liste d'adjacence. Si sa régulation est signée, ses deux éléments sont les listes d'adjacence des arcs respectivement positifs et négatifs.

ATTRIBUTS - PARAMÈTRES DE SIMULATION :

- *sync* (bool) : mise à jour synchrone ou asynchrone
- *p* (float) : facteur de perturbation
- *q* (float) : facteur de changement de contexte
- *zeroes, ones* (list) : indices des gènes à fixer à 0 ou 1 dans l'état de départ

ATTRIBUTS - CONFIGURATION :

- *currentfct\_vector* (list) : contexte actuel, sous forme de liste des fonctions de régulation actuellement utilisées pour chaque gène.
- *x* (list) : état actuel, composé de  $n$  bits.

## D.2 Méthodes

Les méthodes présentées dans cette sous-section sont liées à la classe PBN. Pour une instance  $p$  représentant un PBN, appeler la méthode *fct* avec l'argument *arg* s'écrit dans le script : `p.fct(arg)`.

---

*simulation(N)*

**Description :**

À partir d'un état de départ choisi dans un sous-espace de  $\{0, 1\}^n$ , on itère le PBN les étapes décrites en figure 1.4.1.

**Arguments :**

$N$                       Nombre d'itérations.

---

*copy\_PBN(args)*

**Description :**

Renvoie une copie du PBN, dont des attributs sont modifiés si voulu.

**Arguments :**

*args*                      Attributs à modifier.

**Sortie :**

Une copie potentiellement modifiée du PBN.

**Exemple :**

```
> bnS = generateBN(4, 3, sync = True) # on génère un réseau booléen à mise à jour synchrone
> bnAS = bnS.copy_PBN(sync = False, p = 0.1) # le même réseau mais avec une mise à jour
asynchrone et une perturbation
```

---

*STG(f, plot\_attrs = True, draw\_labels = True)*

**Description :**

Le graphe de transition d'état est calculé à partir des successeurs de chaque état. Si le mode de mise à jour du modèle est asynchrone, les arêtes sont pondérées par la probabilité de transition. Les attracteurs sont coloriés avec une palette jaune à rouge selon leur taille, et les états transients en gris.

**Arguments :**

$f$	Contexte sélectionné.
$plot\_attrs$	Affichage dans des fenêtres supplémentaires des transitions restreintes à chaque attracteur cyclique ou complexe.
$draw\_labels$	Affichage des probabilités de transition sur les arêtes (si asynchrone).

**Sortie :**

Liste d'adjacence du STG, et ensemble des attracteurs.

---

$STG\_PBN(plot\_attrs = True, draw\_labels = True)$

**Description :**

Le PBN possède plusieurs contextes, notons  $w_f(x, y)$  la probabilité de transition d'un état  $x$  à un état  $y$  dans le contexte  $f$ . Pour le STG du PBN, on calcule les poids de ses arcs selon la somme pondérée suivante :  $w_{PBN}(x, y) = \sum_{f \text{ contexte}} c_f w_f(x, y)$ . Voir figure 1.10b.

**Sortie :**

Liste d'adjacence du STG, et ensemble des attracteurs.

---

$stationary\_law(show\_all = True, T = 100, N = 200, R = 100, pre = False, prio\_attrs = [])$

**Description :**

Simule le réseau selon l'algorithme 2 et affiche l'histogramme des états échantillonnés.

En mise à jour asynchrone, les états également présents dans la loi stationnaire synchrone sont coloriés dans une couleur différente.

En mise à jour synchrone, les états sont partitionnés selon la hauteur des barres, et la taille des bassins d'attraction est inférée selon la formule en section 2.1.

**Arguments :**

$show\_all$	Si True, affichage des barres de tous les états. Si False, affichage de seulement ceux visités lors de l'échantillonnage.
$T$	Nombre d'itérations initiales dans chaque simulation.
$N$	Nombre d'itérations pour lesquelles on consigne l'état visité, après les $T$ premières.
$R$	Nombre de simulations.
$pre$	Si True, on n'affiche que le tableau des fréquences et pas l'histogramme.



**Sortie :**

Tableau des fréquences de visite des états.

---

*regulation\_graph()***Description :**

Si le PBN possède un paramètre *regulation*, on affiche le graphe de régulation. Si les régulations strictement activatrices ou inhibitrices, le graphe est signé et ses arcs sont coloriés en vert et rouge. Si les régulations duales, le graphe affiché n'est pas signé.

---

*PBN\_to\_file(filename)***Description :**

Consigne les attributs du modèle dans un fichier texte.

**Arguments :**

filename            Fichier dans lequel écrire, créé dans le dossier 'output'.

---

*file\_to\_PBN(filename)***Description :**

Renvoie un objet PBN à partir du parsing d'un fichier texte dans le format décrit précédemment. Contrairement aux précédentes, ce n'est pas une méthode de classe.

**Sortie :**

Objet PBN dont la description est lue du fichier.

**Arguments :**

filename            Fichier à lire.

title                Titre du PBN. Si None, il prend le nom du fichier lu.

**Exemple :**

```
> bn = generateBN(4, 3) # on génère un réseau
> bn.PBN_to_file("fichier.pbn") # on le consigne dans un fichier
```

```
> bncopy = file_to_PBN("fichier.pbn") # on lit le fichier pour reformer le réseau
```

### Détails :

Le format du fichier texte est le suivant : Exemple donné dans l'article de référence [5, sect. 3] :

```
"""
sync = (0 ou 1)
p = (flottant entre 0 et 1)
q = (flottant entre 0 et 1)
init = """
indep = (0 ou 1)

(Si indep == 0)
targets, factors
w = (poids du contexte 1)
gene_1, f11
...
gene_n, f1n
...
w = (poids du contexte m)
gene_1, fm1
...
gene_n, fmn

(Si indep == 1)
targets, factors
gene_1, f11, c11
...
gene_1, f1l1, c1l1
...
gene_n, fn1, cn1
...
gene_n, fnln, cnln
"""

"""
sync = 1
p = 0
q = 1
init = """
indep = 1

targets, factors
x0, x1 | x2, 0.6
x0, (x1 | x2) & !(x1 & x2 & !x0), 0.4

x1, (x0 | x1 | x2) & (x0 | !x1 | !x2) & (x2 | !x0 | !x1), 1

x2, (x0 & (x1 | x2)) | (x1 & x2 & !x0), 0.5
x2, x0 & x1 & x2, 0.5
"""
```

L'exemple est un PBN indépendant à 3 gènes :  $x_0$ ,  $x_1$ , et  $x_2$ . Ils possèdent respectivement 2, 1, et 2 fonctions de régulation potentielles. Le nombre de contextes que l'on peut former est alors  $2 \times 1 \times 2 = 4$ .

## D.3 Générateurs

Ces fonctions ne sont pas des méthodes de classe.

---

*generateBN*( $n$ ,  $k$ ,  $sync$ ,  $v = False$ ,  $f = False$ ,  $p\_neg = 0.5$ ,  $p = 0$ )

### Description :

Génère et renvoie un BN selon les procédés décrit en section 2.3.

### Arguments :

$n$	Nombre de gènes du BN.
$k$	Nombre de régulateurs de chaque gène.
$v$	Si False, tous les gènes ont $k$ régulateurs. Si True, le nombre de régulateurs de chaque gène est tiré entre 0 et $k$ .
$f$	Si True, le PBN est généré selon le procédé de la section 2.3.1 (sans régulation duale, fonction par défaut). Si False, le PBN est généré selon le procédé de la section 2.3.2 (régulations duales autorisées, tables de vérité aléatoires).
$p\_neg$	Dans le cas $f$ =True, la probabilité qu'un régulateur soit inhibiteur. La probabilité qu'un régulateur soit activateur est alors de 1- $p\_neg$ .
$sync, p$	Attributs du PBN.

#### Sortie :

Objet PBN.

---

`generate_Random_PBN( $m, n, k, indep, sync = True, p = 0, q = .1$ )`

#### Description :

Génère et renvoie un PBN à partir de tables de vérité aléatoires, comme présenté en section 2.4.2.

#### Arguments :

$m$	Liste ou entier, le nombre de fonctions de régulation de chaque gène. Leurs tirages sont équiprobables.
$n$	Nombre de gènes du PBN.
$k$	Nombre de régulateurs de chaque gène.
$indep, sync, p, q$	Attributs du PBN.

#### Sortie :

Objet PBN.

---

`generate_Extended_PBN( $BN, i\_modifs = None, p\_ref = 0.8, dist = 1, part = 'poly', q = 1$ )`

À partir d'un BN, renvoie un PBN des fonctions voisines selon le procédé décrit en section 2.4.1.

#### Arguments :

<i>BN</i>	Réseau booléen à étendre.
<i>i_modifs</i>	Liste des indices des gènes dont on étend les fonctions.
<i>p_ref</i>	Probabilité associée à la fonction de référence.
<i>dist</i>	Distance à explorer dans le diagramme de Hasse de chaque fonction. Si 1, parents + enfants. Si 2, précédents + frères/sœurs. Si 3, précédents + grands-parents + petits-enfants.
<i>part</i>	Type partitionnement des voisins en fonction de la distance. Si ' <i>poly</i> ' : les voisins à distance $k$ ont un poids $r^k$ . Si ' <i>div</i> ', les voisins à distance $k$ ont un poids $r/k$ . Si ' <i>equal</i> ', les voisins ont tous le même poids.
<i>q</i>	Attribut du PBN, les autres étant copiés du BN pris en argument.

**Sortie :**

Objet PBN.