

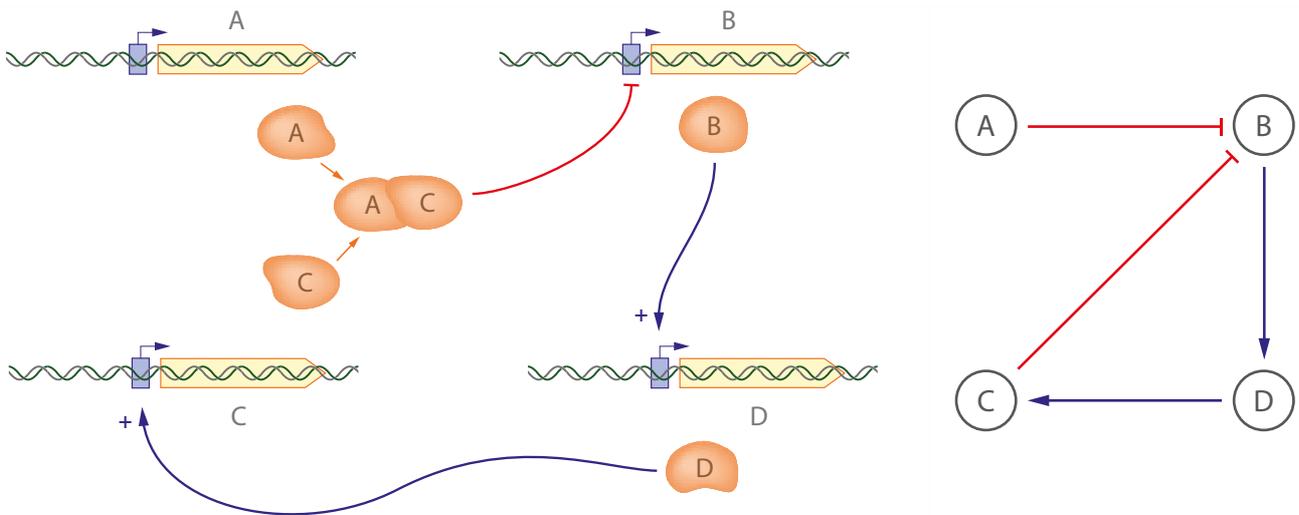
TD/Miniprojet

Réseaux booléens aléatoires : modèle pour les réseaux de gènes

Contexte

Le génome des organismes multicellulaires (plantes, animaux) comporte en général quelques dizaines de milliers de gènes (~20 000 dans le génome humain).

Ces différents gènes sont en interaction les uns avec les autres : certains gènes en activent d'autres et certains gènes en inhibent d'autres. Ceci constitue des "réseaux génétiques d'interaction".



Exemple de 4 gènes A, B, C, D en interaction : à gauche, les protéines A et C ensemble inhibent le gène B ; la protéine B active le gène D, dont la protéine active C. Le schéma à droite reprend l'organisation logique de ce même réseau

La question posée est celle de la modélisation dynamique de ces systèmes pour répondre à différentes questions liées :

- Ce type de réseau a-t-il des états stables ?
- Combien d'états stables différents peuvent exister pour un réseau donné ?
- Quelle est la dynamique de convergence vers ces états stables ?
- Comment la géométrie et la connectivité du réseau influencent son comportement ?

Dans un organisme multicellulaire complexe, il y a en général plusieurs types cellulaires (lymphocytes, neurones, hépatocytes, fibroblastes...) qui correspondent à différents états stables du réseau de gènes, avec certains gènes spécifiquement "éteints" ou "allumés" (les gènes de l'hémoglobine ne sont "allumés" que dans les globules rouges, par exemple)

Les réseaux booléens : un modèle minimaliste

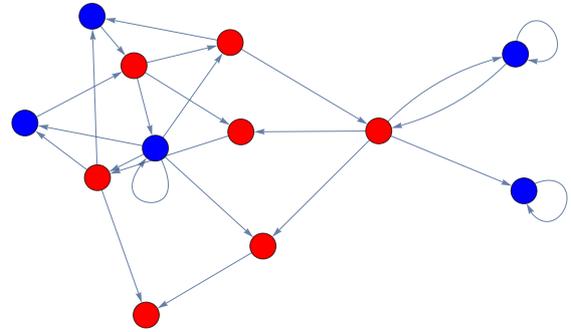
La figure ci-dessus illustre la parenté entre les réseaux génétiques (à gauche) et un type de structure logique appelé **réseau booléen** (à droite) :

- Un réseau booléen s'appuie sur un **graphe orienté**¹.

¹ Dans un graphe orienté, les arêtes du graphe ont un sens : elles partent d'un sommet pour aboutir à un autre.

- Chaque nœud ou sommet du graphe (correspondant aux gènes dans l'analogie ci-dessous) peut avoir deux états : *vrai* ou *faux* ("allumé" ou "éteint"...). Ces états évoluent en fonction des interactions avec les autres sommets
- Les arêtes du graphes déterminent ces interactions entre les sommets : l'état ultérieur d'un sommet est déterminé par celui des sommets qui possèdent une arête orientée pointant vers lui.

Le schéma ci-contre montre un exemple de réseau booléen simple avec une douzaine de sommet et une vingtaine d'arêtes. La couleur rouge ou bleue indique l'état de chaque sommet à un instant t donné. Notez que certains sommets sont ont des arêtes qui partent et aboutissent à eux-même. Ceci correspond à une situation où on a un rétro-contrôle ou *feedback* d'un gène sur lui-même.



On qualifie ces réseaux de "booléens", parce que l'état de chaque sommet ne peut prendre les deux valeurs booléennes *vrai* ou *faux*.

Paramètres du réseau

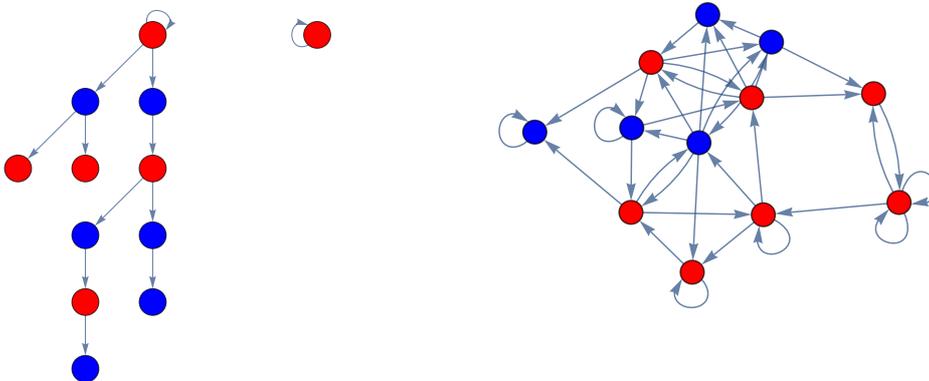
Ces réseaux peuvent se caractériser par plusieurs paramètres topologiques. On utilisera par la suite ces paramètres pour déterminer comment ils sont reliés au comportement du réseau :

Ordre d'un sommet

L'ordre d'un sommet est le nombre d'arêtes entrantes sur ce sommet.

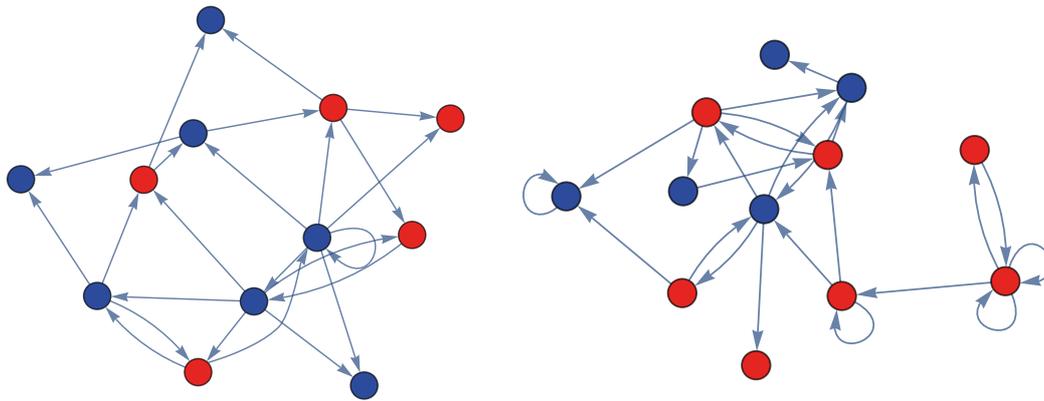
Connectivité

On définit la connectivité du réseau comme étant l'ordre moyen de ses sommets ou le **nombre moyen d'arêtes entrantes** sur chaque sommet. Le schéma ci-dessous montre un réseau de connectivité 1 (à gauche) et un réseau de connectivité 3 (à droite).



Homogénéité

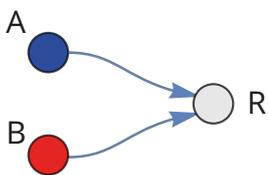
Le réseau est dit homogène d'ordre k lorsque tous ses sommets sont d'ordre k . Autrement dit, chaque sommet du réseau a **exactement k arêtes entrantes**.



Les deux réseaux ci-dessus ont une connectivité de 2 (nombre moyen d'arêtes entrantes par sommet). Ils ont tous les deux 12 sommets et 24 arêtes. Celui de gauche est homogène : chaque sommet a **exactement 2** arêtes entrantes. Celui de droite ne l'est pas, certains sommets n'ont qu'une arête entrante (le bleu en haut, par exemple) et d'autres en ont trois (le bleu tout à gauche).

Evolution de l'état des sommets / fonctions booléennes

L'objectif de la modélisation est de voir comment l'état des sommets d'un réseau donné évolue au cours du temps. Pour cela, il faut définir les règles d'évolution de chaque sommet. On utilise pour cela des fonctions booléennes dont les entrées (input) sont l'état des sommets qui lui sont reliés par des arêtes entrantes.



Exemple d'un sommet R (en gris) avec deux arêtes entrantes. Les valeurs de A et B sont les entrées (input) de R. Son état suivant sera déterminé par une fonction booléenne de A et B.

A	B	R	A	B	R	A	B	R	A	B	R	A	B	R	A	B	R
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1
0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	1	0	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
Null: 0			Identity: 1			A-transfer			B-transfer			\bar{A} -transfer			\bar{B} -transfer		

A	B	R	A	B	R	A	B	R	A	B	R	A	B	R	A	B	R
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	1	0	0	1	0	1	1	0	0
1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1
AND: $A \cdot B$			OR: $A+B$			NAND: $\overline{A \cdot B}$			NOR: $\overline{A+B}$			XOR: $A \oplus B$			XNOR: $\overline{A \oplus B}$		

A	B	R	A	B	R	A	B	R	A	B	R
0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	1	0	0	1	0	0	1	1
1	0	0	1	0	1	1	0	1	1	0	0
1	1	0	1	1	0	1	1	1	1	1	1
$\bar{A} \cdot B$			$A \cdot \bar{B}$			$A + \bar{B}$			$\bar{A} + B$		

Il existe différentes fonctions booléennes possibles pour chaque sommet. Dans le cas particulier de deux inputs (comme sur le schéma ci-dessus), ce sont les fonctions booléennes habituelles : AND, OR, NAND, NOR, XOR... Pour 2 inputs, il en existe 16 en tout pour listées ci-dessus avec leurs tables de vérité.

Chaque sommet R se voit initialement attribuer l'une de ces 16 fonctions booléennes. Celle-ci est ensuite utilisée itérativement pour calculer son état à l'instant $t + 1$, à partir des inputs à l'instant t .

Si il y a plus de 2 inputs, on peut assez facilement généraliser ces fonctions booléennes. Avec n inputs, on a 2^n combinaisons d'états possibles en entrée, le nombre de fonctions distinctes est donc 2^{2^n} (128 pour 3 inputs, 65536 pour 4 inputs...).

Réseaux booléens aléatoires

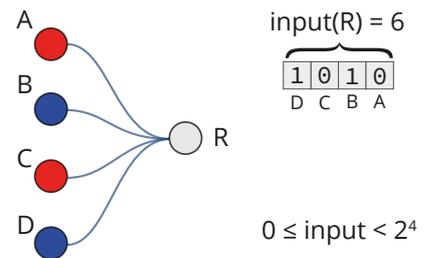
Un réseau booléen aléatoire est un réseau dont chaque sommet est initialisé avec des paramètres entièrement aléatoires :

- Connectivité, c'est à dire ses n sommets en entrée.
- Fonction booléenne de ces n entrées (choisie aléatoirement parmi les 2^{2^n} possibles pour chaque sommet)
- État initial (*vrai* ou *faux*)

Sur ces réseaux, on s'intéresse aux propriétés génériques et aux comportements qui peuvent émerger indépendamment de la géométrie et de l'état initial spécifique du réseaux. Pour déterminer ces éventuelles propriétés émergentes, on se propose de simuler ces réseaux.

Implémentation des fonctions booléennes

Si un sommet à n inputs, le nombre de combinaison d'inputs différents qu'il peut avoir est donc 2^n . On peut représenter cet ensemble d'inputs sous forme d'un nombre entier compris entre 0 et $2^n - 1$, dont la représentation en binaire correspond aux valeurs de ces inputs.



Sur l'exemple du schéma à droite, le sommet R a 4 inputs (A, B, C, D), l'entrée du sommet R peut donc être représenté par un entier entre 0 et 15 ($2^4 - 1$) dont la représentation binaire correspond aux valeurs de ces 4 inputs (6 dans cet exemple).

Étant donné ces 2^n valeurs possibles des entrées, pour chacune de ces valeurs, il y a deux outputs possibles (*vrai* ou *faux*) ce qui donne donc bien 2^{2^n} différentes fonctions booléennes possibles.

Étant donné une valeur d'input i ($0 \leq i < 2^n$), la valeur de la k -ième fonction booléenne pour cet input i correspond à la valeur du i -ième bit de k .

Exemple avec 2 inputs

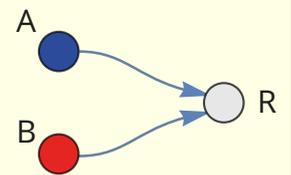
Avec 2 inputs A et B (schéma de droite), on a 4 valeurs possibles en entrée :

(0, 0); (0, 1); (1, 0) et (1, 1).

Il y a $2^{2^2} = 16$ fonctions booléennes différentes de 2 inputs, que l'on peut numéroter de 0 à 15. Si on veut générer la 8^{ème} de ces 16 fonctions booléennes (celle qui a le numéro $k = 7$, puisqu'on démarre à 0), il faut examiner la représentation binaire de 7 : 0111.

Ceci permet de générer la table de vérité de cette fonction booléenne (à droite). La colonne *Input* représente la valeur numérique de l'entrée. La colonne *Valeur* contient le résultat de la fonction booléenne. Les bits de cette colonne correspondent à la représentation binaire de $k = 7$: 0111.

Cet exemple correspond à la fonction logique classique **A ou B**



B	A	Input	Valeur
0	0	0	0
0	1	1	1
1	0	2	1
1	1	3	1

Implémentation des sommets du réseau

En Python, on peut implémenter les sommets comme une classe d'objets. Les items associés à un objet de la classe *sommet* seraient :

- État du sommet (Booléen : True/False)
- Nombre d'input(s)
- Liste d'Index ou de pointeurs sur les inputs (autres sommets qui pointent vers lui)
- Index k de la fonction booléenne

On peut aussi éventuellement définir des fonctions associés aux objets de la classe *sommet* : réinitialisation de l'état de départ, modification de la fonction booléenne associée, transition...

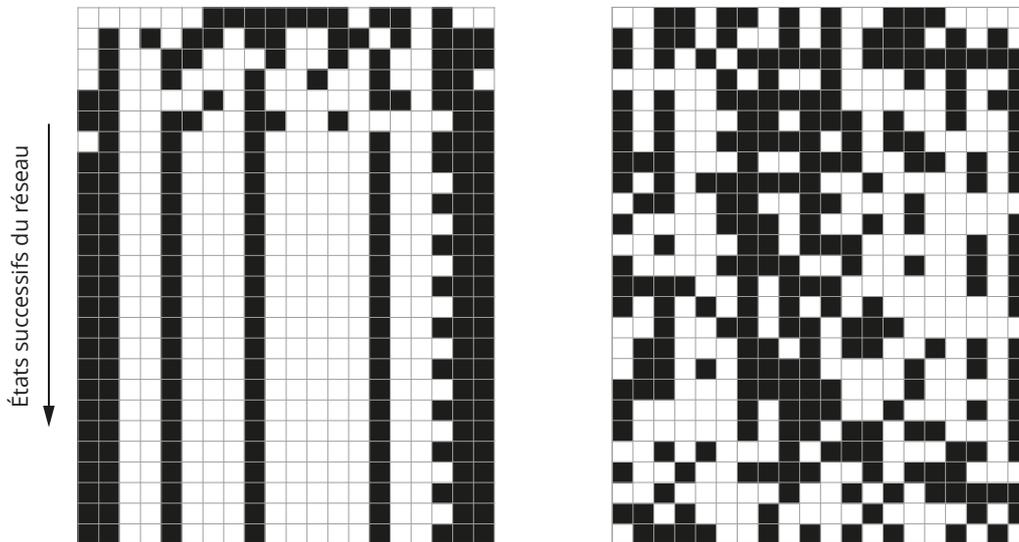
Simulation

Le plus souvent, on représente l'état du réseau à un instant t sous forme d'une ligne de case noires ou blanches, chaque case représentant un sommet du réseau. La case est blanche si le sommet est inactif et noire si il est actif à cet instant donné².



Représentation de l'état instantané d'un réseau avec 20 sommets

Pour représenter l'évolution du réseau au cours du temps, on utilise un tableau dont les lignes représentent les différents états successifs du réseau. Une colonne du tableau représente donc l'évolution d'un sommet au cours du temps :



Représentation de l'évolution de deux exemples de réseau. À gauche, un réseau qui converge vers un état périodique stable après quelques itérations, à droite un réseau à l'évolution chaotique (non périodique).

On peut également suivre l'évolution d'un sommet donné du réseau en fonction du temps.

² Ou toute autre convention de couleur au choix...

Questions posées :

L'objectif de cette étude est d'expérimenter les propriétés générales de ces réseaux booléens aléatoires, en particulier en fonction de la connectivité du réseau (nombre d'entrées de chaque sommet).

Une propriété générale est un comportement typique, observée pour la majorité des réseaux, généralement indépendante du cas particulier.

Parmi ces propriétés générales, on s'intéresse entre autres à :

- La convergence vers un état stable ou périodique (attracteur du système)
- Le nombre d'attracteurs différents (stable ou périodique). On peut tester ce point en examinant la convergence d'un même réseau à partir de différents états initiaux
- La rapidité de la convergence (au bout de combien de transition on tombe en moyenne vers ces états stables ou périodiques)
- La robustesse de ces attracteurs : est-ce que les attracteurs sont globalement stables lorsqu'on modifie une caractéristique ponctuelle du réseau (une fonction booléenne d'un sommet, par exemple ? En termes de signification biologique, cela correspond à la résistance générale du système à une ou plusieurs mutations.

On peut également envisager d'élargir à certaines généralisations afin de voir comment ça influe sur le comportement global :

Les réseaux booléens **non-homogènes** (ceux qui ont un nombre d'entrées variable, par exemple où chaque nœud a en moyenne 2 entrées, au lieu de exactement 2 entrées)

Les réseaux booléens asynchrones (dans lesquels les transitions de chaque sommets ne font pas toutes en même temps).