Utilisation de la logique des hypothèses pour la modélisation des voies de signalisation dans la cellule

Andrei Doncescu [1] - Pierre Siegel [2]
[1] LAAS Toulouse - [2] SNC Toulon

Résumé : La description formelle des interactions dans la cellule est un champ d'investigation très adapté à l'Intelligence Artificielle. On est, en particulier, dans un cadre de représentation de connaissances incomplètes et d'abduction. Les problèmes posés par la complexité algorithmique sont également très présents. Ce papier présente une approche basée sur une logique non-monotone, la logique des hypothèses. Cette logique permet dans un seul formalisme de traiter les questions de raisonnement révisable (défauts) et d'abduction. Des pistes pour la mise en œuvre algorithmique de cette représentation, basées sur la notion de champ de production et d'algorithme de production de cette représentation sont également données.

1 Introduction.

1.1 Biologie, mathématique et informatique.

Aujourd'hui des experts de plus en plus spécialisés ont besoin de la mise à l'échelle de leurs connaissances à partir des données accessibles pour pouvoir découvrir des nouvelles sources de savoir qui se veulent valides et utiles pour apprendre ces systèmes. Dans ce cadre, durant ces dernières décennies, la biologie a connu un développement prolifique dans toutes ses facettes. De nouveaux champs d'applications et d'études comme les biotechnologies, les nanotechnologies ou la bioinformatique ont immergés et prennent une place importante dans le contexte de recherche actuel ayant comme but l'identification, la compréhension et la quantification des phénomènes biologiques vus à l'intérieur du système biologique. La biologie fournit alors de nouveaux défis et champs d'études aux mathématiques, à l'automatique, à l'informatique et aux nouvelles technologies. L'Intelligence Artificielle a ici largement sa place.

Les systèmes biologiques changent nos méthodes d'investigation. Ceci est du à leurs capacités d'avoir une vie autonome par des phénomènes de régulation, de se reproduire, de communiquer. L'analyse de ces systèmes constitue une multitude de données variées par la diversité des modèles biologiques étudiés, des conditions d'analyse, des niveaux d'observation (macroscopique ou microscopique ou moléculaire), des outils d'investigation mis en oeuvre. Cependant, malgré la quantité et la qualité des données générées, de nombreuses questions restent en suspens et l'aspect discret des informations collectées ne permet pas à ce jour d'établir une analyse systémique des phénomènes biologiques.

Dans les tentatives de description du comportement des systèmes vivants, quand la modélisation déductive n'a pas abouti, la démarche par le raisonnement qualitatif basé sur la fonction des molécules a montré ses limites. Si l'on sait, par homologie, attribuer des propriétés à un élément d'un système vivant à partir de sa structure, actuellement, il semble impossible de les déduire ni qualitativement, ni quantitativement. De la même manière, si les propriétés sont connues, nous ne pouvons, clairement, en déduire leur fonction dans la cellule vivante, et à partir des caractéristiques des cellules vivantes, calculer leur comportement dans un environnement déterminé. En général, cette démarche déductive de description comportementale échoue du fait que la fonction d'un élément constitutif du

système vivant dépend du fonctionnement simultané des autres éléments. Le problème récurrent est que, par le calcul ou autrement, les propriétés fonctionnelles de la cellule ne peuvent être déduites des propriétés de ses seuls composants. Dans ce contexte, avec leur capacité à décrire la complexité, les outils mathématiques offrent une perspective d'analyser ces éléments structurels du monde vivant pour proposer des fonctionnalités prenant en compte leurs interactions dynamiques itératives fortement non linéaires.

Diverses approches de cette description du comportement de la cellule sont actuellement développées par la communauté scientifique internationale [5]. La « cellule virtuelle » regroupe un ensemble de logiciels permettant de décrire l'organisation métabolique d'un organisme, d'estimer la distribution de la matière dans son fonctionnement et d'extrapoler ses potentialités de production. L'«E-cell » permet de quantifier l'activité cellulaire à partir de cinétiques parfaitement caractérisées expérimentalement sur le plan catalytique et quantitatif.

La description principalement cinétique qui oblitère l'analyse des mécanismes réels ou possibles pouvant être mis en jeu. Une autre démarche consiste à rechercher une description précise d'une cellule vivante (ou d'une partie) basée sur des mécanismes et des valeurs paramétriques expérimentalement déterminées, à savoir des données excluant toutes valeurs estimées sur des modèles partiels [14].

1.2 La cellule vue du côté de l'Intelligence Artificielle.

Du point de vue de l'Intelligence Artificielle les cellules sont des sources d'informations qui intègrent une myriade de signaux intra et extra cellulaire ayant comme ultime but des sorties optimales. La maladie et le cancer en particulier peuvent être vus comme une pathologie de l'altération dans les réseaux de signalisation de la cellule. L'étude des évènements de signalisation apparaît comme une clef pour les recherches, biologique, pharmacologique et médicale. La propagation de ces types de signaux implique un changement du comportement des protéines sur trois niveaux : régulation de l'activité, interaction et expression. Les trois niveaux sont synchronisés dans une forte dynamique qui conduit aux changements de l'activité des protéines. Depuis une dizaine d'années les réseaux de signalisation ont été étudiés en utilisant des méthodes analytiques basées sur la reconnaissance des protéines par des anticorps spécifiques. En parallèle les puces à ADN (microarrays) sont largement utilisées pour étudier la co-expression de gènes candidats permettant d'expliquer l'étiologie de certaines maladies, dont le cancer. Elles apportent de nombreuses données à la modélisation des graphes d'interactions des gènes entre eux, dont les propriétés mathématiques : connectivité, connexité, présences de boucles positives et négatives sont liées aux propriétés asymptotiques des réseaux de régulation génétique.

En se plaçant du côté logique et informatique, l'étude des réseaux de gênes pose des problèmes identifiés et étudiés en Intelligence Artificielle depuis une trentaine d'année. On est ici est dans le cadre du raisonnement à partir d'informations incomplètes, incertaines, révisables, contradictoires et multi sources. En effet la description logique ou mathématique des voies de signalisation n'est pas complète : les expériences biologiques donnent un certain nombre d'interactions entre protéines mais certainement pas toutes. D'autre part les conditions et la difficulté des expériences font que les données ne sont pas toujours précises. Certaines données peuvent être même erronées et doivent être corrigées ou révisées par la suite. Enfin les informations viennent de sources et d'expériences différentes et peuvent être

contradictoires. A partir de ce constat, général pour la plupart des activités humaines réelles, il faut pourtant raisonner et prendre des décisions. C'est le but de toutes les logiques de l'incertitude et plus particulièrement des logiques non monotones que nous utiliserons dans cette étude.

Une autre problématique de l'IA, fondamentale pour ce cadre, est de compléter in-silico les réseaux de gênes. Le réseau de gêne est incomplet et les expériences biologiques sont faites pour essayer de le compléter. Mais ces expériences sont longues et couteuses. Il faut donc savoir où une molécule (un futur médicament) à une chance d'agir de manière efficace. On est ici dans cadre bien connu du raisonnement abductif. En simplifiant, l'abduction consiste à trouver le/les ensemble "minimaux" d'informations, qu'il faut ajouter à un ensemble F de faits connus, pour déduire un résultat R que l'on veut prouver. Par exemple, ici F peut être le réseau de gênes, R l'inhibition d'une protéine et X des interactions entre gênes non connues. Une telle approche permet donc de cibler les expériences biologiques à faire, et donc de gagner du temps en éliminant les expériences qui ont peu de chance de donner des résultats utilisables

Les deux principales questions qui se posent vont être de représenter formellement ces interactions et d'essayer de les compléter in silico. On est donc bien dans les cadres de la représentation de connaissances incomplètes, de la causalité, du raisonnement incertain et de l'abduction. Et bien entendu il faut d'obtenir des algorithmes d'une complexité algorithmique raisonnable dans la pratique. La cerise sur le gâteau est que tout ceci doit être compréhensible par des biologistes.

Le but est donc d'exprimer les connaissances de biologistes concernant les réseaux de signalisation dans un langage informatique sans ambigüité. Ce langage doit permettre de prendre en compte l'incertain et de donner aussi un sens précis aux formules associées à ce type de connaissances. Ces formules seront utilisées dans un système de démonstration automatique efficace. Ce système permettra, d'une part de déduire des faits à partir des données (déduction), et d'autre part de découvrir d'autres connaissances ou d'expliquer des phénomènes par analogie (raisonnement inductif et abductif), ce qui ne peut pas se faire par des méthodes analytiques du fait de la grande masse de données et de leur complexité. L'objectif étant de développer un système permettant :

1- La modélisation des règles de comportement d'un gène, d'une cellule.
2- Le raisonnement avec des informations incertaines, incomplètes et révisables.
3 - La découverte des connaissances manquantes qui expliquent certains phénomènes biologiques
4- Le choix entre différentes propositions quand plusieurs actions ou solutions sont proposées pour la même situation.
5- L'addition de nouvelles règles de comportement sans remettre en question les règles précédentes.
6- Le raisonnement avec des règles générales sans avoir à compiler toutes les possibilités.

2. Interactions dans la cellule.

2.2.1 Un petit exemple.

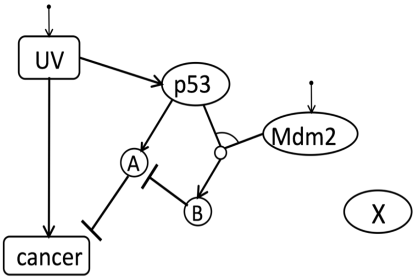

Figure 1

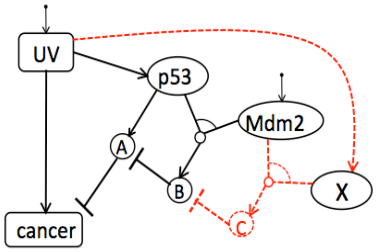

Figure 2

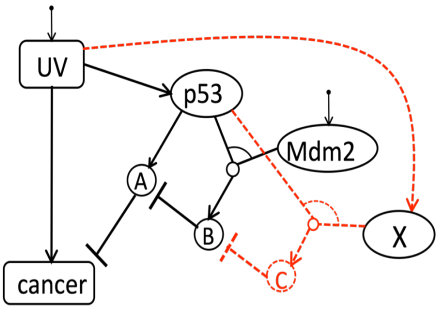

Figure 3

La figure 1 donne un exemple très simplifié d'interactions dans une cellule. Par des mécanismes divers non indiqués ici, les ultraviolets (UV) mettent la cellule en apostase (elle devient de fait immortelle) d'où le cancer. Ceci est représenté par une flèche. D'un autre coté les UV activent la production de la protéine P53. Cette protéine va activer une protéine A qui va bloquer le cancer. Mais P53 lié à la protéine Mdm2 va produire B, qui va bloquer A. Pour un biologiste la question est donc de bloquer le cancer en bloquant B. Les expériences biologiques ont montré que le X pourrait être un candidat pour ce blocage. Les figures 2 et 3 donnent deux types d'interactions possibles avec X pour expliquer ce blocage de B. Ici on a complété le graphe à la main, le problème est d'utiliser l'informatique pour ce faire.

Cet exemple est un jouet bien entendu très élémentaire, dans la pratique cette représentation graphique des voies de signalisation du réseau peut porter sur plusieurs milliers de gênes [20]. Mais le problème est tout de même non trivial car la découverte (l'abduction) va porter d'une part sur la présence éventuelle d'une protéine C et d'autre part sur l'ajout de liens entre protéines.

2.2 Représentation logique.

Les gênes et les protéines sont considérées comme un même objet (les gênes produisent les protéines). On va souvent se restreindre ici à une représentation propositionnelle. Dans la pratique l'étude fine des interactions va demander de représenter des augmentations ou des diminution de concentration de protéines. On sort donc du cadre propositionnel mais les problèmes de base sont les mêmes. Pour représenter une variation de concentration il est par exemple possible d'utiliser des prédicats de type "augmente" ou "diminue" et de limiter l'utilisation de ces prédicats [3].

Pour décrire les interactions entre les gênes dans la cellule, on part d'un langage L de logique classique (propositionnel ou de premier ordre). Dans L, la proposition *A* (resp ¬*A*) signifie que *A* est vrai (faux). On pourra dire par exemple *donne(UV)* pour dire que la cellule est soumise à des ultra-violets, ou encore, *écran-verre –> ¬donne(UV)* pour dire qu'un écran de verre protège des ultra-violets. On est dans un cadre logique, donc il est possible de représenter à peu près tout ce que l'on veut de manière naturelle. Le prix à payer, si on utilise la totalité du langage peut être l'explosion combinatoire des algorithmes.

Les interactions entres gênes sont une forme très simple de causalité. Pour exprimer ces interactions il est courant d'aller à l'essentiel en donnant deux relations binaires *cause(A,B) et bloque(A,B*) [4]. La première relation veut dire, par exemple, qu'une protéine A déclenche la production d'une protéine B la deuxième est une inhibition. De manière classique ces relations sont représentées dans le réseau de gênes, par A —> B et A —| B. Bien entendu cette causalité est élémentaire et de nombreux travaux très savants ont été écrits pour représenter les causalité [10] [11] [12].

Si l'inférence de logique classique *A —> B* est parfaitement décrite formellement, avec toutes les « bonnes » propriétés mathématiques et informatique (tautologie, non contradiction, transitivité, contraposée, modus ponens, compacité ...), la description des propriétés formelles de la causalité, est moins simple. La causalité ne peut pas être vue comme une relation logique classique. Un exemple élémentaire en est celui de l'expression « S'il pleut la pelouse est mouillée » Cette expression ne peut pas être traduite par la formule *pluie —> pelouse-mouillée,* qui signifierai que dès qu'il pleut la pelouse est automatiquement mouillée. En effet, il peut y avoir des exceptions à cette règle (la pelouse est sous un hangar…). On peut aussi changer l'environnement (on bâche la pelouse). Ces règles révisables et avec exceptions sont bien connues en Intelligence Artificielle. Elles ont donné lieu, en particulier, aux logiques non monotones et aux théories de la révision. D'un autre côté et de manière plus technique, on retrouve ici tous les problèmes classiques qui se posent quand on veut essayer de formaliser et d'utiliser la négation par échec dans des langages de programmation tels que Prolog ou Solar [13].

Pour donner les liens entre nos relations causales *cause* et *bloque*, dans un langage classique (calcul propositionnel ou logique du premier ordre) il faut donc faire deux choses :

- décrire les propriétés internes aux relations cause et bloque

- décrire les liens entre ces relations et la logique classique

Tout ceci en prenant en compte le problème de l'incertain et du révisable. Pour le premier aspect on donnera de manière explicite les liens minimum et nécessaires entre les deux relations causales. Les liens avec la logique classique seront décrits dans un premier temps

en logique des défauts. Ensuite, pour prendre en compte l'aspect de la découverte (abduction, champ de production) on les donnera en utilisant la logique des hypothèses.

Dans notre contexte, pour donner ces liens entre les relations cause et bloque on va au plus simple, en utilisant la logique classique. La solution élémentaire est alors de donner explicitement les deux schémas d'axiomes :

(C1) cause(A,B) ∧ cause(B, C) —> cause(A, C)
(C2) cause(A, B) ∧ bloque (B, C) —> bloque(A, C)

Nous pensons qu'il s'agit là du système axiomatique minimum, nécessaire et sans doute suffisant, pour l'application à la cellule. Pour l'instant il n'y a aucun lien formel entre les deux relations. Il est bien entendu possible d'ajouter d'autres axiomes pour prendre en compte ces liens, comme par exemple [8] [9] mais ceci complique le système formel sans que la pertinence de ces ajouts soit évidente dans ce cadre.

2.3 Causalité et inférence classique.

Dans une première approche, les premières propriétés que l'on a envie de donner peuvent s'exprimer naturellement, par des règles du type :

(1) Si A cause B et si A est vrai, alors B est vrai.
(2) Si A bloque B et si A est vrai, alors B est faux.

Suivant le contexte, le *vrai* peut vouloir dire *connu, certain, cru* ... ou encore, plus techniquement dans un système de démonstration automatique *démontré*.

La première idée est d'exprimer ces lois en logique classique par les axiomes :

cause(A, B) ∧ A —> B
bloque(A, B) ∧ A —> ¬B

On peut aussi les exprimer plus faiblement par des règles d'inférences proches du modus ponens :

cause(A, B) ∧ A / B
bloque(A, B) ∧ A / ¬B

Mais ces deux formulations posent problème dès qu'il y a conflit. Si par exemple on a un ensemble F de quatre formules F = {A, B, cause(A, C), bloque(B, C) }, on va dans les deux approches données ci dessus inférer de F, B et ¬B ce qui est inconsistant. Pour résoudre ce type de conflits, on peut essayer d'utiliser des méthodes inspirées par la programmation par contraintes, par exemple l'utilisation de la négation par échec. Il est aussi possible d'utiliser un raisonnement révisable, en particulier une logique non monotone. La première approche pose beaucoup de problèmes théoriques et techniques si on quitte les cas simples. Ces problèmes sont souvent résolus en ajoutant des propriétés au système formel, propriétés qui posent d'autres problèmes ... et on arrive à une belle usine à gaz. On va étudier ici une approche non monotone. Dans un premier temps, on utilise la logique des défauts.

2.4 Interactions et logique des défauts.

Pour résoudre les conflits vus ci dessus, l'idée intuitive est d'affaiblir la formulation des règles de causalité en :

(1') Si A cause B , et si A est vrai, et s'il *est possible que B*, alors B est vrai.
(2') Si A bloque B, et si A est vrai, et s'il *est possible que B est faux* alors B est faux.

La question est alors de décrire formellement le *possible*. Cette question a commencé à se poser en Intelligence Artificielle il y a une trentaine d'année, quand on a voulu formaliser le raisonnement humain naturel. Dans ce type de raisonnement, on est obligé de raisonner avec des informations incomplètes, incertaines, révisables et quelquefois fausses. D'autre part il faut souvent choisir entre plusieurs conclusions possibles antinomiques. L'exemple de base en est : {Les pingouins sont des oiseaux, Les oiseaux volent, Les pingouins ne volent pas}. Si Titi est un pingouin on arrive à une contradiction, le système est inconsistant. Cette inconsistance peut être levée si on arrive à gérer l'exception en remplaçant "Les oiseaux volent" par "*Généralement* les oiseaux volent". Les logiques non monotones décrivent formellement les modes de raisonnement qui prennent en compte ces phénomènes.

On utilisera ici une des logiques non-monotones les plus connues, la logique des défauts de Reiter. Dans cette logique, les règles (1') et (2') vont s'exprimer intuitivement.
(1") Si A cause B , et si A est vrai, et si *B n'est pas contradictoire*, alors B est vrai.
(2") Si A bloque B", et si A est vrai, et si ¬*B n'est pas contradictoire* alors ¬B est vrai.

Dans la logique des défauts, ces règles vont se représenter par des *défauts normaux* et s'écrire :

d1 : cause(A, B) ∧ A : B / B
d2 : bloque(A, B) ∧ A : ¬B / ¬B

Pour le cas élémentaire qui précède, si A est vrai, on a :
W = {A }
D = {d1, d2}

et on obtient deux extensions :
E1 qui contient B (en appliquant d1)
E2 qui contient ¬B (en appliquant d2)

Le conflit est donc résolu, mais se pose le problème des extensions à préférer ; est ce que B est induit ou bloqué? En fait ceci va vraiment dépendre du contexte. On peut par exemple préférer les interactions positives par rapport aux négatives ; ou bien utiliser des méthodes statistiques ou probabiliste. Une autre approche est de pondérer chaque extension en fonction des propriétés du problème. Du point de vue algorithmique cette préférence peut être évaluée pendant le calcul de l'extension ou sur le résultat final. Nous avons appliqué cette dernière technique à un problème industriel, l'utilisation des logiques des défauts pour simuler le comportement d'un officier de quart à bord d'un sous marin en temps de guerre, avec de bons résultats pratiques [21] [22] [23]. Les références sur le choix de l'extension sont nombreuses dans la littérature. On ne tranchera pas dans cet article pour un type de préférence applicable à la cellule, ça dépendra beaucoup du contexte et les biologistes ne sont pas toujours d'accord entre eux.

3. Une alternative à la logique des défauts : la logique des hypothèses.

La logique des défauts pose nombre de problèmes théoriques et pratiques. Pour résoudre ces problèmes on va utiliser la logique des hypothèses [15] [17] [18] [19]. Un fragment de cette logique est proche des modèles stables définis ultérieurement. Un des avantages de la logique des hypothèses est que les informations sont décrites dans une logique bimodale classique. Il sera alors possible d'utiliser ce formalisme pour décrire l'ensemble de toutes les informations, incertaines ou non. Cette logique va aussi permettre d'utiliser des algorithmes de recherche de conséquence pour traiter l'abduction ; ces mêmes algorithmes étant utilisés pour calculer les extensions.

Dans ce cadre on utilisera la notion de champ de production et les algorithmes liés donnés dans [1] [16]. Ces notions ont permis en particulier de définir le langage Solar [13] développé par l'équipe d'Inoue Katsumi.

3.1 Une très rapide introduction aux logiques modales.

Le calcul propositionnel et la logique du premier ordre donnent une valeur de vérité Vrai ou Faux aux formules. Les logiques modales vont nuancer ces valeurs. Au lieu de pouvoir exprimer uniquement "Il fait (ne fait pas) beau", on peut avoir envie de dire "Je sais qu'il fait beau", "Il est possible qu'il fasse beau", "Je crois qu'il fait beau, "Demain il fera beau" .. Techniquement, pour décrire ce type d'informations on va ajouter *un opérateur modal*, qui sera noté la plupart du temps L ou □ (dans la suite on utilisera L). On associe à cet opérateur son dual M défini par M = ¬L¬ [24]. Le sens qu'on donnera à ces couples {opérateur,dual} varie suivant le contexte. On utilise par exemples les couples {nécessaire, possible} {sait que, croit}, {obligatoire, permis}.. Si f est une formule quelconque, Lf veut dire qu'il est nécessaire que f soit vrai et Mf va signifier que f est possible (Mf = ¬L¬ f dit qu'il n'est pas nécessaire que f soit faux). La définition inductive de l'ensemble des formules autorise la combinaison les modalités dans les formules.
Par exemple L(p —> (M(p —>L¬q))) est une formule de logique modale.

Comme pour toute logique on donne alors des règles d'inférence et des schémas d'axiomes. La base est l'ensemble des règles et axiomes de la logique du calcul propositionnel ou de la logique du premier ordre. Pour les logiques *normales* (celles qui ont une sémantique des mondes possibles de Kripke) on a toujours la règle d'inférence de nécessitation (N) : f / Lf qui signifie que si f est un théorème/tautologie alors Lf est un théorème (ou intuitivement, si f est toujours vrai, alors f est nécessaire). Pour un système normal on va aussi ajouter un axiome de distribution, appelé (K). Cet axiome s'écrit L(A —> B) —> (LA —> L B) et permet de retrouver, pour la modalité, les propriétés de distributivité de la logique de premier ordre. Une logique modale qui contient (N) et (K) est appelée système K. Dans cet article on va aussi utiliser l'axiome (T) : LA —> A (si A est nécessaire/démontré alors A est vrai). En ajoutant au système K cet axiome (T) on obtient le système T.

3.2 Logique des hypothèses.

Pour définir la logique des hypothèses on part d'une logique classique (calcul propositionnel ou logique du premier ordre) à laquelle on ajoute deux opérateurs modaux, **L** et **[H].** L'opérateur **L** a les propriétés du système T et **[H]** celles du système K.

a) Intuitivement, si f est une formule **L**f signifie que f est *certain*, ou encore *démontré* dans un cadre de programmation par contrainte).

b) Pour décrire la notion d'hypothèse on va définir **H** comme le dual de [**H**], c'est à dire que **H**f = ¬[**H**]¬f. Intuitivement **H**f signifie que l'on fait l'hypothèse f.

c) Le seul lien entre les opérateurs **H** est **L** est donné par un axiome de liaison :

Hf —> ¬L(¬f) <=> L(¬f) —> ¬ Hf <=> ¬ (Hf ∧ L(¬f))

Si on fait l'hypothèse f, alors impossible de prouver ¬f .

Il est interdit de faire l'hypothèse f et de prouver ¬f en même temps.

Définitions :

- Une *hypothèse* est une formule fermée **Hf**.

- Une *théorie des hypothèses TH* , est un couple TH = { HY , F} ou F est un ensemble de formules du langage et HY un ensemble d'hypothèses

- Pour définir une extension, dans cette logique, on passe en fait à une approche préférentielle. Une *extension* de TH est un ensemble maximal consistant E = F ∪ H'Y ou H'Y est un sous ensemble de HY (on ajoute à F le plus d'hypothèses possible).

3.2 Défauts et logique des hypothèses :

On peut traduire toute logique des défauts en une théorie des hypothèses. Une théorie des défauts Δ ={W, D} se représentera par une théorie des hypothèses THΔ = { HY , F} ou:

- L'ensemble HY des hypothèses, est l'ensemble de tous les Hb, tels que b est la justification d'un défaut de Δ.

- L'ensemble des formules de F est composé :

a) De toutes les formules de type Lf ou f appartient à W

b) Des formules : Lp ∧ Hq —> Lr, pour tout défaut p :q / r de D.

Pour représenter l'information, on utilise en fait ici seulement un fragment de la logique des hypothèses car il n'y a pas de combinaison de modalités.

On montre alors qu'on représente bien la logique des défauts en supprimant tous les cas pathologiques (en particulier, toute théorie a une extension).

Propriétés :

- Si F est consistant THΔ, a toujours au moins une extension.

- Toute extension E de Δ, est représentable par une extension E' de THΔ. Ces extensions E et E' étant telles que f ∈ E si et seulement si Lf ∈ E'.

- La réciproque n'est pas vraie. Certaines extensions de THΔ, n'ont pas leur dual comme extension de Δ (en particulier dans le cas où Δ n'a pas d'extension). On peut reconnaître ces extensions particulières car ce sont celles qui contiennent au moins une formule ¬Hf sans contenir L¬f. Ceci est intéressant car il n'est pas nécessaire d'éliminer des extensions à cause d'une inconsistance locale, comme en logique des défauts.

Calcul d'extensions et abduction :

Un intérêt de la logique des hypothèses est que les algorithmes de calcul d'extension ne sont pas basés sur le calcul d'un point fixe comme en logique des défauts. En effet, soit F un ensemble de formules, g une formule et E = F ∪ H'Y une extension de F. On a alors que g est vrai dans E si et seulement si F ∪ H'Y |— g. En utilisant le théorème de compacité, il existe donc un sous ensemble fini { Hf1 , .. , Hfn} de HY', tel que :

F ∧ Hf1 ∧... ∧ Hfn |— g.

Donc pour savoir s'il existe une extension E dans laquelle g est vrai, il faudra simplement trouver un ensemble fini d'hypothèses { Hf1, ..., Hfn } tel que :

(a) F ∧ Hf1 ∧... ∧ Hfn est consistant

(b) F ∧ Hf1 ∧... ∧ Hfn |— g

La première condition vient du fait, que par définition, une extension est toujours consistante. Elle se vérifie par un test de satisfaisabilité et peut se réécrire :

(a') F |-/- ¬Hf1 ∨... ∨ ¬Hfn

La seconde condition est une propriété d'abduction, équivalente à :

(b') F ∧ ¬g |— ¬Hf1 ∨... ∨ ¬Hfn

Donc pour trouver si g est vrai dans une extension, il suffit d'ajouter la négation de g à F et de trouver une clause négative, dont tous les littéraux sont des négations d'hypothèses, impliqué par F ∧ ¬g. De plus cette clause négative ne doit pas être impliquée par F seul. L'algorithme va donc être un algorithme d'abduction. Ceci est intéressant car ce même algorithme va permettre à la fois de trouver des extensions et de compléter le réseau de gênes en découvrant de nouvelles interactions.

3.3 Champs de production et algorithme.

Pour formaliser la notion de découverte de nouvelles interactions, on peut utiliser les champs de productions [1] [16]. Si on a un ensemble C de clauses (ou un ensemble F de formules) un champ de production est un ensemble P de formules dont on veut savoir lesquelles sont impliquées par C. Des exemples classiques portent sur la découverte et l'étude des pannes dans un système expert [16].

Une autre utilisation des champs de production porte sur l'abduction dont on vient de voir une application dans 3.2. De manière générale, l'abduction peut se décrire naïvement par : étant donné un ensemble de formules *F* et un résultat cherché *g*, trouver une formule *h* consistante avec F, telle que en ajoutant *h* à *F* on déduit *g* (F∧h |— g). Comme g et F sont connus la découverte de g peut se réécrire en logique par un problème de déduction : trouver h tel que F∧¬g |—¬h et F |-/- ¬h

Dans la pratique h n'est presque jamais une formule quelconque, sa forme syntaxique ou sémantique est précisée par le champ de production. Par exemple dans le cas de la logique des hypothèses le champ de production va être l'ensemble des clauses négatives dont tous les littéraux sont des négations d'hypothèses comme dans 3.2. Ce type de champ de production va se retrouver dans la majeure partie des logiques non-monotones et en particulier les logiques préférentielles [2] , la logiques des défauts et les X-logiques [7].

Une autre application des champs de production va être la recherche de liaisons manquantes dans le graphe des voies de signalisations. On peut chercher à compléter ce graphe avec une seule liaison et le champ de production est alors l'ensemble des clauses négatives unaires du type ¬cause(A,B) ou ¬bloque(A,B). Si on veut compléter avec tous les ensembles de liaisons possibles on pourra prendre comme champ de production l'ensemble des clauses ne contenant que des négations de cause(A,B) et bloque(A,B). Dans ce dernier cas, pour éviter l'explosion combinatoire des résultats il est possible de limiter la longueur des clauses produites. En jouant avec la forme des champs de production et en quittant le strict domaine des clauses bien d'autres utilisations sont possibles.

Pour l'implémentation des algorithmes de production de clauses on été définis dans [1] et [7] avec des bons résultats. La notion de champ de production et les algorithmes liés ont été la base du langage SOLAR développé par l'équipe d'Inoue Katsumi et utilisé dans un cadre de bioinformatique [3] [8] [13].

4. Représentation et utilisation des voies de signalisation.

On va représenter les réseaux de gênes en utilisant la logique des hypothèses et jouer avec cette représentation.

4.1 Représentation.

On se donne donc un réseau qui décrit certaines interactions dans une cellule. On va décrire ce réseau dans une logique des hypothèses TH = { HY, F }.

Pour l'instant on considère que HY contient toutes les formules du type Hf si f est une formule du langage. En fait dans la pratique, cet ensemble va être réduit.

L'ensemble F décrit toutes les informations connues/certaines sur le réseau. Comme les informations sont certaines elles seront traduites naturellement par des formules modales du type **L**f . L'ensemble F va donc contenir en particulier :

- **L**cause(A, B) , si A —>B est un arc du réseau.
- **L**bloque(A, B) , si A —| B est un arc bloquant
- **L**p si une protéine p est activée, et **L**¬p si elle est bloquée.
- On représentera aussi toutes les relations complexes entre protéines. Par exemple si p1 et p2 se lient pour produire une protéine p3, on écrira **L**(p1 ∧p2) —> **L**p3. On pourra aussi représenter les interactions externes avec les UV, avec les médicaments...
- La traduction des défauts normaux d1 et d2 définis dans 2.4 s'écrit naturellement :

d'1 : **L**cause(a, b) ∧ **L**a ∧ **H**b —> **Lb**
d'2 : **L**bloque(a, b) ∧ **L**b ∧ **H**¬b —> **L**¬b

Le langage de représentation est souple (c'est l'intérêt d'une représentation logique). Si par exemple on veut considérer que certaines causalités sont certaines, on ne les traduira pas par des défauts mais par des formules sans hypothèses :

Lcause(a, b) ∧ **L**a —> **Lb**
Lbloque(a, b) ∧ **L**b—> **L**¬b

Bien entendu ce dernier cas peut amener à des contradictions et il faudra alors, avant d'utiliser le réseau, vérifier sa consistance.

4.2 Causalité Abduction et hypothèses

On a maintenant un langage de description pour décrire les réseaux de gênes. Ce langage est assez naturel à condition d'admettre les notions de certitude et d'hypothèse. Il permet de représenter les logiques des défauts et l'incertain. L'intérêt d'un tel langage est qu'il va permettre d'essayer de découvrir de nouvelles voies de signalisation et de constituer des plans d'expériences.

En fait si on a un réseau de gênes, on peut se poser dans un premier temps trois questions :

1) Est ce que le réseau est cohérent?

2) Est ce que dans un certain état du réseau, une protéine p est activée / bloqué?

3) Quelles sont les conditions qui manquent pour activer/ bloquer une protéine?

La première question est simple, c'est de la déduction et plus particulièrement un test de consistance. Le réseau est traduit dans une théorie des hypothèses HH = {HY, W}. Comme aucune hypothèse n'apparait sous forme positive dans la forme normale conjonctive de W, on peut démontrer qu'il suffit simplement de prouver que W est consistant après avoir affecté toutes les hypothèses à faux. On peut aussi sortir du cadre strict de la logique des hypothèses en disant que certaines hypothèses sont vraies, mais les algorithmes sont alors beaucoup moins simples.

La deuxième question est également de la déduction classique. Il faut prouver que p (resps ¬p) est impliqué par l'ensemble W des faits connus : W |— p (resp W |— ¬p). Bien entendu s'il existe deux chaînes de signalisation (de fait deux extensions) l'une donnant p et l'autre ¬p, on ne pourra rien prouver. Et il faudra choisir entre ces extensions.

La troisième question est du domaine de l'abduction. Les conditions manquantes, peuvent être diverses. On peut ajouter au réseau des liens de causalité, des protéines activées ou bloquées, ou encore des hypothèses ; la liste n'est pas exhaustive.

La logique des hypothèses va permette de représenter ces trois possibilités dans le même formalisme. Le champ de production sera composé de toutes les formules qui expriment ces trois données, c'est à dire de toutes les formules qui s'écrivent sur l'ensemble de propositions :

$$PP = \mathbf{H}Y \cup \{\mathbf{L}p\} \cup \{\mathbf{L}\neg p\} \cup \{\mathbf{L}\text{cause}(a, b)\} \cup \{\mathbf{L}\text{bloque}(a, b)\}$$

Bien entendu, le champ de production est ici énorme et il faut le réduire, ou mieux le décrire mieux. C'est l'une des suites du travail à faire.

4.3 Coupure de liaisons.

Une question subsidiaire plus compliquée est de savoir quel(s) arc(s) supprimer dans le réseau de gènes pour obtenir le résultat cherché. Dans ce cas on peut considérer que les liaisons sont également incertaines. Pour exprimer ceci on va remplacer certaines informations certaines du type, **L**cause(a,b) ou **L**bloque(a,b) par des règles avec hypothèses:

H cause(a, b) —> **L**cause(a, b)
H cause(a, b) —> **L**cause(a, b)

On pourrait faire la même chose avec les protéines ou l'absence de protéines en ajoutant des règles du type **H**p —> **L**p et **H**¬p —> **L**¬p. Il s'agit là d'une technique extrême car rien n'est certain, à manier avec précaution car les champs de production sont énormes. Mais elle est intéressante formellement, car elle permet de couper des liaisons sans utiliser des techniques lourdes de suppressions d'information.

5 Conclusion

Nous avons présenté une première approche d'utilisation de la logique des hypothèses et des algorithmes de production de clauses pour représenter, utiliser et compléter les voies de signalisation dans la cellule. Tout ceci se faisant à l'aide d'un même formalisme et d'un même algorithme d'abduction.

Le problème de l'implémentation et de la complexité algorithmique n'a pas été abordé. Il sera sans doute nécessaire de réduire les possibilités d'expression du langage pour arriver à des temps de calcul raisonnables. On retrouve ici la même problématique que pour la résolution pratique des problèmes NP-Complets. Bien entendu le problème est ici plus compliqué, les logiques modales étant dans un cas général PSPACE. Mais dans notre application, on utilise seulement un fragment de ces logiques modales car on ne combine pas les opérateurs modaux. La complexité pratique peut alors rester accessible en contrôlant l'écriture des règles. Par exemple dans un premier temps on peut essayer de se restreindre à quelque chose qui ressemble à des clauses de Horn.

Nous venons de montrer [21] [22] que l'utilisation de la logique des défauts avec une implémentation simple avec Prolog permet de modéliser le comportement humain pour les simulations de combat naval de sous-marins. Les résultats pratiques sont intéressants, une simulation de 4 heures de navigation prend moins d'une seconde sans optimisation du programme. En utilisant dans un premier temps des techniques analogues, ou peut être des ASP, tous les espoirs sont permis.

Une autre question est de donner des bonnes restrictions utile des champs de productions utilisés pour éviter l'explosion combinatoire des résultats. Ici beaucoup reste à faire et l'expérimentation guidera. Mais l'enjeu pratique est important.

Références .

1. Boi J.M, Innocenti E. Rauzy A., P. Siegel P. (92*) Production Fields : a New approach to Deduction Problemss and two Algorithms for propositional Calculus* . Revue d'Intelligence Artificielle, vol 6 - n° 3/1992 , p 235, 255

2. Bossu G., Siegel P. : *Saturation, Nonmonotonic Reasoning and the Closed World Assumption* . Artificial Intelligence 25, Janvier 1985, p. 13-63

3. Doncescu A., Yamamoto Y. and Inoue K. *Biological systems analysis using Inductive Logic Programming*. Proceedings of the 21st International Conference on Advanced Information Networking and Applications (AINA 2007), pages 690-695, IEEE Computer Society, 2007.

4. Doncescu A. , Inoue K. and Yamamoto. K *Knowledge- based discovery in systems biology using CF-induction*. New Trends in Applied Artificial Intelligence: Proceedings of the 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2007), Lecture Notes in Artificial Intelligence, volume 4570, pages 395-404, Springer, 2007

5. Demongeot J. *Multi-stationarity and cell differentiation*. J. Biol. Systems., 6, 1-2 (1998).

6. Farmer M., Zakariya M. *: Enabling e-Technology for bio-informatics*. In: UK-Japan High Technology Industry Forum, London, July2004

7. Forget L, Rish V. , Siegel P. *Preferential logics are X-logics*. Journal of Logic Computational, 10, 2000, pp. 1-13. R[[13] L. Forget and V. Risch and P. Siegel. *Preferential logics are X-logics*. Journal of Logic Computational, 10, 2000, pp. 1-13.

8. Inoue K : *Induction as Consequence Finding*. In: Machine Learning, 55(2):109-135, 2004.

9. Inoue K, Saito H.: *Circumscription Policies for Induction*. In: Proceedings of 14th Conf. on Inductive Logic Programming, LNAI 3194, pp.164-179, Springer, September 2004.

10. D. Kayser, F. Levy : *Modélisation symbolique du raisonnement causal*, Intellecta 2004/1, 38, pp 291-232

11. Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., & Turner, H. (Mars 2004). *Nonmonotonic causal theories*. Artificial Intelligence vol.153 n o 1-2 pp.49-104.

12. Ginsberg, M. L., Smith, D. E. (Juillet 1988). *Reasoning about action II: the qualification problem*. Artificial Intelligence vol.35 n°3 pp.311-342.

13. Nabeshima H. , Iwanuma K., Inoue K. Ray O. : SOLAR: *An automated deduction system for consequence finding*. AI Commun. 23(2-3): 183-203 (2010)

14. Roux-Rouquié M., Hood L., Imbeaud S., Auffray C. *Issues in Computational Methods for Functional Genomics and Systems Biology*. CMSB 2003: 182-186

15. Schwind P. , Siegel P : *Modal logic for Hypothesis Theory,* Fundamentae Informaticae, col 21, n° 1-2 89-101

16. Siegel P. *Représentation et utilisation de la connaissance en calcul propositionnel.* Thèse de Doctorat d'état en Informatique, Université d'Aix-Marseille II, juillet 1987. Sous la direction d'Alain Colmerauer.

17. Siegel P. : *A modal language for Nonmonotonic Reasonning*. Proc. Workshop DRUMS/CEE Marseille 24-27 Février 90.

18. Siegel P. C.Schwind (91) *Hypothesis theory for nonmonotonic reasoning.* First International Proc. Workshop on Nonstandard Queries and Answers, Toulouse, July 1991.

19. Siegel P. , Schwind C. (93) *Modal logic based theory for nonmonotonic reasoning.* Journal of Applied Non classical Logic, vol 3 - n° 1/1993, P 73-92.

20. Tran N. , Baral C. (2007) *Hypothesizing and reasoning about signaling networks*. Journal of Applied Logic 7 (2009) 253-274

21. Toulgoat I. *Modélisation de combat humain dans les simulations de combat naval*. Thèse de doctorat, université de Toulon 31 Janvier 2011

22. Toulgouat I, Siegel, Lacroix Y, Botto J. : *Operator Decision in naval Action's Simulation",* 13th international workshop on Non-Monotonic Reasoning, NMR'10, mai 2010

23. Toulgouat I , Siegel P., Lacroix Y, Botto J : Operator decision modeling : application to a scenario involving two submarines, *in Computer Application and Information Technology in the Maritine Industries,-COMPIT10* Gubbio, Italy, 12-14 April 2010

24. http://fr.wikipedia.org/wiki/Logique_modale