# Différents types de règles pour la reconstruction de réseaux de gènes à partir de données d'expression

Marie Agier

# Différents types de règles pour la reconstruction de réseaux de gènes à partir de données d'expression

Marie Agier

Diagnogène S.A., Clermont-Ferrand
LIMOS, UMR 6158 CNRS, Université Clermont-Ferrand II
agier@isima.fr

***Résumé***

*Les réseaux de gènes ont pour objectif de modéliser les processus biologiques qui contrôlent le développement d'un caractère phénotypique particulier d'un organisme.*

*Dans cet article, nous nous intéressons plus particulièrement à la reconstruction de réseaux de gènes à partir de données d'expression, obtenues grâce à la technique des puces à ADN. Une approche basée sur la découverte de différents types de règles entre gènes est proposée. Plus spécifiquement, nous proposons de générer des règles dont la sémantique est bien-formée, i.e. pour laquelle le système d'Armstrong est juste et complet. Le formalisme de l'approche ainsi que le problème de data-mining sous-jacent sont décrits.*

*Une implémentation de la méthode proposée est disponible sur internet. Elle se présente sous la forme d'un nouveau module intégré au logiciel MeV de TIGR (The Institute for Genomic Research), dédié à l'analyse de données d'expression de gènes.*

***Mots-clés*** *: Règles, Axiomes d'Armstrong, Réseaux de gènes, Données d'expression*

***Abstract***

*Gene networks aim to model the biological processes which control the development of a particular phenotypical nature of an organism. In this article, we are interested more particularly in the reverse engineering of gene networks from gene expression data, obtained thanks to the microarray technology. An approach based on the discovery of different types of rules between genes is proposed. More specifically, we propose to generate rules whose semantics is well-formed, i.e. for which the Armstrong's axiom system is sound and complete. The formalism of the approach as well as the subjacent problem of data-mining are described. An implementation of the method suggested is available on Internet. It is appeared as a new module integrated into the MeV software of TIGR (The Institute for Genomic Research), dedicated to the analysis of gene expression data.*

***Key-words**: Rules, Armstrong's axioms, Gene networks, Expression data*

# 1 INTRODUCTION

Les réseaux de régulation ont pour objectif de modéliser les processus biologiques qui contrôlent le développement d'un caractère phénotypique particulier d'un organisme. La compréhension du fonctionnement de ces réseaux permettra d'appréhender les perturbations liées à des pathologies et de sélectionner les cibles thérapeutiques les plus pertinentes pour traiter efficacement les patients.

Les réseaux de régulation se présentent sous forme de graphes et représentent des interactions entre diverses entités biologiques dans une cellule [7]. Les entités peuvent être de différents types : gènes, ARN, protéines, métabolites... Plusieurs types d'interactions sont également possibles en fonction de la nature des entités qu'elles associent : réaction chimique, catalyse d'une réaction par une enzyme, régulation de l'expression d'un gène... La figure 1 représente un exemple de réseau de régulation[1] avec différentes entités et différentes relations reliant ces entités.

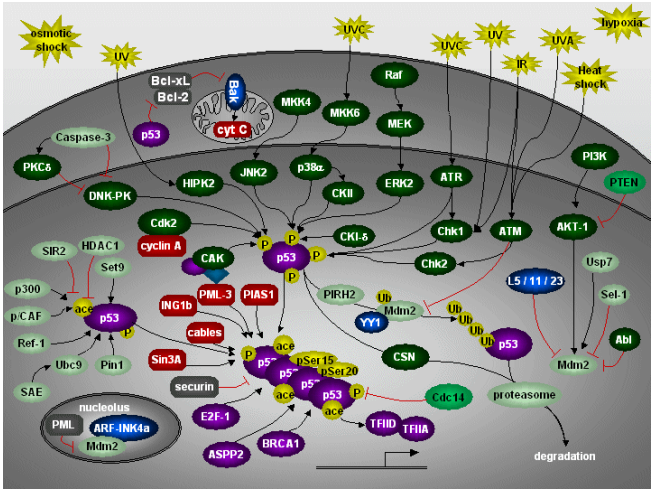



FIG. 1 – Exemple de réseau biologique

Divers types de réseaux plus spécifiques peuvent être étudiés : les réseaux d'interaction protéine-protéine, les réseaux de gènes, les réseaux métaboliques, les réseaux de signalisation... L'objectif étant de comprendre les interactions dans et entre ces différents types de réseaux. En fonction des don-

[1] http ://www.gene-regulation.com

nées disponibles, de la nature des interactions et de la taille des réseaux, différentes méthodes sont utilisées pour inférer ces réseaux.

Dans le cadre de ce travail, nous nous intéressons plus particulièrement à la reconstruction de réseaux de gènes à partir de données d'expression. Diverses méthodes ont déjà été introduites dans ce domaine en particulier. Nous présentons dans cet article une nouvelle approche basée sur la découverte de différents types de règles entre gènes.

L'originalité de notre travail réside dans le fait que nous proposons aux utilisateurs une approche globale pouvant inclure plusieurs sémantiques pour les règles. En effet, nous utilisons la même technique de génération des règles, les même méthodes de post-traitement et de visualisation des règles quelle que soit la sémantique traitée. Ceci est possible grâce à un cadre théorique provenant de la théorie des dépendances fonctionnelles et plus particulièrement du système d'axiomes d'Armstrong. Une implémentation de l'approche proposée est présentée dans ce papier.

Ce travail s'inscrit dans le cadre du projet GeneRules, débuté en 2004, dont le site web est le suivant : http ://www.isima.fr/agier/GeneRules.

***Organisation du papier***

La section 2 décrit brièvement le principe et les objectifs de la reconstruction de réseaux de gènes à partir de données d'expression. La section 3 présente le cadre théorique de notre approche basée sur la découverte de règles entre gènes. La section 4 discute des intérêts pratiques du cadre présenté. L'implémentation de ces travaux est présentée dans la section 5. Enfin, la section 6 conclut et présente quelques perspectives.

## 2 Reconstruction de réseaux de gènes à partir de données d'expression

Suite au succès rencontré par les techniques de puces à ADN pour mesurer l'expression des gènes à grande échelle, l'inférence des réseaux de gènes à partir de ces données d'expression (cf figure 2) a suscité depuis quelques années un intérêt croissant [39, 36, 25, 24, 16].

L'objectif de la reconstruction de réseaux de gènes est de proposer à partir de données expérimentales, des interactions probables entre les gènes, qui pourront être ensuite plus profondément validées avec par exemple des expérimentations plus poussées.

De manière générale, les relations entre les gènes sont rarement directes, i.e. qu'ils n'interagissent pas physiquement. Une relation entre deux gènes

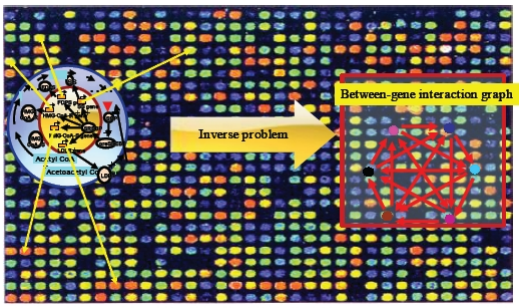



FIG. 2 – Reconstruction de réseaux de gènes à partir de données d'expression

A et B dans un réseau, signifie simplement l'idée qu'un changement dans l'activité du gène A causera un changement dans l'activité du gène B, ce changement étant le résultat d'une succession de modifications d'activité au niveau des produits associés aux deux gènes.

Le principe est donc d'extraire des interactions entre les gènes à partir de données d'expression temporelles ou non, en utilisant des méthodes d'inférence efficaces. Plusieurs approches ont été proposées pour inférer les réseaux de gènes à partir des données d'expression, parmi lesquelles les réseaux booléens, les réseaux bayésiens, l'analyse des corrélations ou les équations différentielles.

Pour les réseaux booléens, les données d'expression temporelles sont binarisées, ainsi à un instant donné, un gène pourra être soit dans l'état $ON$ pour exprimer qu'il est actif, soit dans l'état $OFF$ pour exprimer qu'il est inactif. L'objectif est alors de trouver des règles logiques permettant de déterminer l'état d'un gène à l'instant $t+1$, à partir de l'état de ce gène et des autres gènes à l'instant $t$. Plusieurs méthodes d'inférence des règles logiques, permettant d'obtenir les réseaux, ont été proposées [30, 5, 29].

Les réseaux bayésiens se basent sur le calcul de probabilités [20, 27], en partant de la formule de Bayès sur les probabilités conditionnelles. Pour chaque couple de gènes $\{g_i, g_j\}$, la probabilité d'avoir $g_i$ sachant $g_j$ est calculée. Si cette probabilité est importante, alors on aura une relation de causalité entre ces deux gènes. L'objectif est de trouver pour un jeu de données, le modèle avec le meilleur score i.e. celui ayant la plus grande probabilité d'être correct avec les données étudiées.

Une autre méthode consiste à calculer la corrélation entre chaque paire de gènes, basé sur le coefficient de corrélation de Pearson. Si la valeur absolue de la corrélation est supérieure à un certain seuil alors il existera dans le réseau une relation positive ou négative (selon le signe) entre les deux gènes. Ces réseaux sont connus sous le nom de "Relevance Networks" [13].

Pour les équations différentielles [18], l'évolution du niveau d'expression de chaque gène dans le temps est supposée fonction linéaire des niveaux d'expression des autres gènes. Les coefficients linéaires obtenus représentent l'influence de chaque gène sur la régulation du gène étudié. Le signe du coefficient donne le sens de la relation (positive ou négative) et sa valeur absolue donne la force de l'interaction.

D'autres méthodes d'inférence ainsi que des variantes des méthodes présentées ont émergé ces dernières années, comme les réseaux booléens temporels [35] ou probabilistes [34], les réseaux bayésiens dynamiques...

Ce qu'il faut retenir est que la signification de la relation entre les gènes est très différente d'une méthode à l'autre. Toutes ces méthodes proposent des sémantiques différentes aux réseaux et il est important lorsque l'on étudie un réseau de gènes de garder en mémoire la méthode qui a été utilisée pour le construire. L'objectif commun de toutes ces méthodes est bien de décrire des régularités observées dans les données.

# 3 DÉCOUVERTE DE DIFFÉRENTS TYPES DE RÈGLES ENTRE GÈNES

Dans ce papier, nous nous intéressons à la découverte de règles entre gènes. L'inférence de règles d'association pour la reconstruction de réseaux de gènes a été développée dans plusieurs articles [28, 15, 14]. D'autre part, diverses méthodes d'inférence des dépendances fonctionnelles peuvent être intéressantes pour les données d'expression de gènes lorsque le bruit est pris en compte dans la définition de satisfaction des DF [11, 6]. Nous proposons ici une approche permettant de généraliser ce cadre à *plusieurs* sémantiques pour les règles, compte tenu des divers objectifs biologiques et des caractéristiques possibles des données [3].

Les données d'expression de gènes se présentent sous la forme d'une *relation* dont les *attributs* correspondent aux gènes et les *tuples* correspondent aux échantillons étudiés. Les valeurs réelles représentent le niveau d'expression des gènes pour les différents échantillons (cf table 1).

| r | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ |
|---|---|---|---|---|---|---|
| $t_1$ | 1.7 | 1.5 | 1.2 | -0.3 | 1.4 | 1.6 |
| $t_2$ | 1.8 | -0.7 | 1.3 | 0.8 | -0.1 | 1.7 |
| $t_3$ | -1.8 | 0.4 | 1.7 | 1.8 | 0.6 | -0.4 |
| $t_4$ | -1.7 | -1.4 | 0.9 | 0.5 | -1.8 | -0.2 |
| $t_5$ | 0.0 | 1.9 | -1.9 | 1.7 | 1.6 | -0.5 |

TAB. 1 – Relation $r$ composée de 5 tuples et de 6 gènes

Soit une relation $r$ sur $U$, une *règle* est une expression de la forme $X \rightarrow Y$ qui se lit "X implique Y" avec $X, Y \subseteq U$ deux sous-ensembles d'attributs.

La *sémantique* d'une règle $X \rightarrow Y$ sur $U$ est la *signification*, le *sens* que l'on souhaite donner à la règle : Soit une relation $r$, une règle $X \rightarrow Y$ est dite *satisfaite* dans $r$ avec la sémantique $s$, notée $r \models_s X \rightarrow Y$ si la sémantique de la règle est vraie (ou valide) dans $r$.

Dans l'exemple 1, nous donnons plusieurs règles satisfaites dans la relation $r$ donnée dans la table 1, chacune avec une sémantique particulière.

*Exemple 1* Voici quelques exemples de règles satisfaites dans la relation $r$ en fonction de la sémantique donnée :

1. $g_1 \rightarrow g_3$ : $\forall t \in r$, si $t[g_1] \geq 1.0$ alors $t[g_3] \geq 1.0$.

2. $g_2 \rightarrow g_4$ : $\forall t_i, t_{i+1} \in r$ tel que $t_i$ et $t_{i+1}$ sont deux tuples consécutifs, si $t_{i+1}[g_2] - t_i[g_2] \geq 1.0$ alors $t_{i+1}[g_4] - t_i[g_4] \geq 1.0$.

3. $\{g_4, g_5\} \rightarrow g_6$ : $\forall t \in r$, si $t[g_4] > 0.0$ et $t[g_5] > 0.0$ alors $t[g_6] < 0.0$.

□

Cet exemple simplifié montre qu'il est possible à partir d'un jeu de données d'expression, de générer un grand nombre de réseaux de gènes puisque à chaque sémantique correspond un réseau particulier.

L'objectif de notre approche est donc de proposer aux biologistes la possibilité de générer plusieurs réseaux ou un seul réseau "global" (cf section 4.3) à partir de plusieurs sémantiques, qu'ils choisissent en fonction de leurs objectifs et des caractéristiques de leurs données.

## 3.1 Cadre théorique

A partir des exemples donnés précédemment, nous pouvons dégager deux caractéristiques inhérentes aux sémantiques qui nous seront utiles pour définir cette notion :

- *Les sous-ensembles de la relation sur lesquels la règle s'applique*. Il est possible par exemple d'étudier tous les tuples un à un, de considérer toutes les paires de tuples ou bien uniquement les paires de tuples consécutifs lorsqu'un ordre a été défini (cf exemple 1).
- *Les prédicats qui donnent réellement le sens de la règle* : "**si** $Pred_1$ est vrai pour $X$ **alors** $Pred_2$ est vrai pour $Y$". Notons que ces deux prédicats peuvent être les mêmes.

Les sémantiques données précédemment peuvent alors se caractériser à partir d'un ensemble, qui sera noté $c$, contenant les sous-ensembles $r'$ de $r$ devant être considérés (par exemple, $c(r) = \{\{t\} \mid t \in r\}$), et de deux prédicats $Pred_1$ et $Pred_2$. Les prédicats sont des **conditions** devant être définies sur $X$ (ou $Y$) et $r'$ **seulement**. Aucun autre attribut ou sous-ensemble de $r$ n'est autorisé dans leur définition. En d'autres termes, ils doivent être définis sur $\pi_X(r')$ (ou $\pi_Y(r')$).

Afin de préciser les conditions admissibles, nous en donnons une définition inductive comme suit :

Une **condition simple** sur un ensemble d'attributs $X$ et une relation $r'$, est une expression de la forme : `<terme>` $\theta$ `<terme>`, où :

- $\theta$ est un opérateur de comparaison : $=, <, >, \leq, \geq, \neq$.
- Un `<terme>` est un des éléments suivants (avec $A, B \in X$, $Y \subseteq X$ et $t \in r'$) :
  - Une valeur dans $\pi_X(r')$ : $t[A], t[B], ...$
  - Une constante : $a, b, 8, \varepsilon, null, ...$
  - Une fonction : $fct(r', X), fct(r', A), fct(t, Y)$, ... e.g. $d(t[A], t[B])$ ou $|r'|$.

Une **condition** sur $X$ et $r'$, est une expression composée d'une ou plusieurs **conditions simples** sur $X$ et $r'$, liées grâce aux connecteurs logiques : AND, OR, NOT, (). De plus, les variables $A, B, Y, ...$ (resp. $t$) sont introduites en utilisant les quantifieurs $\forall$ et $\exists$ sur $X$ (resp. $r'$) dans la partie déclarative de la **condition**.

Nous avons ainsi dissocié les caractéristiques propres des sémantiques, de la définition de la satisfaction des règles (cf définition 1). L'exemple 2 ci-dessous reprend les illustrations données dans l'exemple 1 et précise les sémantiques sous jacentes.

*Exemple 2* Les trois sémantiques peuvent être redéfinies avec les ensembles et prédicats suivants :

1. – $c(r) = \{\{t\} | t \in r\}$.
   – $Pred_1(X, \{t\}) = Pred_2(X, \{t\}) = [\, \forall A \in X, t[A] \geq 1.0\,]$.

2. – $c(r) = \{\{t_i, t_j\} | t_i, t_j \in r$ AND $t_j[time] = t_i[time] + 1\}$ (où $time$ est un attribut externe donnant l'ordre des tuples).
   – $Pred_1(X, \{t_i, t_j\}) = Pred_2(X, \{t_i, t_j\}) = [\, \forall A \in X, t_j[A] - t_i[A] \geq 1.0\,]$.

3. – $c(r) = \{\{t\} | t \in r\}$.
   – $Pred_1(X, \{t\}) = [\, \forall A \in X, t[A] > 0.0\,]$.
   – $Pred_2(X, \{t\}) = [\, \forall A \in X, t[A] < 0.0\,]$.

□

Dans la suite, nous dénotons par $C$ la classe de l'ensemble des sémantiques pouvant être définies par un ensemble $c$ et deux prédicats $Pred_1$ et $Pred_2$ définis pour un ensemble d'attributs.

Nous définissons à présent la satisfaction des règles pour les sémantiques appartenant à la classe $C$ :

**Définition 1**
*Soient $r$ une relation définie sur $U$, $X, Y \subseteq U$ deux ensembles d'attributs et $s \in C$ une sémantique caractérisée par $c$, $Pred_1$ et $Pred_2$.*
*La règle $X \rightarrow Y$ est **satisfaite** dans $r$ pour la sémantique $s$, notée $r \models_s X \rightarrow Y$ si et seulement si :*

$$\forall r' \in c(r),$$
$$si\ Pred_1(X, r')\ est\ vrai\ alors\ Pred_2(Y, r')\ est\ vrai.$$

***Remarque***

Certaines sémantiques sont définies de telle sorte que quelle que soit la relation $r$ et quels que soient les ensembles d'attributs $X$ et $Y$, toutes les règles $X \rightarrow Y$ sont fausses (resp. vraies). En pratique, ces sémantiques que nous appellerons sémantiques *pathologiques*, ne présentent aucun intérêt et ne seront pas prises en compte. Leur définition formelle est donnée dans la suite :

**Définition 2**
*Soit $s \in C$ une sémantique caractérisée par un ensemble $c$ et deux prédicats $Pred_1$ et $Pred_2$. La sémantique $s$ est dite **pathologique** si pour toute relation $r$ sur $U$, une des conditions suivantes est vraie :*
- *$c(r)$ est égal à l'ensemble vide.*
- *$\forall X \subseteq U$ et $\forall r' \in c(r)$, $Pred_1(X, r')$ est vrai et $Pred_2(X, r')$ est faux.*
- *$\forall X \subseteq U$ et $\forall r' \in c(r)$, $Pred_1(X, r')$ est faux.*
- *$\forall X \subseteq U$ et $\forall r' \in c(r)$, $Pred_2(X, r')$ est vrai.*

Dans la suite, les notations suivantes sont utilisées :
- $C_P$ dénote la classe des sémantiques pathologiques de $C$, i.e. $C_P \subset C$.
- $C_X$ dénote la classe des sémantiques non pathologiques de $C$, i.e. $C_X = C \setminus C_P$.

## 3.2 Sémantiques bien-formées

Notre approche a la particularité de prendre en compte un nombre important de sémantiques. Plus formellement, nous nous intéressons à un groupe particulier de sémantiques : les sémantiques bien-formées. Ces sémantiques ont la particularité de satisfaire des propriétés intéressantes venant de la théorie des dépendances fonctionnelles et plus particulièrement des axiomes d'Armstrong [32, 17] :

– Il est tout d'abord possible de **raisonner** sur les règles à partir des axiomes d'Armstrong sans accéder aux données. A partir d'un ensemble de règles $F$, il est possible de savoir si une règle est **impliquée** par cet ensemble de règles en temps linéaire [8]. Ainsi, si on dispose d'une relation $r$ qui satisfait $F$ alors on sait que toutes les règles pouvant être déduites de $F$ par les axiomes d'Armstrong seront satisfaites dans cette relation.
– Nous pouvons également travailler sur des "petites" **couvertures** des règles [31, 23] et proposer un processus de découverte spécifique à la couverture considérée mais applicable à *toute* sémantique bien-formée, ce qui laisse entrevoir une grande généricité dans le traitement opérationnel.
– De plus, il est aussi possible de proposer des couvertures pour les règles approximatives [22].

Les sémantiques bien-formées sont définies de la façon suivante :

**Définition 3**
*Une sémantique $s$ est* bien-formée *si les axiomes d'Armstrong sont justes et complets pour $s$.*

Malheureusement mais de façon non surprenante, il n'y a pas d'équivalence entre la classe des sémantiques bien-formées et la classe $C_X$ que nous avons définie auparavant, i.e. la classe des sémantiques non pathologiques de $C$.

Pour pallier ce problème, nous donnons quelques restrictions syntaxiques sur la définition 1 permettant de nous assurer qu'une sémantique est bien-formée [2]. En d'autres termes, étant donnée une sémantique, il n'est plus nécessaire de montrer la justesse et la complétude du système d'axiomes d'Armstrong mais il suffit de montrer que la définition de cette sémantique est en accord avec ces nouvelles restrictions syntaxiques.

Nous définissons alors $C_A \subseteq C_X$ la classe des sémantiques pouvant être caractérisées par un ensemble $c$ et un prédicat $Pred$ défini pour un seul attribut.

Pour les sémantiques appartenant à la classe $C_A$, la définition de la satisfaction des règles devient :

**Définition 4**
*Soient $r$ une relation définie sur $U$, $X, Y \subseteq U$ deux ensembles d'attributs et $s \in C_A$ une sémantique caractérisée par $c$ et $Pred$.*
*La règle $X \rightarrow Y$ est* ***satisfaite*** *dans $r$ pour la sémantique $s$, notée $r \models_s X \rightarrow Y$ si et seulement si :*

$$\forall r' \in c(r),$$

$si\ \forall A \in X, Pred(A, r')\ est\ vrai\ alors\ \forall A \in Y, Pred(A, r')\ est\ vrai.$

La différence est double par rapport à la définition 1 :
- Premièrement, il n'y a plus qu'un seul prédicat et non plus deux comme précédemment.
- Deuxièmement, une restriction est posée sur le prédicat : Il doit être satisfait pour chaque **attribut** $A \in X$ plutôt que d'être satisfait par **l'ensemble d'attributs** $X$.

Notons que deux prédicats peuvent être syntaxiquement différents et être pourtant équivalents. Nous donnons dans la suite la définition de l'équivalence entre deux prédicats :

**Définition 5**
*Soit $s \in C_X$ une sémantique caractérisée par un ensemble $c$ et deux prédicats $Pred_1$ et $Pred_2$. Les deux prédicats $Pred_1$ et $Pred_2$ sont dits **équivalents**, notés par $Pred_1 \equiv Pred_2$, si et seulement si pour toute relation $r$ sur $U$ et pour tout ensemble d'attributs $X \subseteq U$, nous avons :*
$\{r' \in c(r) \mid Pred_1(X, r')$ *est vrai*$\} = \{r' \in c(r) \mid Pred_2(X, r')$ *est vrai*$\}$.

Comme nous le souhaitions, nous avons maintenant une équivalence entre cette nouvelle classe de sémantiques $C_A$ et l'ensemble des sémantiques bien-formées :

**Théorème 1** Soit $s \in C_X$ une sémantique. La sémantique $s$ est bien-formée si et seulement si $s \in C_A$.

**Preuve 1** Soit $s \in C_X$ une sémantique. Nous devons d'abord montrer que si $s \in C_A$ alors $s$ est bien-formée puis que si $s$ est bien-formée alors $s \in C_A$ ou de façon équivalente que si $s \notin C_A$ alors $s$ n'est pas bien-formée. Une preuve de ce théorème est donnée dans [1].

Ce théorème permet donc de montrer rapidement qu'une sémantique est ou non bien-formée, comme par exemple pour les sémantiques de l'exemple 2.

*Exemple 3* Les deux premières sémantiques peuvent être caractérisées par les ensembles et les prédicats suivants, ce qui suffit pour montrer que ces deux sémantiques sont bien-formées :

1. – $c(r) = \{\{t\} | t \in r\}$.
   – $Pred(A, \{t\})[\, t[A] \geq 1.0\,]$.

2. – $c(r) = \{\{t_i, t_j\} | t_i, t_j \in r$ AND $t_j[time] = t_i[time] + 1\}$.
   – $Pred(A, \{t_i, t_j\}) = [\, t_j[A] - t_i[A] \geq 1.0\,]$.

Par contre, la troisième sémantique n'est pas bien-formée puisque les deux prédicats $Pred_1$ et $Pred_2$ ne sont pas équivalents.

□

# 4 INTÉRÊTS PRATIQUES

Nous avons défini un cadre théorique permettant d'inclure un grand nombre de sémantiques : l'ensemble des sémantiques bien-formées. Nous allons voir maintenant les intérêts pratiques de ce cadre de travail dans le processus de découverte des règles et des réseaux puisque, en effet, les trois étapes principales sont identiques quelle que soit la sémantique choisie : la génération des règles, le post-traitement et la visualisation des règles.

## 4.1 Génération des règles

Le problème de génération des règles a été largement étudié dans le contexte des règles d'association [4] et des dépendances fonctionnelles [17, 26]. Notre objectif ici n'est pas de développer de nouveaux algorithmes pour chaque sémantique mais au contraire d'uniformiser la génération des règles pour toute sémantique bien-formée. Nous souhaitons également nous baser sur les propriétés définies pour les règles d'association ou les dépendances fonctionnelles.

Rappelons tout d'abord, qu'une correspondance existe entre un ensemble de règles et un système de fermeture [21] (un système de fermeture $C$ sur $U$ est tel que $U \in C$ et $\forall X, Y \in C$, $X \cap Y \in C$). Deuxièmement, étant données une relation $r$ et une sémantique bien-formée $s$, il est toujours possible de définir un système de fermeture par rapport à l'ensemble des règles satisfaites dans la relation $r$ avec la sémantique $s$.

Deux techniques principales existent pour calculer une couverture des règles :
- Les méthodes qui énumèrent le système de fermeture pour générer par exemple une couverture minimum [23], généralement utilisées pour la génération des règles d'association [40].
- Les méthodes qui évitent l'énumération du système de fermeture pour générer la couverture canonique, généralement utilisées pour l'inférence des dépendances fonctionnelles [32, 17].

Dans les deux cas, la première étape consiste à calculer une *base* du système de fermeture, étape primordiale durant laquelle a lieu l'accès aux données. Le cadre théorique que nous proposons nous permet de calculer de façon identique une base du système de fermeture pour toute sémantique bien-formée :

**Définition 6**
*Soient $r$ une relation définie sur $U$ et $s$ une sémantique bien-formée caractérisée par l'ensemble $c$ et le prédicat $Pred$.*
*Soit $B_s(r)$ l'ensemble défini de la façon suivante :*

$$B_s(r) = \bigcup_{r' \in c(r)} \{A \in U \mid Pred(A, r') \; est \; vrai\}.$$

Nous avons alors le résultat suivant qui étend à notre contexte, un résultat bien connu obtenu dans le cadre des dépendances fonctionnelles [9] :

**Proposition 1** $B_s(r)$ est une *base* du système de fermeture par rapport à l'ensemble $F_s(r)$ des règles satisfaites dans $r$ pour la sémantique $s$.

**Preuve 2** La preuve de cette proposition est donnée dans la version étendue du papier.

Le calcul de la base $B_s(r)$ est une étape primordiale du processus de génération puisque c'est lors de cette étape qu'a lieu l'accès aux données.

*Exemple 4* Considérons la première sémantique donnée dans l'exemple 1 et la relation $r$ donnée dans la table 1.
La base du système de fermeture $B_s(r)$ se calcule de la façon suivante :

$$B_s(r) = \bigcup_{t \in r} \{\ A \in U \mid t[A] \geq 1.0\ \}.$$

Pour cette relation $r$, nous avons :

$$B_s(r) = \{\{g_1, g_2, g_3, g_5, g_6\},\\ \{g_1, g_3, g_6\}, \{g_3, g_4\}, \{\}, \{g_2, g_4, g_5\}\}.$$

Par exemple, une couverture minimale des règles peut être calculée à partir de la base $B_s(r)$ avec les règles suivantes :
$\{g_1 \rightarrow \{g_3, g_6\}$, $\{g_1, g_4\} \rightarrow g_2$, $g_2 \rightarrow g_5$, $\{g_2, g_3\} \rightarrow g_1$, $g_5 \rightarrow g_2$, $g_6 \rightarrow g_1\}$.

□

## 4.2 Post-traitement des règles

Plusieurs indices de qualité ont été introduits pour les règles d'association [4, 12, 33] afin de mesurer la pertinence des règles obtenues. Le cadre théorique que nous proposons nous permet d'étendre la définition de la plupart des indices existants à toute sémantique bien-formée.

En effet, la plupart des indices de qualité pour une règle $X \rightarrow Y$ sont calculés à partir de quatre valeurs : $n$, $n_X$, $n_Y$, $n_{XY}$ dont la définition peut être adaptée à l'ensemble des sémantiques bien-formées :

- $n$ : nombre de sous-ensembles $r' \in c(r)$.
- $n_X$ : nombre de sous-ensembles $r' \in c(r)$ et tels que $\forall A \in X$, $Pred(A, r')$ est vrai.
- $n_Y$ : nombre de sous-ensembles $r' \in c(r)$ et tels que $\forall A \in Y$, $Pred(A, r')$ est vrai.

– $n_{XY}$ : nombre de sous-ensembles $r' \in c(r)$ et tels que $\forall A \in X \cup Y$, $Pred(A, r')$ est vrai.

A partir de ces paramètres, cinq indices de qualité sont calculés pour chaque règle $X \rightarrow Y$ de la façon suivante :

Support ($X \rightarrow Y$) = Support ($Y \rightarrow X$) = $P(XY) = n_{XY}/n$.
Le support correspond à la probabilité que X et Y soient simultanément satisfaits [4].

Confiance ($X \rightarrow Y$) = $P(XY|X) = n_{XY}/n_X$.
La confiance correspond à la probabilité que Y soit satisfait sachant que X est satisfait. Lorsque la confiance est égale à 1, la règle est dite *exacte*, sinon elle est dite *approximative* [4].

Lift ($X \rightarrow Y$) = Lift ($Y \rightarrow X$) = $P(XY)/P(X)P(Y) = (n_{XY} * n)/(n_X * n_Y)$.
Le lift mesure la dépendance entre X et Y. Il correspond au rapport entre la probabilité réelle d'avoir X et Y satisfaits et la probabilité attendue si X et Y étaient statistiquement indépendants [12].

Leverage ($X \rightarrow Y$) = Leverage ($Y \rightarrow X$) = P(XY) - P(X)*P(Y) = $(n_{XY}/n) - ((n_X/n) * (n_Y/n))$.
Le leverage mesure aussi la dépendance entre X et Y, mais mesure cette fois la différence entre les deux probabilités [33].

Conviction ($X \rightarrow Y$) = (P(X)*P(nonY)) / P(X et nonY) = $(n_X * (n - n_Y))/(n * (n_X - n_{XY}))$.
La conviction compare la probabilité attendue d'avoir X satisfait et Y non satisfait s'ils étaient indépendants avec la probabilité réelle d'avoir X satisfait et Y non satisfait [12].

Notons que d'autres indices, comme ceux définis dans [37] pourraient être ajoutés sans grande difficulté.

## 4.3 Visualisation des règles

Plusieurs approches de visualisation des règles ont été proposées, essentiellement pour les règles d'association.

La représentation la plus classique et la plus répandue est la représentation textuelle des règles. Des indices de qualité sont donnés pour chaque règle permettant par exemple de trier les règles en fonction de leur pertinence. Toutefois, cette méthode ne permet pas une vision globale de l'ensemble des règles.

Des représentations par matrice ont ensuite émergées : les matrices itemset-itemset et les matrices itemset-règles [38]. Dans le premier cas, les colonnes correspondent aux parties gauches des règles et les lignes correspondent aux parties droites tandis que dans le second cas, les colonnes correspondent aux

itemsets qui composent les parties droites et gauches des règles et à chaque ligne correspond une règle. Les indices de qualité sont représentés grâce à des effets de couleur, de forme ou de taille. Ces approches nécessitent un effort considérable de la part des experts puisque cette visualisation n'est pas très intuitive.

Dans les représentations par graphe, les noeuds représentent les itemsets et les arcs représentent les règles. Les indices de qualité peuvent être mis en avant en augmentant la taille des noeuds ou des arêtes, en utilisant plusieurs couleurs ou plusieurs formes. Ces méthodes de visualisation sont très intéressantes dès lors que le nombre de règles n'est pas très important. Lorsque celui-ci augmente, les arêtes se mélangent rapidement et la visualisation est vite illisible. Toutefois, ces approches de visualisation sont très intuitives et sont particulièrement familières aux experts du domaine étudié i.e. les biologistes.

D'autres méthodes de visualisation ont été proposées comme par exemple la représentation 3D par réalité virtuelle [10].

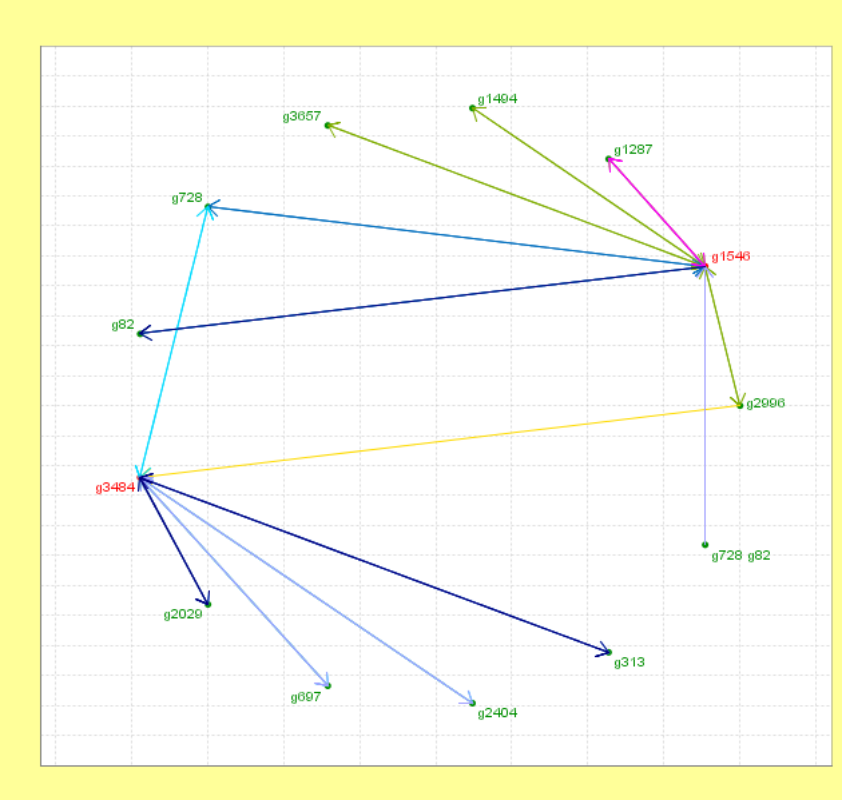



FIG. 3 – Réseau global avec plusieurs sémantiques et deux gènes centraux

Notre objectif est avant tout de réaliser un outil convivial pour les biologistes afin de leur permettre de visualiser les relations entre les gènes. Nous avons donc opté tout naturellement pour une visualisation par graphe (cf figure 3) qui nous semble être la méthode la plus facile à interpréter et la plus intuitive pour les experts, notre objectif étant aussi de nous positionner dans le cadre de la reconstruction de réseaux de gènes.

Toutefois, un des inconvénients de ce type de représentations est que les réseaux deviennent vite illisibles lorsque le nombre de règles est important. Or la phase de post-traitement des règles proposée permet de diminuer considérablement leur nombre pour ne garder que les règles les plus intéressantes en fonction des divers indices de qualité.

Le second problème posé par ce type de représentations est la visualisation de règles composées de plusieurs items. Or, dans le cadre spécifique de la découverte de règles entre gènes, l'interprétation des règles est une étape particulièrement délicate et très difficile pour les biologistes, étant donné qu'une règle entre deux gènes signifie rarement une implication directe mais plutôt une suite d'interactions entre les gènes et leurs divers produits associés (protéines, facteurs de transcription...). C'est pourquoi les biologistes sont rarement intéressés par les règles avec plus de 3 ou 4 items, l'interprétation devenant très vite impossible.

Cette représentation permet par contre, de mettre en avant quelques gènes d'intérêt que les utilisateurs voudront voir apparaître en parties gauches ou droites des règles. Les biologistes pourront ainsi visualiser l'ensemble des règles associées à ces quelques gènes qui les intéressent plus particulièrement.

## 5 IMPLÉMENTATION

Nous avons choisi de développer un outil convivial afin de faciliter son utilisation par les biologistes. Pour cela, un nouveau module appelé RG (Rule Generation) a été intégré à un logiciel gratuit et open-source consacré à l'analyse de données d'expression de gènes, le logiciel MeV (MultiExperiment-Viewer) [19].

Cet outil fait partie d'une suite de quatre logiciels, appelée TM4, développée par The Institute for Genomic Research (TIGR). Cette suite est entièrement consacrée au traitement des données issues de biopuces : stockage, analyse des images, normalisation, analyse statistique et informatique des données... Le logiciel MeV est l'application consacrée à cette dernière étape, sa convivialité fait de cet outil, un des plus utilisés actuellement.

La version étendue du logiciel MeV avec le module RG est disponible sur le site http ://www.isima.fr/agier/GeneRules.

La découverte des règles au sein du module RG s'opère en différentes étapes :

1. **Caractérisation des données**

   Tout d'abord, il est possible de préciser si les expériences appartiennent à différents groupes et si les données sont temporelles. En fonction des caractéristiques des données, les sémantiques adaptées seront proposées.

2. **Choix des gènes centraux et des sémantiques**

   Les biologistes spécifient ensuite les gènes qu'ils veulent voir apparaître en partie gauche et en partie droite des règles (cf figure 4). Cette étape permet de se concentrer sur les gènes d'intérêt. Ensuite se fait le choix des sémantiques et des seuils associés en fonction des objectifs de l'étude.

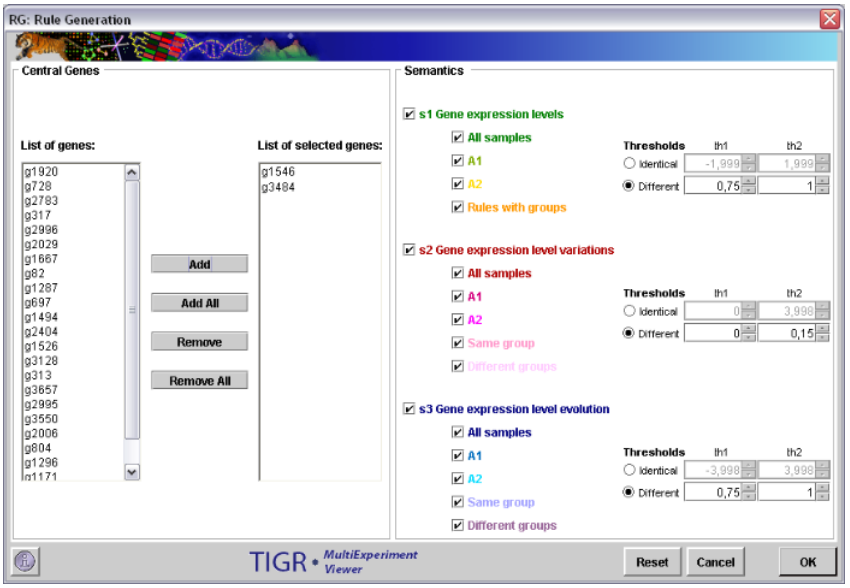



FIG. 4 – Choix des gènes centraux et des sémantiques

3. **Génération des règles**

   L'inférence des règles commence avec le calcul de la base du système de fermeture (cf section 4).

4. **Post-traitement des règles**

   Un filtre est possible à partir des cinq indices de qualité proposés : support, confiance, lift, leverage et conviction (cf figure 5).

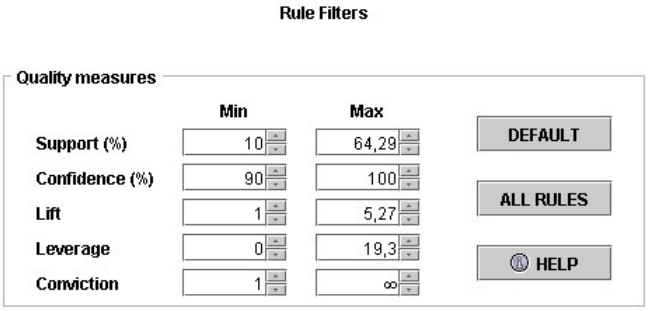
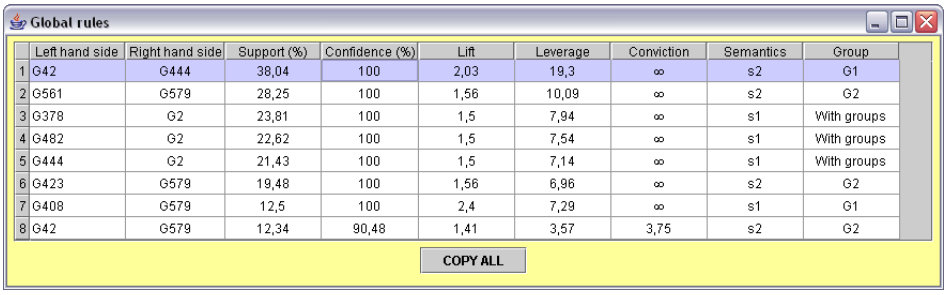



FIG. 5 – Filtrage des règles par indices de qualité

5. **Visualisation des règles**

   Les règles obtenues peuvent être visualisées sous forme de réseau global (cf figure 3) regroupant des règles obtenues avec différentes sémantiques, qui sont représentées par diverses couleurs, ou bien sous forme textuelle avec le détail des différents indices de qualité (cf figure 6).




| | Left hand side | Right hand side | Support (%) | Confidence (%) | Lift | Leverage | Conviction | Semantics | Group |
|---|---|---|---|---|---|---|---|---|---|
| 1 | G42 | G444 | 38,04 | 100 | 2,03 | 19,3 | ∞ | s2 | G1 |
| 2 | G561 | G579 | 28,25 | 100 | 1,56 | 10,09 | ∞ | s2 | G2 |
| 3 | G378 | G2 | 23,81 | 100 | 1,5 | 7,94 | ∞ | s1 | With groups |
| 4 | G482 | G2 | 22,62 | 100 | 1,5 | 7,54 | ∞ | s1 | With groups |
| 5 | G444 | G2 | 21,43 | 100 | 1,5 | 7,14 | ∞ | s1 | With groups |
| 6 | G423 | G579 | 19,48 | 100 | 1,56 | 6,96 | ∞ | s2 | G2 |
| 7 | G408 | G579 | 12,5 | 100 | 2,4 | 7,29 | ∞ | s1 | G1 |
| 8 | G42 | G579 | 12,34 | 90,48 | 1,41 | 3,57 | 3,75 | s2 | G2 |

FIG. 6 – Représentation textuelle des règles

# 6 CONCLUSION

La reconstruction de réseaux de gènes à partir de données d'expression est un problème traité par de nombreuses équipes actuellement. L'enjeu est très important puisque la découverte de ces réseaux permet une meilleure compréhension des relations entre les gènes à l'intérieur des cellules.

L'approche que nous proposons pour l'inférence de ces réseaux, est basée sur la notion de règles entre gènes. Cette approche permet de construire différents réseaux en fonction de la sémantique choisie pour les règles et offre également la possibilité de générer des réseaux "globaux" incluant plusieurs sémantiques.

Les points clés de notre approche est la possibilité de considérer des données temporelles ou non, selon la sémantique choisie ou bien encore de prendre en compte des informations externes comme la présence de groupes d'échantillons. Le système d'axiomes d'Armstrong nous offre également de bonnes propriétés, comme par exemple la possibilité de raisonner sur les règles.

Le module que nous avons implémenté est fonctionnel, l'objectif à terme étant d'ajouter d'autres méthodes d'inférence des réseaux afin de proposer aux biologistes un outil complet. Ceux-ci pourraient en effet générer à partir d'un même jeu de données, différents réseaux et mesurer alors la "force" des relations obtenues.

## RÉFÉRENCES


[1] M. Agier. *De l'analyse de données d'expression à la reconstruction de réseaux de gènes*. PhD thesis, Université Clermont 2, December 2006.

[2] M. Agier et J-M. Petit. A new and useful syntactic restriction on rule semantics for tabular data. In *Proceedings of the* $21^{th}$ *Bases de Données Avancées*, pages 135–150, Saint-Malo, France, 2005.

[3] M. Agier, J-M. Petit et E. Suzuki. Towards ad-hoc rule semantics for gene expression data. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pages 494–503, Saratoga Springs, New-York, USA, 2005. Springer-Verlag.

[4] R. Agrawal, T. Imielinski et A.N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington DC, USA, 1993. ACM Press.

[5] T. Akutsu, S. Miyano et S. Kuhara. Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Journal of Computational Biology*, 7 :331–343, 2000.

[6] A. Aussem et J-M. Petit. $\epsilon$-functional dependency inference : application to dna microarray expression data. In P. Pucheral, éditeur, *Proceedings of the Bases de Données Avancées*, Evry, France, 2002.

[7] Z. Bar-Joseph, G. K. Gerber, T. I. Lee, N. J. Rinaldi, J. Y. Yoo, F. Robert, D. B. Gordon, E. Fraenkel, T. S. Jaakkola, R. A. Young et D. K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11) :1337–1342, 2003.

[8] C. Beeri et P.A. Berstein. Computational problems related to the design of normal form relation schemes. *ACM Transaction on Database Systems*, 4(1) :30–59, 1979.

[9] C. Beeri, M. Dowd, R. Fagin et R. Statman. On the structure of Armstrong relations for functional dependencies. *Journal of the ACM*, 31(1) :30–46, 1984.

[10] J. Blanchard, F. Guillet et H. Briand. A user-driven and quality-oriented visualization for mining association rules. In *Proceedings of the IEEE International Conference on Data Mining*, pages 493–496, 2003.

[11] P. Bosc, D. Dubois et H. Prade. Fuzzy functional dependencies -an overview and a critical discussion. *Journal of the American Society for Information Science*, 49 :217–235, 1998.

[12] S. Brin, R. Motwani, J.D. Ullman et S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 255–264, 1997.

[13] A.J. Butte, P. Tamayo, D. Slonim, T.R. Golub et I.S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97 :12182–12186, 2000.

[14] G. Cong, A.K.H. Tung, X. Xu, F. Pan et J. Yang. Farmer : Finding interesting rule groups in microarray datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 143–154, 2004.

[15] C. Creighton et S. Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19 :79–86, 2003.

[16] H. de Jong et D. Ropers. Qualitative approaches towards the analysis of genetic regulatory networks. *System Modeling in Cellular Biology : From Concepts to Nuts and Bolts*, pages 125–148, 2006.

[17] J. Demetrovics et V.D. Thi. Some remarks on generating Armstrong and inferring functional dependencies relation. *Acta Cybernetica*, 12(2) :167–180, 1995.

[18] P. D'Haeseleer, S. Liang et R. Somogyi. Gene expression data analysis and modelling. In *Proceedings of the Pacific Symposium on Biocomputing*, Hawai, 1999.

[19] AI. Saeed et al. TM4 : a free, open-source system for microarray data management and analysis. *Biotechniques*, 34(2) :374–78, 2003.

[20] N. Friedman, M. Linial, I. Nachman et D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7 :601–620, 2000.

[21] B. Ganter et R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.

[22] G. Gottlob et L. Libkin. Investigations on Armstrong relations, dependency inference, and excluded functional dependencies. *Acta Cybernetica*, 9(4) :385–402, 1990.

[23] J-L. Guigues et V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Math. Sci. Humaines*, 24(95) :5–18, 1986.

[24] AJ. Hartemink. Reverse engineering gene regulatory networks. *Nature Biotechnology*, 23(5) :554–555, 2005.

[25] PM. Haverty, MC. Frith et Z. Weng. Carrie web service : automated transcriptional regulatory network inference and interactive analysis. *Nucleic Acids Research*, 32(Web-Server-Issue) :213–216, 2004.

[26] Y. Huhtala, J. Kärkkäinen, P. Porkka et H. Toivonen. Efficient discovery of functional and approximate dependencies using partitions. In *Proceedings of the* $14^{th}$ *IEEE International Conference on Data Engineering*, pages 392–401, 1998.

[27] D. Husmeier. Reverse engineering of genetic networks with bayesian networks. *Biochemical Society Transactions*, 31 :1516–1518, 2003.

[28] A. Icev, C. Ruiz et E.F. Ryder. Distance-enhanced association rules for gene expression. In *Proceedings of the Workshop on Data Mining in Bioinformatics BIOKDD*, Washington DC, USA, 2003.

[29] T.E. Ideker, V. Thorsson et R.M. Karp. Discovery of regulatory interactions through perturbation : Inference and experimental design. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 5, pages 302–313, 2000.

[30] S. Liang, S. Fuhrman et R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 18–29, 1998.

[31] D. Maier. Minimum covers in the relational database model. *Journal of the ACM*, 27(4) :664–674, 1980.

[32] H. Mannila et K-J. Räihä. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12(1) :83–99, 1994.

[33] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.

[34] I. Shmulevich, E.R. Dougherty, S. Kim et W. Zhang. Probabilistic boolean networks : A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2) :261–274, 2002.

[35] A. Silvescu et V. Honavar. Temporal boolean network models of genetic networks and their inference from gene expression time series. *Complex Systems*, 13 :54–70, 2001.

[36] LA. Soinov, MA. Krestyaninova et A. Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology*, 4 :R6.1–R6.10, 2003.

[37] P-N. Tan, V. Kumar et J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4) :293–313, 2004.

[38] P. C. Wong, P. Whitney et J. Thomas. Visualizing association rules for text mining. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 120–123, 1999.

[39] X. Wu, Y. Ye et L. Zhang. Graphical modeling based gene interaction analysis for microarray data. *SIGKDD Explorations*, 5(2) :91–100, 2003.

[40] M.J. Zaki. Generating non-redundant association rules. In *Proceedings of the* $6^{th}$ *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34–43. ACM Press, 2000.