Vers une modélisation booléenne des règles d'association

Abdelhak Mansoul et Baghdad Atmani,

Equipe de recherche SIF « Simulation, Intégration et Fouille de données »
Laboratoire d'Informatique d'Oran - LIO
Département Informatique, Faculté des Sciences, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie
mans_abdel@yahoo.fr, atmani.baghdad@gmail.com

Résumé. L'extraction de règles d'association est une tâche populaire en fouille de données. L'une des premières méthodes utilisées est la méthode GUHA initiée par Hajek, Havel et Chytil [9], ensuite l'algorithme APRIORI, issu des travaux d'Agrawal, Imielinski, Swami et Srikant [6], [7]. Seulement ces méthodes produisent une trop grande masse de règles, ce qui induit un coût considérable en volume et en temps de traitement.
Dans cet article, nous proposons une extraction de règles d'association assistée par une modélisation booléenne des résultats obtenus, dans le but de parer aux inconvénients cités auparavant. Ceci, est dans la perspective de recherche d'un processus de génération de règles d'association par inférence, intégré dans un automate cellulaire.

Mots clés: Automate cellulaire, Fouille de données biologiques, Induction de règles, Règle d'association, modélisation booléenne.

1 Introduction

Parmi les travaux de recherche en fouille de données, l'extraction de règles d'association est sans doute la technique qui a attiré le plus l'attention des chercheurs et pour laquelle beaucoup de travaux ont été effectués [6], [7], [9], [10], [11]. Nous avons essayé de concevoir un processus assez novateur en se basant principalement sur le principe de représentation cellulaire des règles d'association afin de parer aux insuffisances liées à cette méthode de fouille de données.

Ce processus s'effectue en 3 étapes :

1. extraction de motifs fréquents et génération des règles d'association en utilisant l'algorithme Apriori [7];
2. modélisation booléenne des règles d'association par la machine CARI « Cellular Automaton for Rules Induction » ;
3. gestion des règles par le moteur d'inférence de CARI.

La structure des séquences biologiques. Les séquences biologiques expérimentales que nous utilisons ne sont pas dans leurs structures primaires à base de nucléotides

(ex : AAGTCGTTGCTGGC). Elles se présentent sous la forme de fichiers textes et dans des formats de données spécifiques (FASTA, STADEN, etc.) [2], [12] et contiennent des entités sémantiques (le gène, sa localisation, …..) (Fig. 1).

Nous exploiterons donc ce format pour définir un prétraitement spécifique et obtenir une structure bien appropriée à la fouille de données, où les entités sémantiques deviennent des descripteurs potentiels.

```
1: aac
 aminoglycoside 2-N-acetyltransferase [Mycobacterium tuberculosis CDC1551]
 Other Aliases: MT0275
 Annotation: NC_002755.2 (314424..314969, complement)
 GeneID: 923198
2: accD
 acetyl-CoA carboxylase, carboxyl transferase, beta subunit [Mycobacterium tuberculosis CDC1551]
 Other Aliases: MT0927
 Annotation: NC_002755.2 (1006705..1008192, complement)
 GeneID: 926242
……...
```

Fig. 1. Morceau d'une séquence génomique de la bactérie modèle [12].

Donc, la problématique abordée dans ce papier, est la recherche de règles d'association sur des données biologiques (séquences génomiques et protéiques), d'une bactérie modèle appelée : Mycobacterium Tuberculosis, avec post-traitement des résultats obtenus, par un automate cellulaire afin d'avoir une base de règles optimisée et des temps de traitements assez réduits.

2 Architecture du système BIODM

Notre système BIODM est composé de deux grands modules ERAB et CARI. Le module ERAB, acronyme d'Extraction de Règles d'Association à partir de données Biologiques, produit des règles d'association et les transmet au module cellulaire CARI pour générer et gérer les règles d'association booléennes.

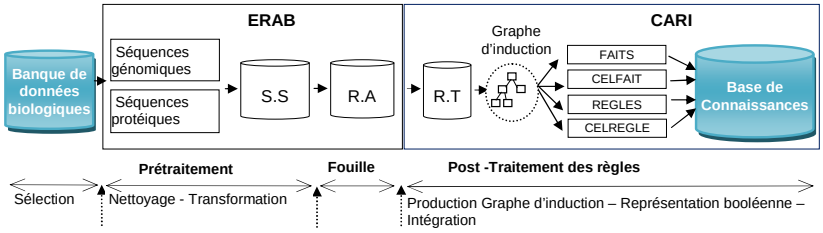


Fig. 2. Architecture générale du système BIODM (S.S : Séquences Structurées R.A : Règles d'Association R.T : Règles Transitoires).

2.1 Production des règles booléennes par la machine cellulaire CARI

CARI (Cellular Automaton for Rules Induction) est un automate cellulaire qui simule le principe de fonctionnement de base d'un moteur d'inférence classique en utilisant deux couches finies d'automates finis. La première couche, CELFAIT, pour la base des faits et la deuxième couche, CELREGLE, pour la base de règles. Chaque cellule au temps t+1 ne dépend que de l'état des ses voisines et du sien au temps t. Dans chaque couche, le contenu d'une cellule détermine si et comment elle participe à chaque étape d'inférence. A chaque étape, une cellule peut être active (1) ou passive (0), c'est-à-dire participe ou non à l'inférence. Le principe adopté est simple [1] :

- toute cellule i de la première couche CELFAIT est considérée comme fait établi si sa valeur est 1, sinon, elle est considérée comme fait à établir. Elle se présente sous trois états : état d'entrée (EF), état interne (IF) et état de sortie (SF) ;
- toute cellule j de la deuxième couche CELREGLE est considérée comme une règle candidate si sa valeur est 1, sinon, elle est considérée comme une règle qui ne doit pas participer à l'inférence. Elle se présente sous trois états : état d'entrée (ER), état interne (IR) et état de sortie (SR). Les matrices d'incidence R_E et R_S représentent la relation entrée/sortie des faits et sont utilisées en chaînage avant et en chaînage arrière en inversant leur ordre.

La dynamique de CARI pour simuler le fonctionnement d'un moteur d'inférence, utilise deux fonctions de transitions δ_{fact} et δ_{rule}, où δ_{fact} correspond à la phase d'évaluation, de sélection et de filtrage, et δ_{rule} correspond à la phase d'exécution.

- La fonction de transition δ_{fact} :

$$\delta_{fact}\,(EF, IF, SF, ER, IR, SR) = (EF, IF, EF, ER+(R_E^T.EF), IR, SR)\ ;$$

- La fonction de transition δ_{rule} :

$$\delta_{rule}\,(EF, IF, SF, ER, IR, SR) = (EF+(RS.ER), IF, SF, ER, IR, \hat{}ER)$$

où la matrice R_E^T désigne la transposée de R_E et ^ER désigne la négation du vecteur booléen ER.

2.2 Les étapes du processus adopté

Le processus de fouille de données adopté par notre système (ERAB + CARI) est composé de 4 étapes majeures :

1. sélection des données

A partir de la banque de données de NCBI [12], nous récupérons les séquences biologiques (table 1), qui vont servir à constituer la base de test expérimentale (table 2).

2. prétraitement

Un nettoyage et une transformation des données, du format original vers un formalisme adéquat, sont faits. Suivra alors une «binarisation», une opération nécessaire pour l'étape suivante.

3. fouille de données

Une recherche de règles d'association est faite par l'algorithme Apriori [7], avec calcul systématique du support et de la confiance pour ne retenir que les règles confiantes.

4. post-traitement des règles d'association

(a) transformation

Les règles d'association extraites sont transformées et représentées selon un formalisme transitoire aidant à la production d'un graphe d'induction. Ainsi la règle d'association R_i est traduite en une règle transitoire selon le principe suivant :

$$(R_i\ ,\ Antécédent_i\ ,\ Conséquent_i\ ,\ s,\ c) \rightarrow (R_i\ ,\ Prémisse_i\ (Antécédent_i\ )\ ,\ Conclusion_i\ (Conséquent_i\ )\ )\ ;$$

(b) production du graphe d'induction

Un algorithme utilisera en entrée R_i , les faits de $Premisse_i$ et de $Conclusion_i$, et en sortie il donnera un graphe d'induction où l'on aura : un sommet s_i qui désignera un nœud sur lequel on fait un test, avec des résultats possibles binaires ou à valeurs multiples ;

(c) modélisation booléenne

1. génération des règles cellulaires à partir du graphe d'induction sous la forme :

   $$R_i : Si\ \{\ Prémisse_i\ \}\ Alors\ \{\ Conclusion_i\ ,\ Sommet_i\ \}$$

   où $Prémisse_i$ est composée des items de $Antécédent_i$ et la $Conclusion_i$ est composée des items de $Conséquent_i$;

2. les règles générées (étape 4.c.1) sont représentées en couches cellulaires où :

   $$\{R_i\} \rightarrow \text{REGLES et } \{\ Prémisse_i\ ,\ Conclusion_i\ ,\ Sommet_i\ \} \rightarrow \text{FAITS}$$

3. intégration par la machine cellulaire, des règles générées, dans la base de connaissances pour les exploiter à travers différentes stratégies d'inférence.

La dynamique de la machine cellulaire est assurée par les deux fonctions de transition citées auparavant, δ_{fact} et δ_{rule} (2.1).

3 Exemple d'illustration de l'induction des règles cellulaires

Le processus général que notre système d'apprentissage applique à un échantillon est illustré par un exemple à partir de la 3e étape (2.2). Nous supposons avoir obtenu 2 règles d'association avec les gènes suivants : aceA-2, pstS-3, argC et phhB.

3e étape : fouille de données

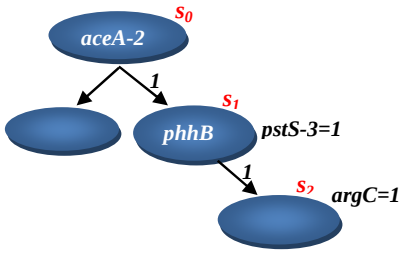
Règle	Antécédent	Conséquent	Support %	Confiance %
R1	aceA-2=1	pstS-3=1	45	77
R2	aceA-2=1, phhB=1	argC=1	45	70

4e étape : post-traitement des règles d'association

(a) transformation

Règle	Prémisse	Conclusion
R1	aceA-2=1	pstS-3=1
R2	aceA-2=1, phhB=1	argC=1

(b) production du graphe d'induction



(c) Modélisation booléenne

1. Génération des règles cellulaires à partir du graphe d'induction
   R1 : Si { s_0 } Alors { pstS-3=1, s_1 }
   R2 : Si { s_1 } Alors { argC=1, s_2 }

2. Représentation des règles en couches cellulaires
   Les règles booléennes produites R1 et R2 sont représentées par les couches CELFAIT (FAITS + CELFAIT) et CELREGLE (REGLES + CELREGLE) et les matrices d'entrée (R_E) et de sortie (R_S).

1. La couche CELFAIT

FAITS	CELFAIT		
	EF	IF	SF
s_0	1	0	0
pstS-3=1	0	1	0
s_1	0	0	0
argC=1	0	1	0
s_2	0	0	0

2. La couche CELREGLE

REGLES	CELREGLE		
	ER	IR	SR
R1	0	1	1
R2	0	1	1

3. La matrice d'incidence d'entrée R_E

R_E	R1	R2
s_0	**1**	0
pstS-3=1	0	0
s_1	0	**1**
argC=1	0	0
s_2	0	0

4. La matrice d'incidence de sortie R_S

R_S	R1	R2
s_0	0	0
pstS-3=1	**1**	0
s_1	**1**	0
argC=1	0	**1**
s_2	0	**1**

Fig. 3. Les couches cellulaires de l'automate cellulaire CARI.

3.1 La dynamique du moteur d'inférence cellulaire.

La dynamique de l'automate cellulaire CARI, pour simuler le fonctionnement d'un moteur d'inférence, utilise les deux fonctions de transitions $\boldsymbol{\delta_{fact}}$ et $\boldsymbol{\delta_{rule}}$, où $\boldsymbol{\delta_{fact}}$ correspond à la phase d'évaluation, de sélection et de filtrage, et $\boldsymbol{\delta_{rule}}$ correspond à la phase d'exécution.

1. Évaluation, sélection et filtrage (application de δ_{fact})
   $(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{fact}} (EF, IF, EF, ER + (R_E^T \cdot EF), IR, SR)$;
2. Exécution (application de δ_{rule})
   $(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{rule}} (EF + (RS.ER), IF, SF, ER, IR, \hat{}ER)$.

Nous montrons une simulation sur CELFAIT et CELREGLE de l'exemple illustré auparavant (Fig. 3), en considérant que $\mathbf{G_0}$ est la configuration initiale de l'automate cellulaire.

G_0

FAITS
s_0
pstS-3=1
s_1
argC=1
s_2

CELFAIT		
EF	IF	SF
1	0	0
0	1	0
0	0	0
0	1	0
0	0	0

REGLES
R1
R2

CELREGLE		
ER	IR	SR
0	1	1
0	1	1

1. Application de :

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{fact}} (EF, IF, EF, ER + (R_E^T \cdot EF), IR, SR)$$

FAITS
s_0
pstS-3=1
s_1
argC=1
s_2

CELFAIT		
EF	IF	SF
1	0	**1**
0	1	0
0	0	0
0	1	0
0	0	0

REGLES
R1
R2

CELREGLE		
ER	IR	SR
1	1	1
0	1	1

2. Application de :

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{rule}} (EF + (RS.ER), IF, SF, ER, IR, \hat{}ER)$$

G_1

FAITS
s_0
pstS-3=1
s_1
argC=1
s_2

CELFAIT		
EF	IF	SF
1	0	1
1	1	0
1	0	0
0	1	0
0	0	0

REGLES
R1
R2

CELREGLE		
ER	IR	SR
1	1	**0**
0	1	1

3. Application de la fonction de transition globale : $\Delta = \delta_{rule} \circ \delta_{fact}$

La machine cellulaire passe de $\mathbf{G_1}$ à $\mathbf{G_2}$ avec $\Delta(G_1) = G_2$ si $G_1 \xrightarrow{\delta_{fact}} G'_1$ et $G'_1 \xrightarrow{\delta_{rule}} G_2$

$\mathbf{G_2}$

FAITS
s_0
pstS-3=1
s_1
argC=1
s_2

CELFAIT		
EF	IF	SF
1	0	1
1	1	1
1	0	1
1	1	0
1	0	0

REGLES
R1
R2

CELREGLE		
ER	IR	SR
1	1	0
1	1	**0**

Le cycle s'arrête car aucune règle n'est applicable.

4 Expérimentation

Pour examiner l'efficacité pratique de notre système, nous avons implémenté BIODM, et nous avons mené des tests expérimentaux sur une machine Intel Celeron CPU 540 à 186 GHz 512 Mo RAM, avec une base de test réelle et synthétique (table 2).

La base de test. Nous avons considéré les 4 souches échantillon du mycobacterium tuberculosis (table 1), et nous avons pris les 15 premiers gènes de chaque souche (table 2), avec la supposition que ces gènes soient assez représentatifs et distinctifs de chaque souche prise séparément.

Table 1. Base de données expérimentale [12].

N°	Souche	Nombre de Gènes
1	Mt CDC1551	4293
2	Mt F11	3998
3	Mt H37Ra	4084
4	Mt H37Rv	4048

Table 2. Base de test : échantillon de 15 gènes de chaque souche.

N°	Gènes
1	aac accD aceA-1 aceA-2 aceB aceE ackA acnA acp-1 acp-2 acpP acpS acs adh adk
2	aceE acpP acpS adk alaS alr argC argD argJ argS aroB aroE aroK aspS atpC
3	aac aao accA1 accA2 accA3 accD1 accD2 accD3 accD4 accD5 accD6 aceAa aceAb aceE acg
4	35kd_a aac aao accA1 accA2 accA3 accD1 accD2 accD3 accD4 accD5 accD6 aceAa aceAb aceE

4.1 Les résultats expérimentaux

Temps de traitement. En utilisant la base de test, le système BIODM donne des résultats intéressants et montre l'évolution exponentielle du temps d'exécution (table 3). Nous pouvons constater également que l'exécution de l'algorithme Apriori prend une part importante en temps d'exécution du système en totalité, i.e. dans ses phases les plus importantes de l'expérimentation à savoir : la génération des règles d'association par Apriori et la génération des règles booléennes par CARI.

Confiance %	Support %	Nombre de Gènes	Items générés	Nombre de règles	Temps d'exécution Apriori	Temps d'exécution global
10	70	15	41	69	0.00 s	0.00 s
30	50	15	41	147701	0.86 s	2.23 s
50	30	15	41	147687	0.86 s	2.23 s
70	10	15	41	835284	4.92 s	10.26 s

Table 3. Evolution du temps d'exécution (génération des règles d'association par Apriori).

Espace de stockage. Nous constatons qu'à titre indicatif pour un ensemble de 147687 règles d'association, le fichier occupe un espace de stockage de 8.65 MO alors que pour les règles cellulaires correspondantes, le fichier est de 6.13 MO.
Nous constatons qu'une représentation cellulaire est plus intéressante du point de vue espace de stockage et que sur un ensemble de règles encore plus conséquent, nous aurons un gain en espace de stockage assez significatif qui se répercutera positivement sur la performance du système, et mentionnons là aussi que ce ne sont que des résultats partiels qu'il faudra consolider avec un échantillon plus important.

Table 4. Evolution de l'espace de stockage.

Nombre de règles d'association produites	Espace de Stockage (règles d'association)	Espace de stockage (représentation booléenne)
69	1.78 KO	0. 81 KO
147701	8.65 MO	6.11 MO
147687	8.65 MO	6.12 MO
835284	48.8 MO	39.14 MO

5 Conclusion et perspectives

Dans ce papier, qui doit beaucoup aux travaux engagés dans le cadre des automates cellulaires [1], [5], [8], et après avoir évoqué les inconvénients des méthodes à base de règles en fouille de données, nous avons voulu utiliser une technique assez novatrice, dans la mesure où nous voulions adopter le principe des automates cellulaires.

Notre contribution est double : nous avons non seulement souhaité utiliser Apriori avec post-traitement des règles d'association, mais aussi exploiter les performances d'un moteur d'inférence cellulaire, dont nous exploiterons la méthode d'inférence qu'il utilise, dans un contexte de fouille de données.

De ce fait, deux objectifs nous ont guidés dans la proposition d'un automate cellulaire pour l'optimisation, la génération, la représentation et l'utilisation d'une base de règles d'association booléennes. Le premier, c'est d'avoir une base de règles optimisée et des temps de traitements assez réduits grâce à une modélisation cellulaires, et le deuxième c'est d'apporter une contribution à la construction des systèmes à base de connaissances en adoptant une nouvelle technique cellulaire.

Ainsi, les avantages de notre méthode basée sur la machine cellulaire CARI peuvent être récapitulés comme suit :

- un prétraitement simple et minimal de la base de règles d'association, pour sa transformation en matrice binaire selon le principe de couches cellulaires ;
- la facilité d'implémentation des fonctions de transitions δ_{fact} et δ_{rule} qui sont de basses complexités, efficaces et robustes et concernent des valeurs extrêmes et bien adaptées aux situations avec beaucoup de règles.

Tout ce travail s'inscrit dans la perspective de recherche d'un processus de fouille de données basée sur la génération de règles d'association par inférence, intégré dans un automate cellulaire.

Références

1. Atmani, B., Beldjilali, B.: Knowledge Discovery in Database : induction graph and cellular automaton. Computing and Informatics Journal, Vol. 26 N°2 171-197 (2007)
2. Carbonnelle, B., Dailloux, M., Lebrun, L., Maugein, J., Pernot, C.: Cahier de formation en biologie médicale N°29 (2003)

3. Guillaume, S.: Traitement des données volumineuses, mesures et algorithmes Évaluation et validation d'extraction de règles d'association et règles ordinales, Thèse de doctorat, Université de Nantes (2000)
4. Han, J., Kamber, M.: Data Mining : concepts and techniques. Morgan Kaufmann Publishers (2001)
5. Abdelouhab, F., Atmani, B.: Intégration automatique des données semi-structurées dans un entrepôt cellulaire, Troisième atelier sur les systèmes décisionnels, pp. 109-120. Mohammadia – Maroc 10 et 11 octobre (2008)
6. Agrawal, R., Imielinski, T., Swami, A.: Mining associations between sets of items in large databases, Proc. of the ACM SIGMOD Conf., Washington DC, USA (1993)
7. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), pp 487-499, Santiago, Chile (1994)
8. Benamina, B., Atmani, B.: WCSS : un système cellulaire d'extraction et de gestion des connaissances, Troisième atelier sur les systèmes décisionnels, 10 et 11 octobre 2008, Mohammadia – Maroc, pp. 223-234 (2008)
9. Hajek, P., Havel, I., Chytil, M.: The GUHA method of automatic hypotheses determination,Computing 1, pp. 293-308 (1966)
10. Hipp, J., Guntzer, U., Gholamreza, N.: Algorithms for association rule mining - a general survey and comparison. SIGKDD Explorations, vol. 2, 1, pp. 58-64 (2000)
11. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of associations rules. In Proceedings of 1997 ACM SIGMOD Int'l Conference on KDD and Data Mining, KDD'97, Newport Beach, Californie (1997)
12. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov