Extraction de motifs ensemblistes dans des contextes bruités

Karima Mouhoubi, Lucas Létocart, Céline Rouveirol

Extraction de motifs ensemblistes dans des contextes bruités

Karima Mouhoubi[1], Lucas Létocart[1], Céline Rouveirol[1]

LIPN, UMR CNRS 7030, Université Paris 13
99 av. J.B. Clément, 93430 Villetaneuse, France
prenom.nom@lipn.univ-paris13.fr

Résumé : La recherche de motifs ensemblistes dans des matrices de données booléennes est une problématique importante dans un processus d'extraction de connaissances. Elle consiste à rechercher tous les rectangles de 1 dans une matrice de données à valeurs dans {0,1}. Plusieurs algorithmes ont été développés pour répondre à ce problème, mais s'adaptent difficilement à des données réelles susceptibles de contenir du bruit. Un des effets du bruit est de pulvériser un motif pertinent en un ensemble de sous-motifs recouvrants et peu pertinents, entraînant une explosion du nombre de motifs résultats. Dans le cadre de ce travail, nous proposons une nouvelle approche heuristique basée sur les algorithmes de graphes pour la recherche de motifs ensemblistes dans des contextes binaires bruités. Cette méthode est fondée sur les algorithmes de flot maximal/coupe minimale pour rechercher des sous graphes denses dans un graphe associé à la matrice des données. Pour évaluer notre approche, différents tests ont été réalisés sur des données synthétiques puis sur des données réelles issues d'applications bioinformatiques, à savoir des données d'expression de gènes.

Mots-clés : Motifs fréquents bruités, sous-graphes denses.

1. Introduction

La recherche de motifs ensemblistes dans des données booléennes consiste à rechercher tous les rectangles de 1 dans une matrice à valeurs dans $\{0, 1\}$.

Lorsque les données booléennes sont le résultat de traitements sur des données numériques issues de processus expérimentaux complexes, celles-ci peuvent alors contenir du bruit. En effet, lors des étapes de traitement et de binarisation des données, une valeur 1 peut accidentellement être mise à 0 et vice versa. La figure 1 illustre un exemple d'un contexte booléen non bruité (matrice A) où, pour une fréquence minimale de 2, deux motifs fréquents maximaux peuvent être extraits ainsi que la même matrice mais en

introduisant du bruit (matrice B).

A

	A_1	A_2	A_3	A_4	A_5	A_6
O_1	1	1				
O_2	1	1				
O_3			1	1	1	1
O_4			1	1	1	1
O_5			1	1	1	1
O_6			1	1	1	1

B

	A_1	A_2	A_3	A_4	A_5	A_6
O_1	1	1	1			
O_2	1	1				
O_3			1	1	1	1
O_4			1	1	1	1
O_5			1	1	1	1
O_6			0	1	1	1

FIGURE 1: Effet du bruit sur le nombre de motifs maximaux dans une matrice booléenne.

L'effet du bruit va donc être de fractionner des motifs importants vérifiant certaines contraintes, telle que la fréquence minimale, en un nombre exponentiel de petits fragments non pertinents.

La prise en compte du bruit pour la découverte de motifs a fait l'objet d'un nombre important de travaux de recherche dans le domaine de la fouille de données (MANNILA et SEPPANEN (2004), BESSON et al. (2005, 2006), PENSA et al. (2005), LIEU et al. (2006) et CHENG et al. (2007)). Pour résoudre ce problème, la plupart des travaux ont repris le principe de recherche par niveau de l'algorithme Apriori d'AGRAWAL et al. (1993) et sont donc limités à l'utilisation de contraintes anti-monotones pour élaguer l'espace de recherche. Une contrainte C est anti-monotone si $\forall E_1, E_2$ $(E_1 \subseteq E_2 \wedge C(E_2)) \Rightarrow C(E_1)$ où E_1 et E_2 sont des ensembles d'attributs ou d'objets. En effet, si un ensemble ne satisfait pas une contrainte, ses sur-ensembles non plus et peuvent donc être élagués (AGRAWAL et al., 1993). Dans le travail de MANNILA et SEPPANEN (2004), les auteurs recherchent toutes les régions ayant une densité et un support minimal donnés. Cependant, le choix de ces paramètres reste une tâche difficile et nécessite une connaissance préalable sur les données. De plus, la méthode reste très coûteuse puisqu'elle utilise une recherche par niveau. BESSON et al. (2006) se sont intéressés à la recherche des ensembles d'objets et d'attributs qui sont fortement associés, appelés *bi-ensembles*. Pour cela, ils utilisent des contraintes de densité sur chaque ligne et colonne. En effet, l'utilisateur doit fixer deux paramètres α et β de telle sorte que les bi-ensembles extraits ne doivent pas contenir plus de α zéros sur chaque ligne et β zéros sur chaque colonne. De plus, les

bi-ensembles doivent être maximaux et vérifier une contrainte de pertinence qui exige que le nombre d'exceptions sur chaque ligne (resp. colonne) d'un bi-ensemble soit inférieur à celui de toute autre ligne (resp. colonne) du reste de la matrice de données, avec le paramètre γ défini par l'utilisateur. Ceci donne lieu à des bi-ensembles très pertinents. Toutefois, la contrainte de densité utilisée reste très stricte et donc très coûteuse à calculer lorsque la taille des données est grande. Un autre travail de BESSON et al. (2005) présente le modéle des $\alpha\beta$-*concepts*. Ce dernier permet la recherche en deux étapes de tous les concepts formels (des rectangles maximaux de 1 modulo des permutations des lignes et des colonnes) en limitant le nombre d'exceptions par ligne et par colonne. La première étape permet d'extraire tous les concepts formels de l'ensemble des données, notés $\alpha\beta$-*concepts* initiaux. A partir de ces derniers et en se basant sur la stratégie par niveau, ils calculent à chaque niveau *l* les concepts candidats par l'union des $\alpha\beta$-*concepts* de niveau *l-1* et ne gardent que ceux qui ne possédent pas plus de α zéros sur chaque ligne et β zéros sur chaque colonne.

Citons également le modèle *Approximate Frequent Itemsets* (AFI) proposé dans LIEU et al. (2006) qui tolère aussi, comme dans les travaux de BESSON et al. (2005, 2006) , une fraction contrôlée d'exceptions sur chaque ligne ainsi que sur chaque colonne. Basée sur le principe de recherche par niveau, les auteurs ont proposé un support qui prend en compte la longueur des motifs ainsi que les taux d'erreur tolérés sur chaque ligne et colonne. Ce modèle nous permet de récupérer des sous matrices de données intéressantes mais, comme BESSON et al. (2005, 2006), reste très coûteux vus les traitements qui se font sur chaque ligne et colonne.

D'autres travaux (UNO et al. (2008) et LI et al. (2008)) se sont intéressés à l'énumération de toutes les sous matrices denses, appelées quasi-bicliques, dans des données transactionnelles. Cependant ces méthodes s'avèrent très coûteuse en temps d'execution ainsi que le nombre de résultats qui reste relativement très grand.

L'objectif de ce travail est donc d'explorer une voie alternative pour résoudre ce problème difficile de recherche de motifs bruités de manière efficace : utiliser et adapter des algorithmes d'optimisation combinatoire, notamment d'algorithmes de graphes combinés avec des méthodes de la fouille de données. Nous avons proposé dans MOUHOUBI et al. (2011) une méthode $HERD$ (Heuristique d'Extraction de Régions Denses) qui consiste, à partir des matrices booléennes, à construire des graphes bipartis augmentés et pondérés puis à rechercher les sous graphes denses les plus larges dans ces

derniers en se basant sur les algorithmes de flot maximal/coupe minimale. Le calcul de ces régions denses en 1 se fait en deux étapes : nous calculons en premier lieu l'ensemble de tous les maximaux fréquents pour un support minimal σ. Lors de la deuxième étape, pour chaque maximal fréquent, nous recherchons la région dense en 1 maximale qui le contient et qui vérifie la contrainte de densité minimale δ.

La nouvelle approche que nous présentons dans ce papier est basée sur un principe similaire à celui proposée dans MOUHOUBI et al. (2011). Nous proposons ici une nouvelle stratégie pour le calcul des motifs initiaux ainsi qu'un support adaptatif, ce qui nous permet d'obtenir de meilleurs résultats. Pour évaluer cette approche, nous l'avons implémentée et testée sur des jeux de données synthétiques ainsi que sur des données réelles issues de la Bioinformatique.

Le reste de ce papier est structuré comme suit. Nous présentons dans la section 2 un ensemble de définitions dont nous aurons besoin par la suite. Dans la section 3, nous exposons notre approche $HEMB$ (Heuristique d'Extraction de Motifs Bruités) basée sur des algorithmes de la théorie des graphes. Dans la section 4, nous exposons quelques tests et résultats obtenus. Nous concluons notre papier par quelques perspectives encourageantes de notre travail.

2. Préliminaires

Nous introduisons dans cette section quelques définitions fondamentales utilisées dans ce papier.

Définition 1 (Contexte formel et motif)
Soient A un ensemble fini d'attributs, O un ensemble fini d'observations, et R une relation binaire de O vers A. On appelle contexte formel le triplet $D = (O, A, R)$. Notons que ce dernier peut être modélisé par une matrice booléenne où les lignes et les colonnes correspondent respectivement aux observations et aux attributs. Un motif *m* est un sous-ensemble d'attributs.

Définition 2 (Motif fréquent)
Une observation $o \in O$ supporte un motif *m* si tous les attributs de *m* apparaissent dans l'observation *o*. Le support d'un motif *m*, *supp(m)*, est le cardinal de l'ensemble d'observations qui supportent m. Un motif *m* est dit fréquent relativement à un support minimal *minsup* si son support est supérieur ou égal à *minsup*. Un motif fréquent *m* est dit maximal s'il n'est sous motif d'aucun autre motif fréquent du contexte.
Pour un support minimal *minsup*=4 les motifs A_3, A_4, A_5, A_6, A_4A_5, A_5A_6,

A_4A_6 et $A_4A_5A_6$ du contexte B de la figure 1 sont fréquents. $A_4A_5A_6$ et A_3 sont des motifs fréquents maximaux.

Définition 3 (Graphe orienté, graphe biparti et densité)
Un graphe G orienté est un couple formé de deux ensembles : un ensemble de sommets V et un ensemble d'arcs E reliant des sommets de V deux à deux. Un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles V_1 et V_2 tel que chaque arc ait une extrémité dans V_1 et l'autre dans V_2. La densité d'un graphe peut être définie de différentes manières selon les besoins. En général dans la théorie des graphes et dans le cadre de ce travail, elle est définie par le rapport $\frac{|E|}{|V|^2}$. Dans le cas d'un graphe biparti, elle est définie par le rapport $\frac{|E|}{|V_1|X|V_2|}$. On dit d'un graphe qu'il est dense, relativement à un paramètre $\delta \in [0, 1]$, si sa densité est supérieure ou égale à δ.

Définition 4 (Sommet fortement associé à un ensemble de sommets)
Cette propriété permet de déterminer dans un graphe biparti $G = (V_1, V_2, E)$ l'ensemble des sommets de V_1 ayant un degré important et/ou reliés à des sommets de V_2 de degrés élevés. Un sommet $v_i \in V_1$ est fortement associé aux sommets $v_j \in V_2$ si et seulement si
$\sum_{v_j \in V_2 \wedge d(v_j) \neq 0} (\frac{d(v_i)}{max_{v_k \in V_1}(d(v_k))} + \frac{d(v_j)}{max_{v_k \in V_2}(d(v_k))}) > max_{v_k \in V_1}(d(v_k))$,
tel que $d(v)$ est le degré du sommet v.

Définition 5 (st-Coupe minimale)
Soit $G = (V, E)$ un graphe orienté possédant un sommet "source" s de degré sortant non nul et un sommet "destination" t de degré entrant non nul. À tout arc (x, y) est associé un entier $c(x, y)$ positif ou nul, sa capacité. Une st-coupe est une partition de V en $S \cup T$ où $s \in S$, $t \in T$. La capacité de la coupe notée $c(S, T)$ est la somme des capacités des arcs de S vers T. Une st-coupe est dite minimale si sa capacité est minimale.

3. L'approche proposée

Nous présentons dans cette section notre méthodologie pour l'extraction de motifs dans des contextes bruités. De manière simple, notre objectif est de rechercher les régions denses en 1 les plus grandes dans des matrices de données booléennes. Notre algorithme $HEMB$ permet d'extraire, à partir d'un motif initial m_0, la sous matrice dense maximale d qui inclue m_0 et qui respecte les contraintes suivantes :

1. chaque attribut de d a une densité supérieure à un seuil δ,
2. chaque observation de d est fortement associée à ses attributs (Définition 4).

De cette manière, nous arrivons à extraire toutes les régions maximales qui incluent m_0 et possèdant une densité supérieure à δ. Notons que chaque sous matrice extraite est maximale du fait qu'aucune de ses sur-matrices ne vérifie les contraintes 1 et 2.

3.1. Recherche des régions denses

Nous partons d'un motif initial m_0, composé d'un sous ensemble d'attributs de la matrice des données. Dans le but de rechercher la région dense et maximale qui l'inclut, nous construisons dans un premier temps le graphe correspondant (Algorithme 2) puis nous calculons une coupe minimale. Pour cela, nous avons opté pour l'algorithme de flot maximal "push-relabel" de CHERKASSKY et GOLDBERG (1997) puisque la valeur maximale du flot dans un graphe de s à t est égale à la capacité minimale d'une coupe séparant s de t (théorème flot-max/coupe-min de L.R. FORD et D.R. FULKERSON (1956)). Les capacités affectées aux arcs du graphe sont adaptées de manière à récupérer, après le calcul de la coupe minimale, un sous graphe dense qui comporte en plus des sommets attributs de m_0 un ensemble d'observations O_0 qui sont fortement associées à ces attributs. Lors de la prochaine étape, nous construisons le graphe correspondant aux observations O_0 de manière à récupérer, après le calcul de la coupe minimale, un sous ensemble d'attributs ayant des densités supérieures à δ pour ces observations O_0. Comme illustré dans l'Algorithme 1, ce processus est répété jusqu'à ce que le sous graphe dense extrait à l'étape n soit identique à celui extrait à l'étape $n-1$, dans ce cas notre sous graphe ne peut plus être augmenté et le processus est arrêté.

Comme illustré dans l'Algorithme 2, la construction d'un graphe lors de l'appel avec les observations diffère de celle avec les attributs. Cette différence réside dans l'affectation des poids (capacités) puisque les critères de sélection d'une observation sont différents de ceux d'un attribut, tenant compte du fait que le nombre d'attributs dans les matrices des données est beaucoup plus grand que le nombre d'observations (matrices d'expression de gènes). De plus, nous nous intéressons aux cas où les observations ne sont pas très denses mais dans lesquelles figurent des gènes très denses.
Rappelons qu'une coupe minimale partitionne l'ensemble des sommets d'un graphe en deux sous ensembles S, qui contient la source s, et T qui contient

la destination t. L'idée est de construire notre graphe pondéré de manière à ce que tous les sommets qui forment un sous graphe dense appartiennent à l'ensemble S après le calcul de la coupe minimale. Pour cela, nous ajoutons dans un premier temps deux sommets : une source s et une destination t. Nous relions la source à tous les sommets de l'ensemble $E_{sommets}$ pour lequel nous construisons le graphe correspondant, et pour les forcer à appartenir à S nous affectons le poids $+\infty$ à chacun des arcs qui les relie à s (Figure 2).

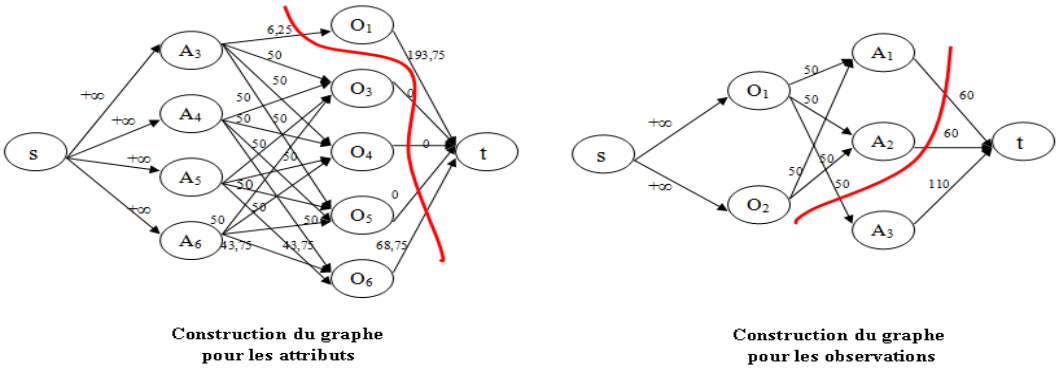

FIGURE 2: Construction des graphes et calcul des coupes minimales

Dans le cas où les sommets de l'ensemble d'appel $E_{sommets}$ sont des attributs, chacun de ces sommets $a_i \in E_{sommets}$ sera relié à tous les sommets observations o_j qui sont à 1 dans la matrice des données de cet attribut (ie, $D[o_j][a_i] = 1$). Le poids affecté à chacun de ces arcs est fonction du degré sortant $d^+(a_i)$ de son sommet source et du degré entrant $d^-(o_j)$ de son sommet destination dans le graphe obtenu. Tous les sommets o_j sont reliés à la destination t. Nous présentons dans la Figure 2 (gauche) le graphe correspondant au motif $A = A_3A_4A_5A_6$ de la matrice B (Figure 1).

Lors de l'appel avec les sommets observations, de la même manière, chacun des sommets $o_i \in E_{sommets}$ sera relié à tous les sommets attributs a_j qui sont à 1 dans la matrice des données de cet attribut (ie, $D[o_i][a_j] = 1$). Nous affectons à chacun de ces arcs le poids $\frac{100}{|E_{sommets}|}$ et nous relions chaque sommet a_j à la destination t. Un sommet a_j relié à tous les sommets de $E_{sommets}$ aura la somme de ses poids entrants égale à $|E_{sommets}|$ x $\frac{100}{|E_{sommets}|} = 100$.

De ce fait et pour pouvoir extraire les sommets qui sont reliés à plus de (δ x 100)% des sommets de $E_{sommets}$, nous affectons le poids (2x(100xδ) - $poids^-(a_j)$) à chaque arc reliant le sommet a_j et t ; sachant que $poids^-(a_j)$ designe la somme des poids des arcs entrant vers a_j. Ainsi, après le calcul de la coupe minimale, seuls les sommets ayant leurs poids supérieurs à (100xδ) vont être selectionnés. La Figure 2 (droite) illustre le graphe construit pour les observations $E = \{O_1, O_2\}$, sachant que $\delta = 0.8$.

Algorithme 1 : L'algorithme de recherche des régions denses

Entrées : D : matrice des données, $M\_initiaux$: motifs initiaux, δ : densité minimale de chaque attribut, σ : le support minimal
Sorties : SGD : l'ensemble des sous graphes denses

début
  $D\_motifs\_initiaux$ = D ; σ'=σ;
  $densite$ = DENSITE($D\_motifs\_initiaux$);
  m = MOTIF_INITIAL($D\_motifs\_initiaux$, σ');
  tant que *($m \neq \emptyset$)* **faire**
    A_{pre}={ } ; O_{pre}={ } ; i=0
    G_i=CONSTRUIRE_GRAPHE(m_i, D);
    (O_i, A_i)= ST_COUPE-MINIMALE(G_i);
    tant que *($A_{pre} \neq A_i$ ou $O_{pre} \neq O_i$)* **faire**
      SGD=SGD $\cup$ (O_i, A_i) ;
      A_{pre}=A_i ; O_{pre}=O_i ; i++;
      si *(i est impair)* **alors**
        G_i=CONSTRUIRE_GRAPHE(L_{pre}, D, δ);
      sinon
        G_i=CONSTRUIRE_GRAPHE(A_{pre}, D, 0);
      (O_i, A_i)= ST_COUPE-MINIMALE(G_i);
    METTRE_A_ZERO($D\_motifs\_initiaux$, O_i, A_i);
    $densite$' = DENSITE($D\_motifs\_initiaux$);
    si *($\frac{\sigma \times densite'}{densite} \geq 0.1$)* **alors**
      σ'=$\frac{\sigma \times densite'}{densite}$;
    m = MOTIF_INITIAL($D\_motifs\_initiaux$, σ');
fin

Notons que l'approche HERD proposée dans MOUHOUBI et al. est basée sur le même principe de recheche des régions denses. Dans HERD, nous calculons l'ensemble de tous les motifs fréquents maximaux que nous utilisons comme motifs initiaux. Cependant, le calcul de ces maximaux fréquents reste une tâche très coûteuse en temps d'execution et particulièrement pour des données denses. De plus, l'utilisation de tous les maximaux fréquents comme motifs initiaux entraîne des redondances dans les résultats. En effet, considérons le contexte de la Figure 3 dans lequel 2 motifs maximaux peuvent être extraits : $A_1A_2A_4$ et $A_1A_3A_4$. Pour une densité minimale δ=0.7, ces motifs

Algorithme 2 : CONSTRUIRE_GRAPHE

Entrées : $E_{sommets}$: ensemble des sommets, D : matrice des données, δ : paramètre de densité
Sorties : $G(V, E)$: le graphe construit

début
  $V = E_{sommets} \cup \{s, t\}$;
  pour tous les $e_i \in E_{sommets}$ **faire**
    $E = E \cup (s, e_i)$; poids$(s, e_i) = +\infty$;
  suivant *les éléments de* $E_{sommets}$ **faire**
    cas où *(les éléments de* $E_{sommets}$ *sont des observations)*
    pour tous les $o_i \in E_{sommets}$ **faire**
      pour tous les $D[o_i][a_j] = 1$ **faire**
        $V = V \cup a_j$; $E = E \cup (o_i, a_j)$;
        poids$(o_i, a_j) = \frac{100}{|E_{sommets}|}$;
    pour tous les $a_j \in V \backslash (E_{sommets} \cup \{s, t\})$ **faire**
      $E = E \cup (a_j, t)$;
      poids $(a_j, t) =$ 2x(100xδ) - $poids^-(a_j)$;
    cas où *(les éléments de* $E_{sommets}$ *sont des attributs)*
    pour tous les $a_i \in E_{sommets}$ **faire**
      pour tous les $D[o_j][a_i] = 1$ **faire**
        $V = V \cup o_j$; $E = E \cup (a_i, o_j)$;
        poids$(a_i, o_j) = \left(\frac{d^+(a_i)}{max_{a_k \in E_{sommets}}(d^+(a_k))} + \frac{d^-(o_j)}{max_{o_k \in O_j}(d^-(o_k))}\right)$ x $\frac{100}{|E_{sommets}|}$;
    pour tous les $o_j \in V \setminus (E_{sommets} \cup \{s, t\})$ **faire**
      $E = E \cup (o_j, t)$;
      poids $(o_j, t) = max_{o_k \in O_j}(d^-(o_k)) \times \frac{200}{|E_{sommets}|}$ - $poids^-(o_j)$
fin

initiaux nous mènent à l'extraction de la même région dense maximale composée des attributs A_1, A_2, A_3, A_4 et des observations O_1, O_2, O_3.

	A_0	A_1	A_2	A_3	A_4	A_5
O_0	0	1	0	0	0	1
O_1	1	1	1	1	1	0
O_2	0	1	0	1	1	0
O_3	0	1	1	0	1	0

FIGURE 3: Exemple de contexte formel entraînant des redondances.

Pour remédier à ces problèmes, nous favorisons dans ce travail l'extraction de motifs initiaux à faible recouvrement, ce qui permet d'éviter d'engendrer

des motifs initiaux qui vont mener à des régions denses maximales redondantes et donc d'optimiser le temps d'exécution. Dans le but d'obtenir des motifs initiaux les plus disjoints possible, nous extrayons les motifs initiaux et les régions denses maximales à partir de deux contextes différents initialement égaux à la matrice des données. Les régions denses maximales extraites sont mises à zéro dans la matrice d'extraction des motifs initiaux ce qui permet d'éviter l'extraction des motifs initiaux à fort recouvrement.

Nous exposons dans ce qui suit la nouvelle méthode pour le calcul des motifs initiaux ainsi qu'un support adaptatif qui nous permettent d'améliorer les résultats en temps d'execution ainsi qu'en nombre de résultats redondants.

3.2. Calcul des motifs initiaux

Nous présentons dans cette partie une heuristique pour l'extraction des motifs initiaux. Etant donnés une matrice booléenne $D\_motifs\_initiaux = (O, A)$, un support minimal σ et un motif initial m_0 initialement vide, nous extrayons dans un premier temps l'attribut a_j ayant le support le plus élevé dans $D\_motifs\_initiaux$. L'attribut extrait est ajouté à m_0 uniquement si le motif résultat vérifie la contrainte de support minimal. En suivant le même principe, lors de chaque étape, nous recherchons un attribut a_j qui forme avec les attributs de m_0 un motif de support le plus élevé possible et vérifiant le support minimal. Le processus est arrêté lorsqu'aucun des attributs de $A \setminus m_0$ ne forme un motif fréquent avec les attributs de m_0 (Algorithme 3).

Algorithme 3 : MOTIF_INITIAL

Entrées : $D\_motifs\_initiaux$: matrice de calcul des motifs initiaux, σ : support minimal
Sorties : m_0 : un motif initial

```
1 début
      /* a_j : un attribut de D_motifs_initiaux                    */
2     a = argmax_{a_j}(SUPPORT(a_j));
3     m_0 = ∅;
4     tant que (SUPPORT(a ∪ m_0) ≥ σ) faire
5         m_0 = m_0 ∪ a;
6         a = argmax_{a_j ∧ a_j ∉ m_0}(SUPPORT(m_0 ∪ a_j));
7 fin
```

3.3. Support adaptatif

Dans le but d'éviter des redondances dans les résultats, comme illustré dans la ligne 17 de l'Algorithme 1, la région dense maximale extraite lors

d'une étape est mise à 0 dans le contexte d'extraction des motifs initiaux. Avec cette méthode, le cas de résultats redondants ne se produit pas : la région dense extraite pour le premier motif initial sera mise à 0 dans la matrice d'extraction des motifs initiaux et donc le deuxième motif initial ne sera pas extrait. Nous tenons à préciser que les sous matrices denses maximales extraites ne sont mises à 0 que dans le contexte d'extraction des motifs initiaux et pas dans celui d'extraction des régions denses. Afin d'assurer l'extraction du plus grand nombre de régions denses maximales et d'éviter la perte d'informations, nous avons défini un support minimal adaptatif. Le support minimal d'extraction des motifs initiaux est mis à jour lors de chaque étape en fonction de la modification de la densité du contexte d'extraction des motifs initiaux. Comme les régions denses extraites sont mises à 0 dans le contexte d'extraction des motifs initiaux, la densité de ce dernier diminue au fur et à mesure. De ce fait, des motifs vérifiant le support minimal initial dans la matrice des données peuvent ne pas le vérifier après la mise à 0 des régions extraites. Pour éviter la perte de ces motifs, comme présenté dans la ligne 20 de l'Algorithme 1, le support minimal est mis à jour à chaque itération en fonction du rapport entre la nouvelle densité et la densité initiale et le support initial.

4. Expérimentations et résultats

Afin d'évaluer notre méthode, nous présentons dans cette section quelques tests et expériences réalisés. L'implémentation de l'approche a été réalisée avec le langage C sous Linux. Nous avons utilisé pour l'expérimentation un ordinateur équipé d'un microprocesseur intel(R) Pentium(R) 4 (3 GHz) et d'une mémoire vive de 2GB. L'évaluation a été effectuée sur deux types de jeux de données, synthétiques et des données réelles d'expression de gènes que nous avons comparé aux résultats obtenus par l'approche $HERD$ proposée dans MOUHOUBI et al. (2011) ainsi qu'à ceux obtenus par l'algorithme $Dense$ de MANNILA et SEPPANEN (2004).

Dans le but d'étudier la pertinence de l'approche, nous avons construit des jeux de données dans lesquels nous avons introduit des régions denses et nous avons comparé les résultats de l'algorithme à ce qui devait être extrait (les régions denses introduites). Nous avons repéré deux cas en consiérant la structure des régions denses : le cas des sous matrices denses disjointes (n'ayant aucune observation ou attribut en commun) et le cas de régions denses qui se recouvrent (ayant des observations et/ou attributs en commun).

Après plusieurs tests, nous avons conclu que notre algorithme $HEMB$ as-

sure l'extraction de toutes les régions denses disjointes et maximales vérifiant la contrainte de densité minimale. Dans le cas de d'un fort recouvrement des régions (grand nombre d'observations et d'attributs en commun), par rapport à δ, l'approche extrait une seule région qui réunit ces régions en une seule. Dans le cas contraire, elle extrait les régions indépendamment.

Nous avons aussi évalué l'approche sur des données réelles d'expression de gènes en faisant varier le seuil de densité δ, pour un support minimal de 0.2. Le tableau 2 montre les résultats obtenus par l'approche proposée dans ce papier ainsi que ceux obtenus par l'algorithme $HERD$ sur un jeu de données réelles, les données de SPELLMAN et al. (1998). Nous avons aussi executé l'algorithme $Dense$ sur ces données. Elles se composent de 69 puces à ADN mesurant l'expression de 407 gènes pendant le cycle cellulaire chez la levure. Leur densité est de 0.27 (27%) (ELATI et al. (2007)). Pour un support minimal de 0.2, ces données contiennent 734 motifs maximaux fréquents.

	$HEMB$				$HERD$			
δ	nombre de résultats	densité moy	taille max	temps	nombre de résultats	densité moy	taille max	temps
0.5	353	0.75	36	2s	5380	0.81	95	1m 32s
0.6	857	0.83	36	6s	5096	0.85	90	1m 30s
0.7	896	0.91	36	7s	3485	0.89	43	1m 23s
0.8	561	0.97	21	12s	1208	0.96	13	27s

TABLE 1: Résultats des expérimentations sur les données de SPELLMAN et al. (1998).

Nous mettons en évidence dans la première colonne du tableau 1 les différentes valeurs de densité minimale δ. Nous calculons pour chaque algorithme le nombre de motifs extraits, leur longueur maximale ainsi que le temps d'exécution de chaque algorithme en minutes (m) et secondes (s). Nous présentons dans la deuxième colonne le nombre de résultats extraits vérifiant la contrainte de densité minimale. La troisième colonne contient les densités moyennes des résultats obtenus. Les résultats obtenus nous montrent que notre algorithme permet d'extraire des motifs de densités importantes, de grandes tailles et surtout en un temps d'exécution raisonnable par rapport à notre précédente approche et surtout par rapport à $Dense$. Les expérimentations lancées avec l'algorithme $Dense$ pour des densités inférieures à 0.8 n'ont pu être terminées. Lors de l'appel avec $\delta = 0.7$, les calculs ont été arrêtés au niveau 3, après plusieurs heures de calculs. Pour $\delta = 0.6$ et 0.5, les calculs n'ont pas

pu être terminés à cause d'un manque d'espace mémoire. Nous avons obtenu avec $Dense$ pour $\delta = 0.8$, après plus de 8 heures de calcul, un très grand nombre de résultats (74 356 006 motifs).

Nous présentons dans le tableau 2 les résultats otenus en terme de nombre de régions denses maximales par les trois approches pour les différentes valeurs de densité ainsi que le nombre de régions denses maximales différentes.

	$HEMB$		$HERD$		$Dense$
δ	denses maximales	denses maximales différentes	denses maximales	denses maximales différentes	denses maximales
0.5	91	86	734	229	-
0.6	203	200	734	363	-
0.7	336	334	734	555	-
0.8	446	446	734	714	-
0.9	482	482	734	726	1969

TABLE 2: Régions denses maximales extraites

Notons que le nombre de régions denses maximales extraites par $HERD$ reste égal au nombre de motifs maximaux fréquents, dans ce cas 734, toutefois ces régions ne sont pas toutes différentes. Comme mentionné précédement, le bruit pulvérise un maximal fréquent en plusieurs maximaux, de ce fait, plusieurs motifs initiaux peuvent mener à une seule région dense ce qui explique les résultats redondants. Cependant, comme illustré dans le tableau, le nombre de redondances est considérablement réduit dans cette nouvelle approche par rapport aux résultats obtenus par $HERD$. De plus, nous remarquons que le nombre de régions denses maximales obtenu par l'algorithme de $Dense$ s'avère très élevé par rapport à nos résultats.
Pour confirmer la non perte d'informations des motifs extraits par notre approche, nous nous sommes intéressés à l'étude de l'apparition des gènes dans les motifs denses maximaux extraits par notre approche ainsi que par $Dense$ pour un support minimal de 20% et une densité de 90%. Nous représentons par l'histogramme de la figure 4 les fréquences d'apparition de chaque gène dans les régions denses maximales extraites par $Dense$. Nous représentons sur l'axe des x les fréquences d'appartions dans les régions maximales et sur l'axe des y le nombre de génes qui apparaissent dans x régions denses maximales. En analysant ces résultats ainsi que ceux obtenus par notre méthode, nous concluons que pratiquement tous les gènes apparaissent bien au moins une fois dans l'une des régions extraites par $Dense$ ainsi que par $HEMB$

ce qui nous confirme la non perte d'informations. Cependant, comme illustré dans la figure 4, certains gènes apparaissent plus fréquement dans les régions denses maximales obtenues avec $Dense$; en effet, certains gènes apparaissent dans plus de 20% des résultats. En résumé, la fréquence d'apparitions des gènes dans les régions maximales extraites par $Dense$ suit une loi de puissance négative. On observe un petit nombre de gènes très représentés dans les résultats et un très grand nombre de gènes peu représentés. Contrairement aux résultats de $Dense$, la distribution des gènes dans les régions extraites par notre algorithme reste homogène (tous les gènes apparaissent dans moins de 15 régions denses maximales). Pour le confirmer, nous avons étudié le recouvrement des résultats.

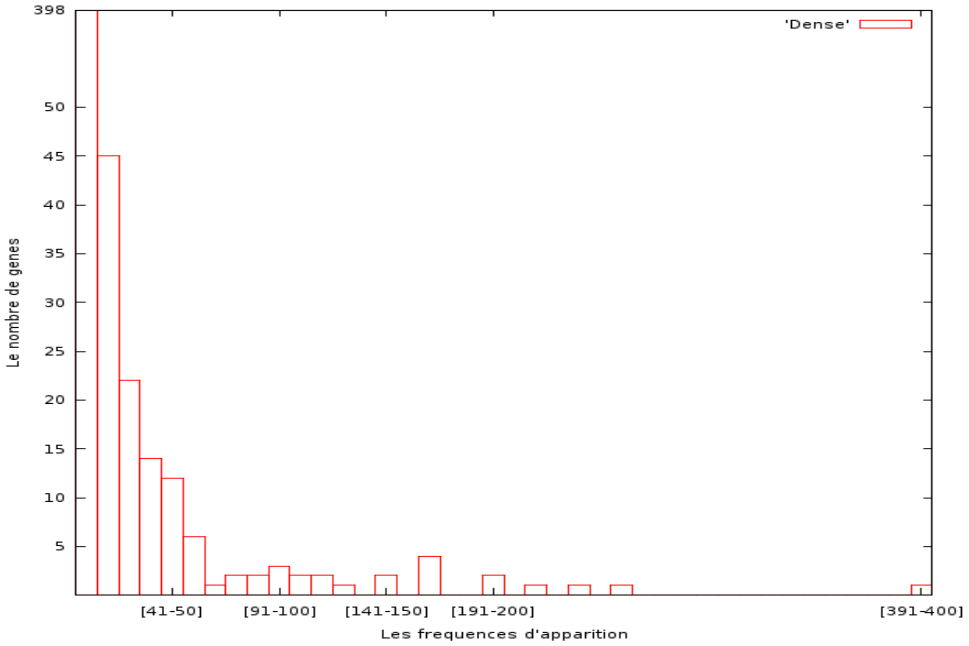

FIGURE 4: Fréquences d'apparition des gènes dans les résultats obtenus par $Dense$

Le tableau 3 résume les taux de recouvrement des régions maximales extraites. La première ligne du tableau illustre les taux de recouvrement, les deuxième et troisième lignes contiennent les nombres de résultats se recouvrant pour chaque taux de recouvrement. On remarque que le nombre de résultats se recouvrant décroit quand le taux de recouvrement augmente pour

les deux approches. Toutefois, le nombre reste très élevé dans les résultats de $Dense$ pour un taux de recouvrement suérieur à 70%. Contrairement à $Dense$, notre approche, comme prévu, permet de limiter le nombre de régions à fort recouvrement en les fusionnant, ce qui explique les différences constatées par rapport à $Dense$.

	[1-10]%	[11-20]%	[21-30]%	[31-40]%	[41-50]%	[51-60]%	[61-70]%	[71-80]%	[81-90]%
$HEMB$	101	337	143	260	517	8	13	5	5
$Dense$	0	25065	52671	93091	57642	7230	13582	5121	892

TABLE 3: Les taux de recouvrement des régions denses maximales

5. Conclusion

Nous avons présenté dans ce papier une nouvelle approche basée sur les algorithmes de graphes pour la recherche de motifs dans des contextes bruités. Cette approche a été implémentée et évaluée sur des jeux de données synthétiques et des données d'expression issues de la Bioinformatique. Les résultats obtenus sont très encourageants concernant la qualité et la taille des motifs extraits et en un temps d'exécution raisonnable. Ces résultats constituent une première satisfaction au vu du degré de difficulté du problème abordé et confirme l'intérêt d'améliorations futures de la méthode d'extraction de motifs bruités par couplage des méthodes de la fouille de données avec des techniques d'optimisation combinatoire.
En guise de perspectives, nous envisageons dans un premier temps, d'éffectuer une analyse biologique des motifs extraits pour confirmer leur pertinance et leur informativité. Au delà de cette amélioration, d'autres perspectives peuvent être envisagées. En effet, nous envisageons d'adapter notre approche pour permettre l'extraction de motifs fréquents dans des graphes dynamiques.

Références

AGRAWAL R., IMIELINSKI T. & SWAMI A. (1993). Mining Association Rules between sets of Items in Large Databases. *proc. ICDM'93*, pp 207-213.
BESSON J., ROBARDET C. & BOULICAUT J.F. (2005). Mining Formal Concepts with a Bounded Number of Exceptions from Transactional Data. *In*

KDID, 3377 :33-45.
BESSON J., BOULICAUT J.F. & ROBARDET C. (2006). Mining a New Fault-Tolerant Pattern Type as an Alternative to Formal Concept Discovery. *LNCS*, 4068 :144-157.
BORGELT C. & KRUSE R. (2002). Induction of Association Rules : Apriori Implementation. *15th Conference on Computational Statistics*, 395-400.
CHENG H., YU P.S. & HAN J. (2007). Approximate frequent itemset mining in the presence of random noise . *In Soft Computing for Knowledge Discovery and Data Mining*, pp 363-389.
CHERKASSKY B.V. & GOLDBERG A.V. (1997). On implementing the pushrelabel method for the maximum flow problem. *Algorithmica ISSN*, 19(4) : 390-410.
ELATI M., NEUVIAL P., BOLOTIN-FUKUHARA M. & al. (2007). LICORN : learning co-operative regulation networks from expression data. *In Bioinformatics*, 23 :2407-2414.
FORD L.R. & FULKERSON D.R.(1955). A simplex algorithm finding maximal networks flows and an application to the Hitchock problem. *Rand Report Rand Corporation*.
LI J., SIM K., LIU G. & WONG L. (2008). Maximal Quasi-Bicliques with Balanced Noise Tolerance : Concepts and Co-clustering Applications. *IN SDM*, pp 72-83
LIU J., PAULSEN S., SUN X. & al. (2006). Mining Approximate Frequent Itemsets In the Presence of Noise : Algorithm and Analysis. *SIAM*.
MANNILA H. & SEPPANEN J. K. (2004). Dense Itemsets. *In ACM SIGKDD*, 683-688.
MOUHOUBI K.,LETOCART L.& ROUVEIROL C. (2011). Heuristique pour l'extraction de motifs ensemblistes bruités. *EGC'11*, pp 467-472
PENSA R. G., BESSON J., ROBARDET C. & al. (2005). Constraint-based mining of fault-tolerant patterns from boolean data. *In KDID*, pp 55-71.
SPELLMAN P.T., SHERLOCK G., ZHANG M.Q. & al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9 :3273-97.
UNO T. & ARIMURA H. (2008). Ambiguous frequent itemset mining and polynomial delay enumeration. *In PAKDD*, Vol.5012, pp 357-368.