N° d'ordre : 2005-ISAL-0071 Année 2005

THÈSE

présentée devant

L'Institut National des Sciences Appliquées de Lyon

pour obtenir

Le Grade de Docteur

Spécialité

Informatique

Ecole Doctorale : Informatique et Information pour la Société

par

Jérémy Besson

Découvertes de motifs pertinents pour l'analyse du transcriptome : application à l'insulino-résistance

Soutenue publiquement le 04/11/2005 devant le jury :

Thomas Schiex, DR INRA, INRA Toulouse	Président
Gilles Bernot, Professeur, Université d'Evry	Rapporteur
Jean-Daniel Zucker, Professeur, Université Paris Nord	Rapporteur
Jean-François Boulicaut, Maître de Conférences HDR, INSA Lyon	Co-directeur
Sophie Rome, CR1 INRA, INRA/INSERM 1235	Co-directeur
Karine Clément, MCU Praticien hospitalier, Inserm « Avenir »	Examinateur

Je tiens tout d'abord a remercier très chaleureusement Jean-François Boulicaut et Sophie Rome qui ont rendu cette thèse très enrichissante à la fois sur le plan scientifique mais aussi sur le plan humain.

Ce travail doit aussi beaucoup à Céline Robardet, ma comparse de travail, pour les longues journées de travail, les séances de graffitis (schémas) sur les tableaux noires et pour les deadlines approchantes.

Merci aussi à tous les membres de l'équipes de "Data Mining et Bases de Données Inductives" de l'INSA (Sylvain Blachon, Claire Leschi, Cyrille Masson, Ieva Mitasiunaité, Ruggero G. Pensa et Christophe Rigotti) et ceux de l'équipe de "Régulation nutritionnelle de l'expression de gènes" de l'INRA pour les discussions amicales et scientifiques que nous avons pu avoir ensembles.

Je souhaite aussi remercier les membres de mon jury pour leur travail et les conseils qu'ils m'ont promulgués.

A tous, un grand Merci.

Novembre 2003

INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

***Directeur* : STORCK A.**

***Professeurs* :**

AMGHAR Y.	LIRIS
AUDISIO S.	PHYSICOCHIMIE INDUSTRIELLE
BABOT D.	CONT. NON DESTR. PAR RAYONNEMENTS IONISANTS
BABOUX J.C.	GEMPPM***
BALLAND B.	PHYSIQUE DE LA MATIERE
BAPTISTE P.	PRODUCTIQUE ET INFORMATIQUE DES SYSTEMES MANUFACTURIERS
BARBIER D.	PHYSIQUE DE LA MATIERE
BASKURT A.	LIRIS
BASTIDE J.P.	LAEPSI****
BAYADA G.	MECANIQUE DES CONTACTS
BENADDA B.	LAEPSI****
BETEMPS M.	AUTOMATIQUE INDUSTRIELLE
BIENNIER F.	PRODUCTIQUE ET INFORMATIQUE DES SYSTEMES MANUFACTURIERS
BLANCHARD J.M.	LAEPSI****
BOISSE P.	LAMCOS
BOISSON C.	VIBRATIONS-ACOUSTIQUE
BOIVIN M. (Prof. émérite)	MECANIQUE DES SOLIDES
BOTTA H.	UNITE DE RECHERCHE EN GENIE CIVIL - Développement Urbain
BOTTA-ZIMMERMANN M. (Mme)	UNITE DE RECHERCHE EN GENIE CIVIL - Développement Urbain
BOULAYE G. (Prof. émérite)	INFORMATIQUE
BOYER J.C.	MECANIQUE DES SOLIDES
BRAU J.	CENTRE DE THERMIQUE DE LYON - Thermique du bâtiment
BREMOND G.	PHYSIQUE DE LA MATIERE
BRISSAUD M.	GENIE ELECTRIQUE ET FERROELECTRICITE
BRUNET M.	MECANIQUE DES SOLIDES
BRUNIE L.	INGENIERIE DES SYSTEMES D'INFORMATION
BUFFIERE J-Y.	GEMPPM***
BUREAU J.C.	CEGELY*
CAMPAGNE J-P.	PRISMA
CAVAILLE J.Y.	GEMPPM***
CHAMPAGNE J-Y.	LMFA
CHANTE J.P.	CEGELY*- Composants de puissance et applications
CHOCAT B.	UNITE DE RECHERCHE EN GENIE CIVIL - Hydrologie urbaine
COMBESCURE A.	MECANIQUE DES CONTACTS
COURBON	GEMPPM
COUSIN M.	UNITE DE RECHERCHE EN GENIE CIVIL - Structures
DAUMAS F. (Mme)	CENTRE DE THERMIQUE DE LYON - Energétique et Thermique
DJERAN-MAIGRE I.	UNITE DE RECHERCHE EN GENIE CIVIL
DOUTHEAU A.	CHIMIE ORGANIQUE
DUBUY-MASSARD N.	ESCHIL
DUFOUR R.	MECANIQUE DES STRUCTURES
DUPUY J.C.	PHYSIQUE DE LA MATIERE
EMPTOZ H.	RECONNAISSANCE DE FORMES ET VISION
ESNOUF C.	GEMPPM***
EYRAUD L. (Prof. émérite)	GENIE ELECTRIQUE ET FERROELECTRICITE
FANTOZZI G.	GEMPPM***
FAVREL J.	PRODUCTIQUE ET INFORMATIQUE DES SYSTEMES MANUFACTURIERS
FAYARD J.M.	BIOLOGIE FONCTIONNELLE, INSECTES ET INTERACTIONS
FAYET M. (Prof. émérite)	MECANIQUE DES SOLIDES
FAZEKAS A.	GEMPPM
FERRARIS-BESSO G.	MECANIQUE DES STRUCTURES
FLAMAND L.	MECANIQUE DES CONTACTS
FLEURY E.	CITI
FLORY A.	INGENIERIE DES SYSTEMES D'INFORMATIONS
FOUGERES R.	GEMPPM***
FOUQUET F.	GEMPPM***
FRECON L. (Prof. émérite)	REGROUPEMENT DES ENSEIGNANTS CHERCHEURS ISOLES
GERARD J.F.	INGENIERIE DES MATERIAUX POLYMERES
GERMAIN P.	LAEPSI****
GIMENEZ G.	CREATIS**
GOBIN P.F. (Prof. émérite)	GEMPPM***
GONNARD P.	GENIE ELECTRIQUE ET FERROELECTRICITE
GONTRAND M.	PHYSIQUE DE LA MATIERE
GOUTTE R. (Prof. émérite)	CREATIS**
GOUJON L.	GEMPPM***
GOURDON R.	LAEPSI****.

GRANGE G. (Prof. émérite)	GENIE ELECTRIQUE ET FERROELECTRICITE
GUENIN G.	GEMPPM***
GUICHARDANT M.	BIOCHIMIE ET PHARMACOLOGIE
GUILLOT G.	PHYSIQUE DE LA MATIERE
GUINET A.	PRODUCTIQUE ET INFORMATIQUE DES SYSTEMES MANUFACTURIERS
GUYADER J.L.	VIBRATIONS-ACOUSTIQUE
GUYOMAR D.	GENIE ELECTRIQUE ET FERROELECTRICITE
HEIBIG A.	MATHEMATIQUE APPLIQUEES DE LYON
JACQUET-RICHARDET G.	MECANIQUE DES STRUCTURES
JAYET Y.	GEMPPM***
JOLION J.M.	RECONNAISSANCE DE FORMES ET VISION

Novembre 2003

JULLIEN J.F.	UNITE DE RECHERCHE EN GENIE CIVIL - Structures
JUTARD A. (Prof. émérite)	AUTOMATIQUE INDUSTRIELLE
KASTNER R.	UNITE DE RECHERCHE EN GENIE CIVIL - Géotechnique
KOULOUMDJIAN J. (Prof. émérite)	INGENIERIE DES SYSTEMES D'INFORMATION
LAGARDE M.	BIOCHIMIE ET PHARMACOLOGIE
LALANNE M. (Prof. émérite)	MECANIQUE DES STRUCTURES
LALLEMAND A.	CENTRE DE THERMIQUE DE LYON - Energétique et thermique
LALLEMAND M. (Mme)	CENTRE DE THERMIQUE DE LYON - Energétique et thermique
LAREAL P (Prof. émérite)	UNITE DE RECHERCHE EN GENIE CIVIL - Géotechnique
LAUGIER A. (Prof. émérite)	PHYSIQUE DE LA MATIERE
LAUGIER C.	BIOCHIMIE ET PHARMACOLOGIE
LAURINI R.	INFORMATIQUE EN IMAGE ET SYSTEMES D'INFORMATION
LEJEUNE P.	UNITE MICROBIOLOGIE ET GENETIQUE
LUBRECHT A.	MECANIQUE DES CONTACTS
MASSARD N.	INTERACTION COLLABORATIVE TELEFORMATION TELEACTIVITE
MAZILLE H. (Prof. émérite)	PHYSICOCHIMIE INDUSTRIELLE
MERLE P.	GEMPPM***
MERLIN J.	GEMPPM***
MIGNOTTE A. (Mle)	INGENIERIE, INFORMATIQUE INDUSTRIELLE
MILLET J.P.	PHYSICOCHIMIE INDUSTRIELLE
MIRAMOND M.	UNITE DE RECHERCHE EN GENIE CIVIL - Hydrologie urbaine
MOREL R. (Prof. émérite)	MECANIQUE DES FLUIDES ET D'ACOUSTIQUES
MOSZKOWICZ P.	LAEPSI****
NARDON P. (Prof. émérite)	BIOLOGIE FONCTIONNELLE, INSECTES ET INTERACTIONS
NAVARRO Alain (Prof. émérite)	LAEPSI****
NELIAS D.	LAMCOS
NIEL E.	AUTOMATIQUE INDUSTRIELLE
NORMAND B.	GEMPPM
NORTIER P.	DREP
ODET C.	CREATIS**
OTTERBEIN M. (Prof. émérite)	LAEPSI****
PARIZET E.	VIBRATIONS-ACOUSTIQUE
PASCAULT J.P.	INGENIERIE DES MATERIAUX POLYMERES
PAVIC G.	VIBRATIONS-ACOUSTIQUE
PECORARO S.	GEMPPM
PELLETIER J.M.	GEMPPM***
PERA J.	UNITE DE RECHERCHE EN GENIE CIVIL - Matériaux
PERRIAT P.	GEMPPM***
PERRIN J.	INTERACTION COLLABORATIVE TELEFORMATION TELEACTIVITE
PINARD P. (Prof. émérite)	PHYSIQUE DE LA MATIERE
PINON J.M.	INGENIERIE DES SYSTEMES D'INFORMATION
PONCET A.	PHYSIQUE DE LA MATIERE
POUSIN J.	MODELISATION MATHEMATIQUE ET CALCUL SCIENTIFIQUE
PREVOT P.	INTERACTION COLLABORATIVE TELEFORMATION TELEACTIVITE
PROST R.	CREATIS**
RAYNAUD M.	CENTRE DE THERMIQUE DE LYON - Transferts Interfaces et Matériaux
REDARCE H.	AUTOMATIQUE INDUSTRIELLE
RETIF J-M.	CEGELY*
REYNOUARD J.M.	UNITE DE RECHERCHE EN GENIE CIVIL - Structures
RICHARD C.	LGEF
RIGAL J.F.	MECANIQUE DES SOLIDES
RIEUTORD E. (Prof. émérite)	MECANIQUE DES FLUIDES
ROBERT-BAUDOUY J. (Mme) (Prof. émérite)	GENETIQUE MOLECULAIRE DES MICROORGANISMES
ROUBY D.	GEMPPM***
ROUX J.J.	CENTRE DE THERMIQUE DE LYON – Thermique de l'Habitat
RUBEL P.	INGENIERIE DES SYSTEMES D'INFORMATION
SACADURA J.F.	CENTRE DE THERMIQUE DE LYON - Transferts Interfaces et Matériaux
SAUTEREAU H.	INGENIERIE DES MATERIAUX POLYMERES
SCAVARDA S. (Prof. émérite)	AUTOMATIQUE INDUSTRIELLE

SOUIFI A.	PHYSIQUE DE LA MATIERE
SOUROUILLE J.L.	INGENIERIE INFORMATIQUE INDUSTRIELLE
THOMASSET D.	AUTOMATIQUE INDUSTRIELLE
THUDEROZ C.	ESCHIL – Equipe Sciences Humaines de l'Insa de Lyon
UBEDA S.	CENTRE D'INNOV. EN TELECOM ET INTEGRATION DE SERVICES
VELEX P.	MECANIQUE DES CONTACTS
VERMANDE P. (Prof émérite)	LAEPSI
VIGIER G.	GEMPPM***
VINCENT A.	GEMPPM***
VRAY D.	CREATIS**
VUILLERMOZ P.L. (Prof. émérite)	PHYSIQUE DE LA MATIERE

***Directeurs de recherche C.N.R.S.* :**

BERTHIER Y.	MECANIQUE DES CONTACTS
CONDEMINE G.	UNITE MICROBIOLOGIE ET GENETIQUE
COTTE-PATAT N. (Mme)	UNITE MICROBIOLOGIE ET GENETIQUE
ESCUDIE D. (Mme)	CENTRE DE THERMIQUE DE LYON
FRANCIOSI P.	GEMPPM***
MANDRAND M.A. (Mme)	UNITE MICROBIOLOGIE ET GENETIQUE
POUSIN G.	BIOLOGIE ET PHARMACOLOGIE
ROCHE A.	INGENIERIE DES MATERIAUX POLYMERES
SEGUELA A.	GEMPPM***
VERGNE P.	LaMcos

***Directeurs de recherche I.N.R.A.* :**

FEBVAY G.	BIOLOGIE FONCTIONNELLE, INSECTES ET INTERACTIONS
GRENIER S.	BIOLOGIE FONCTIONNELLE, INSECTES ET INTERACTIONS
RAHBE Y.	BIOLOGIE FONCTIONNELLE, INSECTES ET INTERACTIONS

***Directeurs de recherche I.N.S.E.R.M.* :**

KOBAYASHI T.	PLM
PRIGENT A.F. (Mme)	BIOLOGIE ET PHARMACOLOGIE
MAGNIN I. (Mme)	CREATIS**

* ***CEGELY***	*CENTRE DE GENIE ELECTRIQUE DE LYON*
** ***CREATIS***	*CENTRE DE RECHERCHE ET D'APPLICATIONS EN TRAITEMENT DE L'IMAGE ET DU SIGNAL*
GEMPPM	*GROUPE D'ETUDE METALLURGIE PHYSIQUE ET PHYSIQUE DES MATERIAUX*
****LAEPSI***	*LABORATOIRE D'ANALYSE ENVIRONNEMENTALE DES PROCEDES ET SYSTEMES INDUSTRIELS*

Résumé

Les nouvelles technologies à haut-débit telles que les puces à ADN et le séquençage ont ouvert de nouvelles perspectives en biologie moléculaire. En revanche, cette grande quantité d'informations pour être vraiment exploitable, nécessite le développement de nouveaux outils, en particulier de Data Mining. Ainsi, nous avons proposé d'utiliser la technologie des "motifs locaux ensemblistes" pour pouvoir extraire des "régularités" dans les données transcriptomiques. Ces régularités permettent aux biologistes d'explorer et de mieux appprehender leurs données. Nous avons développé un nouvel algorithme d'extraction de concepts formels sous contraintes. Cet algorithme permet d'exploiter des contraintes plus pertinentes que celles proposées par les algorithmes traditionnels. L'objectif est d'améliorer la faisabilité des extractions et d'augmenter la pertinence des motifs extraits dans les contextes qui nous intéressent. Nous avons utilisé cet algorithme sur des données réelles. Certains motifs extraits ainsi qu'une validation biologique nous ont permis de découvrir de nouveaux gènes cibles potentiels du facteur de transcription SREBP en réaction à l'insuline. Un autre problème soulevé par les données biologiques, comme dans beaucoup de données réelles, est la présence de bruits dans les données. Or, les méthodes que nous utilisons sont très sensibles au bruit. Nous avons apporté deux contributions à ce problème.

Mots clés

Extraction de connaissances, motifs locaux ensemblistes, concepts formels, motifs tolérants au bruit, extractions sous contraintes, transcriptome, insulino-résistance

Les notations utilisées

$\mathcal{O}$: ensemble d'objets

$\mathcal{A}$: ensemble d'attributs

$\mathbf{r}$: une relation binaire

$\mathcal{C}_{p-n}$: une contrainte nommée n avec comme paramètres p

$\sharp(X)$: le nombre d'éléments de l'ensemble X

$\mathbb{R}$: l'ensemble des réels

$\mathbb{N}$: l'ensemble des entiers

$\mathbb{B}$: les booléens

Table des matières

Table des figures

Liste des tableaux

Introduction

Ce manuscrit présente mes travaux de recherche sur "la découverte de motifs pertinents pour l'analyse du transcriptome appliquée à l'insulino-résistance" mais aussi une collaboration étroite entre une équipe d'informaticiens du LIRIS-CNRS UMR 5205 travaillant dans le domaine de l'extraction de connaissances à partir de données et une équipe de biologistes de l'INRA/INSERM 1235 travaillant sur les mécanismes moléculaires liés à l'insulino-résistance. Les liens très forts entre les notions mathématiques et algorithmiques manipulées dans l'équipe de "Data Mining et bases de données inductives" du LIRIS et celles qui paraissaient pertinentes pour l'analyse du transcriptome en vue de la compréhension des mécanismes de régulation des gènes ont joué le rôle d'un remarquable catalyseur. Il est désormais clair que de nouveaux outils de fouille de données ("Data Mining") peuvent aider les biologistes lors des processus complexes d'analyse de données à très forte valeur ajoutée. Ce travail de thèse résulte d'une collaboration étroite entre informaticiens et biologistes et a donné lieu à des publications en informatique, en biologie mais aussi en bioinformatique. Ce manuscrit montrera aussi aux lecteurs, du moins nous l'espérons, l'enthousiasme et la curiosité qui ont animé toutes les personnes ayant participé à ce projet.

Contexte

L'équipe "Data Mining et bases de données inductives" concentre une partie de ses efforts de recherche sur l'extraction de motifs locaux ensemblistes dans des relations binaires (également appelées données transactionnelles). dans des matrices booléennes . Une valeur "1" entre une ligne l et une colonne c indique que l et c sont liées par une certaine relation. Différentes relations peuvent être représentées dans une même matrice. Par exemple, la table suivante indique si un gène (en colonne) est sur-exprimé ou non dans une condition expérimentale (C_1, C_2, C_3 et C_4) et si l'une de ses protéines associées a une certaine fonction ou pas (F_1 et F_2). On parle souvent d'objets pour les lignes et d'attributs pour les colonnes. Dans une telle matrice, on peut s'intéresser à des bi-ensembles de "1" (valeur vraie), i.e. des sous-ensembles de lignes associés à des sous-ensembles de colonnes et qui ne contiennent que des valeurs "1". Dans la table, $(\{C_3C_4\}, \{G_1G_2\})$ est un tel bi-ensemble. Ce motif indique que

	G_1	G_2	G_3	G_4
C_1	0	0	1	0
C_2	1	0	0	0
C_3	1	1	0	0
C_4	1	1	0	1
F_1	0	0	1	0
F_2	0	0	1	1

les gènes G_1 et G_2 sont simultanément sur-exprimés dans les conditions C_3 et C_4. On peut aussi s'intéresser aux bi-ensembles de "1" maximaux, i.e., des bi-ensembles de "1" tels qu'aucune ligne ou aucune colonne ne puisse être ajoutée sans introduire de "0" (valeur fausse). Cette propriété de maximalité permet de capturer des motifs qui contiennent le plus d'associations possible entre les lignes et les colonnes. De tels motifs sont appelés "concepts formels" et ils sont très étudiés depuis une vingtaine d'années [98]. Ainsi, le bi-ensemble $(\{C_3C_4\}, \{G_1G_2\})$ est un concept formel. Par contre, $(\{C_3C_4\}, \{G_1\})$ n'en est pas un car C_2 ou G_2 pourraient lui être ajoutés sans introduire de valeurs "0". Dans une matrice d'expression booléenne, les biologistes recherchent les ensembles maximaux de gènes et de conditions expérimentales tels que ces gènes soient, par exemple, simultanément sur-exprimés dans ces conditions. Les concepts formels capturent exactement ce type d'information. Il faut noter que les concepts formels ont été utilisés dans de nombreuses applications de l'intelligence artificielle et que de multiples algorithmes d'extraction de concepts formels (e.g., [47, 44, 8]) ont déjà été proposés.

Une matrice booléenne de taille $n \times m$ peut contenir potentiellement $2^{min(n,m)}$ concepts formels. Par exemple, avec une matrice de taille 100×100 (finalement relativement petite), il peut y avoir approximativement 10^{30} concepts formels. En pratique, les jeux de données réelles qui vont nous intéresser peuvent contenir des centaines de milliers ou même des millions de concepts formels. Notons que, même si les tailles de ces collections semblent très grandes, elles restent très petites devant $2^{min(n,m)}$. En d'autres termes, nous ne proposons à l'utilisateur final qu'un sous-ensemble très réduit des associations possibles. Malgré cela, cet utilisateur, par exemple un biologiste, ne sera réellement intéressé que par un sous-ensemble éventuellement très petit de tous les concepts formels qui existent dans ses données. Par exemple, lors d'une analyse, il peut vouloir seulement les concepts formels qui contiennent au moins X gènes dont G_k et au moins Y conditions. Une première approche pourrait être d'extraire tous les concepts formels puis, dans un second temps, de ne fournir que ceux qui satisfont les contraintes de sélection posées par l'utilisateur. Il est cependant clair qu'une telle approche de traitement des contraintes a posteriori ne pourra pas fonctionner dans les applications réelles qui nous intéressent : la première phase ne sera pas faisable du fait de la complexité des calculs. Pour fournir aux utilisateurs les concepts formels ayant les propriétés souhaitées, il est indispensable de travailler à l'extraction

de concepts formels sous contraintes. Dans le cadre des bases de données inductives, on parle alors de requêtes inductives pour les définitions déclaratives des propriétés des motifs recherchés (utilisation d'expressions booléennes sur des contraintes primitives). Ce cadre a été particulièrement étudié au LIRIS et plus généralement dans les projets européens cInQ (IST-2000-26469-achevé) et IQ (FP6-516169-démarrage début Septembre 2005). L'un des principaux axes de recherche du domaine est alors d'évaluer de telles requêtes, autrement dit de calculer tous les motifs qui sont dans la solution d'une requête inductive.

Les difficultés viennent non seulement de la taille des espaces de recherche mais aussi de la taille des solutions (e.g., des millions de motifs). Il faut noter également, et c'est l'une des différences avec les applications classiques en intelligence artificielle, que l'accès aux données (e.g., pour vérifier qu'un motif satisfait une certaine contrainte) est très coûteux lorsque les volumes concernés sont très grands. L'exploitation efficace des contraintes pour optimiser l'évaluation de requêtes inductives a été très étudiée pour certains types de motifs (e.g., les ensembles d'attributs dans des données booléennes, les motifs séquentiels dans des bases de séquences) et certains types de contraintes (e.g., les contraintes de fréquence, certaines contraintes sur la forme des motifs) mais reste très difficile pour de nouveaux types de motifs et ou de nouveaux types de contraintes. En recherchant à exploiter efficacement des contraintes définies par l'utilisateur sur des bi-ensembles, on va pouvoir non seulement permettre des extractions qui étaient jusqu'ici infaisables (i.e., les contraintes permettent à la fois de réduire l'espace de recherche visité mais aussi la taille de la solution) tout en améliorant la pertinence a priori des motifs fournis (i.e., lorsque cela reste possible, tous les motifs délivrés satisfont les contraintes posées et seulement ceux-ci).

Extraire des connaissances dans des données ne peut pas se résumer à la mise en oeuvre d'algorithmes d'extraction de motifs. Les processus d'extraction de connaissances sont habituellement découpés en trois grandes phases : le pré-traitement, l'extraction de motifs proprement dite et le post-traitement. Ainsi, le pré-traitement comporte de nombreuses manipulations sur les données afin de préparer un contexte d'extraction. Il faut, par exemple, intégrer des données issues de multiples sources, normaliser certaines données numériques, encoder des propriétés booléennes, traiter les données manquantes, construire le ou les contextes d'extraction, etc. La phase d'extraction proprement dite peut faire appel à divers algorithmes de fouille de données. Le post-traitement consiste à valider la pertinence des motifs extraits et à les interpréter en termes de connaissances sur le domaine d'application concerné. L'aspect linéaire de ce processus est trompeur : il s'agit de processus fondamentalement itératifs et interactifs. Par exemple, à la suite de l'analyse de motifs extraits, l'utilisateur peut alors décider de ré-itérer le processus mais en révisant le contexte d'extraction pour se focaliser sur certains groupes d'attributs.

Le fil conducteur de cette thèse a été de s'intéresser aux scénarios d'extraction de

connaissances appliqués aux données transcriptomiques pour la compréhension des mécanismes de régulation de l'insulino-résistance. L'un des biais inhérent au contexte de la thèse était de développer et de valider la pertinence des méthodes basées sur des collections de motifs ensemblistes extraits à l'aide de requêtes inductives. Au départ, seul un travail très préliminaire sur l'extraction de règles dans des données d'expression SAGE avait été conduit [6]. Nous avons donc voulu (a) développer de nouvelles méthodes informatiques d'analyse de données basées sur l'extraction de motifs ensemblistes, (b) démontrer le potentiel de telles méthodes en analyse du transcriptome en général, et (c) découvrir de nouvelles connaissances sur l'insulino-résistance chez l'homme par l'exploitation des données d'expression de type Puces à ADN produites par l'UMR INRA/INSERM 1235.

Nous résumons maintenant les principaux travaux et résultats obtenus dans le cadre de cette thèse.

Pré-traitement

Un préalable à toute analyse de données concerne la collecte de toutes les données jugées utiles (avec souvent des problèmes délicats d'intégration de sources de données hétérogènes) et la mise en oeuvre de diverses techniques statistiques (normalisation, sélection, traitement des valeurs manquantes). Ceci étant, ces tâches sont souvent liées à des objectifs d'analyse très spécifiques : nous sommes loin d'une vision intégrée où toutes les données concernant l'analyse du transcriptome seraient déjà intégrées dans un véritable entrepôt de données. Ces étapes de pré-traitement nécessitent une connaissance experte des données et du domaine d'application et sont très souvent difficiles à automatiser. La section 4.2.1 de ce mémoire présente les étapes que nous avons réalisées sur les données concernant l'insulino-résistance. Une fois qu'une base de données est disponible, un autre pré-traitement consiste à construire le ou les contextes d'extraction utiles pour telle ou telle tâche d'analyse. Ainsi, le codage de propriétés booléennes est un point crucial. Par exemple, il s'agit pour nous de décider de propriétés booléennes d'expression des gènes, e.g., la sur-expression, à partir des valeurs d'expression numériques mesurées. Les choix faits à ce niveau vont avoir un impact majeur sur la pertinence des motifs extraits. Des méthodes de codage simples ont été proposées dans [6] mais elles nécessitent de fixer des seuils de discrétisation a priori. Or, aucune connaissance biologique ne permet de fixer de tels seuils de façon satisfaisante. Ainsi, dans [80], nous avons participé à la mise au point d'une technique qui permet de sélectionner parmi un ensemble de discrétisations possibles (avec leurs paramètres) celle qui conserve au mieux les structures globales présentes dans les données. Plus précisément, un clustering hiérarchique est appliqué sur les données réelles (avant discrétisation) ainsi que sur les différents jeux de données discrétisées. La discrétisation retenue est celle dont le dendrogramme (représentation sous forme d'arbre du clustering hiérarchique) se rapproche le plus du dendrogramme obtenu à

partir des données réelles.

Extraction de motifs

Dans des matrices d'expression, nous nous sommes intéressés aux groupes de gènes maximaux qui varient significativement et simultanément dans différentes conditions. De telles associations peuvent être extraites en recherchant les ensembles de gènes dits fermés (voir la section 2.1) dans des matrices d'expression booléennes qui codent cette variation d'expression. Or, en utilisant les extracteurs d'ensembles fermés (fréquents) qui étaient à notre disposition [101, 73, 74, 97, 32], ces extractions étaient infaisables dès lors que l'on travaillait sur des matrices contenant beaucoup de colonnes (les gènes) mais peu de lignes (les conditions). Il s'agit pourtant de la situation habituelle dans le cas des données d'expression issues de technologies à haut débit (Puces à ADN ou SAGE). En effet, les algorithmes d'extraction d'ensembles fermés fréquents ont une complexité exponentielle dans le nombre de colonnes, et si le nombre de lignes est petit, l'usage d'une contrainte de fréquence minimale ne permet pas d'améliorer significativement la situation. Nous avons donc travaillé sur ce problème en collaboration avec le laboratoire GREYC de l'Université de Caen. Nous avons montré (voir la section 2.2) que l'on pouvait trivialement adapter les algorithmes existants (comme par exemple [24]) pour obtenir tous les ensembles fermés contenus dans une matrice booléenne à partir de ceux contenus dans la matrice transposée, pourvu que l'une des dimensions soit petite. En effet, si l'on considère le support d'un ensemble fermé sur les colonnes, c'est-à-dire l'ensemble des lignes qui ne contiennent que des "1" pour ces colonnes, on obtient alors un ensemble fermé sur les lignes. De plus, un ensemble fermé sur l'une des deux dimensions est associé à un et un seul ensemble fermé sur l'autre dimension. Ces paires d'ensembles fermés forment en fait des concepts formels et ce sont les classiques propriétés de la connexion de Galois qui permettent d'établir ces résultats. Dans le cas de données contenant beaucoup de colonnes mais peu de lignes, la complexité ne dépend plus du nombre de colonnes (plusieurs milliers) mais du nombre de lignes. L'extraction était infaisable dans ce type de données, elle devient triviale (instantanée) [83, 11, 27]. Même si la différence entre les deux dimensions de la matrice de données n'est pas aussi importante, la transposition de la matrice peut permettre quand même d'améliorer sensiblement l'efficacité des extractions.

La proposition précédente pose un problème important : les algorithmes d'extraction d'ensembles fermés ou de concepts formels précédemment cités n'exploitent pas les mêmes contraintes sur les deux dimensions. Par exemple, ils permettent d'exploiter activement une contrainte de taille minimale sur les lignes, de présence de certaines lignes ou de taille maximale sur les colonnes. Par contre, il n'est pas possible d'exploiter une contrainte de taille minimale sur les colonnes. Or, nous avons souvent rencontré le besoin en motifs capturant des associations suffisamment fortes car impliquant un nombre d'éléments minimal sur les deux dimensions. De plus, si

l'on souhaite à la fois utiliser la transposition et en même temps extraire les concepts formels ayant au moins un certain nombre de lignes, il faut pouvoir exploiter la contrainte de taille minimale sur les colonnes. Or, lorsque l'extraction de tous les concepts formels est impossible, les seuls concepts que l'on peut extraire avec les algorithmes courants sont ceux contenant le moins de colonnes mais le plus de lignes, et de nombreuses associations potentiellement pertinentes ne peuvent donc pas être extraites.

Nous avons alors proposé un nouvel algorithme d'extraction de concepts formels sous contraintes appelé D-Miner [15, 12] (voir la section 2.3) qui utilise activement d'autres types de contraintes. D-Miner utilise une relation de spécialisation différente. Au lieu de parcourir l'espace de recherche des attributs et par la connexion de Galois de déduire son ensemble de lignes associé, D-Miner débute l'extraction avec le bi-ensemble correspondant au jeu de données total puis successivement découpe l'espace de recherche. Grâce à cet ordre d'énumération, D-Miner peut exploiter activement des contraintes de taille sur les deux dimensions mais aussi d'autres contraintes comme l'aire minimale d'un concept formes ou la présence obligatoire d'éléments dans chacun des ensembles qui constituent les concepts formels. Les expérimentations ont montré l'efficacité de D-Miner en particulier dans les jeux de données contenant beaucoup de concepts formels. La complexité de D-Miner (voir la section 2.3.4) dans le pire des cas est en $O(n^2mT)$ pour une matrice de taille $n \times m$ contenant T concepts. Il faut noter qu'elle est identique à la plupart des autres algorithmes qui sont en $O(n^2mT)$ ou $O(n^3mT)$. Par contre, la complexité en moyenne est en $O((n - \log 2(T) + 1)nmT)$. Ce qui montre bien que plus T est grand, c'est-à-dire plus la matrice contient de motifs, et plus l'algorithme est efficace pour calculer chacun des motifs (proportionnellement). Ces travaux ont été valorisés au sein du projet européen CINQ IST-2000-26469.

L'essentiel des travaux que nous avons réalisé sur les données transcriptionnelles ont été effectués au moyen de concepts formels extraits sous contraintes. Malgré l'intérêt avéré de ces extractions, nous avons cependant voulu étudier le problème important de l'existence des concepts formels dans des données booléennes bruitées. Il s'agit de considérer les situations bien réelles où certaines valeurs dans la matrice booléenne ont été indûment fixées à 0 ou à 1. Non seulement cela peut provenir du bruit dans les données d'origine mais le phénomène peut aussi être amplifié au moment des étapes d'encodage de propriétés booléennes. Malheureusement, les collections de concepts formels sont très sensibles au bruit dans les données et le nombre de concepts formels dans des données bruitées a tendance à exploser. Il est donc apparu crucial de résoudre ce problème. L'idée est de proposer de nouveaux types de motifs contenant une très forte association mais permettant quelques exceptions : des ensembles de gènes et de conditions expérimentales tels que les gènes soient presque tous sur-exprimés dans presque toutes les conditions. Par exemple, le bi-ensemble $(\{C_2C_3C_4\}, \{G_1G_2\})$ dans la table précédente capture des ensembles de lignes et de colonnes qui sont presque toutes associées. D'abord, nous avons participé dans [87] à

un travail sur un clustering hiérarchique de concepts formels. L'idée est de regrouper les concepts formels contenant presque les même ensembles de lignes et de colonnes et de proposer aux utilisateurs non pas les concepts formels extraits mais des groupes de concepts formels. Par ailleurs, nous avons étudié l'expression déclarative d'une tolérance aux exceptions en proposant deux nouveaux types de bi-ensembles sous contraintes [14, 9, 13]. Ces types de motifs sont des extensions des concepts formels vers la prise en compte maîtrisée d'exceptions : on extrait des associations moins fortes que pour les concepts formels mais pertinentes entre les ensembles de lignes et de colonnes. On obtient ainsi des méthodes d'extraction de bi-ensembles plus robustes au bruit.

Le premier type de motifs tolérant au bruit a été proposé dans [14, 9] (voir la section 3.2). Pour les calculer, nous réalisons un post-traitement sur des collections de concepts formels déjà extraites. L'idée est de fusionner les concepts formels et de ne conserver que les bi-ensembles maximaux qui ont un nombre borné de valeurs "0" par ligne et par colonne. Cette méthode permet d'améliorer la qualité des motifs extraits en fusionnant les concepts formels très proches. Pour extraire de tels motifs, les algorithmes d'extraction d'ensembles fréquents maximaux peuvent être adaptés. Pour cela, il suffit de remplacer les attributs par les concepts formels et les ensembles d'attributs par les bi-ensembles issus de la fusion des concepts formels. La contrainte qui borne le nombre de valeurs "0" est anti-monotone et peut donc être exploitée efficacement. Les expériences montrent qu'au delà de plusieurs centaines de concepts formels, ce calcul devient infaisable. En revanche, même sur des sous-collections de concepts formels, cette méthode fonctionne et permet d'améliorer la qualité de la collection extraite : on prend certains des concepts formels et l'on cherche à les fusionner de façon à obtenir des bi-ensembles plus grands et dont les nombres d'exceptions par ligne et par colonne sont bornés.

Le second type de motifs appelé DR-bi-set [13] (voir la section 3.3) offre une définition plus déclarative de ce que peut être un motif tolérant au bruit et pertinent. Intuitivement, on recherche des bi-ensembles contenant principalement des valeurs "1" et tels que les lignes et les colonnes à l'extérieur du bi-ensemble contiennent moins de valeurs "1" : le bi-ensemble a tendance à concentrer les "1". Plus précisément, les DR-bi-sets sont des bi-ensembles contenant un nombre borné de "0" par ligne et par colonne et tels que toutes les lignes et toutes les colonnes à l'extérieur du bi-ensemble contiennent plus de "0" que celles qui sont à l'intérieur. Ce nouveau type de motifs est une généralisation "naturelle" des concepts formels. Pour extraire ce type de motifs, nous avons adapté l'algorithme Dual-Miner [29] d'extraction sous contrainte d'ensembles d'attributs vers une extraction sous-contrainte de bi-ensembles. Le nouvel algorithme DR-Miner permet d'exploiter activement les contraintes monotones et anti-monotones sur les lignes, les colonnes et sur les bi-ensembles. Il augmente considérablement le nombre de contraintes que l'on va pouvoir utiliser dans les requêtes inductives et doit être vu comme un cadre générique pour l'extraction de bi-ensembles sous contraintes. Les validations expérimentales montrent que les DR-bi-

sets sont des motifs très intéressants pour capturer des associations dans des données bruitées. En revanche, l'extraction complète de tous les DR-bi-sets reste difficile en pratique. Ici encore, on peut utiliser DR-Miner pour étendre des associations pertinentes déjà extraites ou validées par l'utilisateur final, par exemple des concepts formels. L'extraction de motifs tolérants au bruit est un domaine particulièrement intéressant pour le traitement de données réelles et cette recherche se poursuit dans le cadre de l'ACI BINGO MD 46 et du contrat IQ FP6-516169.

Post-traitement

Nous avons participé au développement d'un outil de visualisation et de manipulation de concepts formels [87]. Cet outil réalise un clustering hiérarchique non pas sur les gènes ou les conditions expérimentales mais sur les concepts formels. C'est typiquement un outil de post-traitement visant à offrir aux biologistes un moyen d'explorer et de mieux appréhender les associations extraites. Cet outil exploite aussi la grande familiarité des biologistes avec les outils de clustering hiérarchique et les résultats visuels associés. La pertinence de cet outil a été validée dans le cadre d'une coopération avec le laboratoire CGMC concernant l'analyse de données SAGE humaines [18, 19]

Retour sur les scénarios d'extraction

Le développement des outils de pré-traitement, d'extraction de motifs et de post-traitement dédiés aux données d'expression de gènes dont nous avons parlé, nous ont amené à proposer des scénarios prototypiques d'extraction de connaissances dans des données d'expression [79]. Les scénarios proposés abordent les problèmes de l'encodage de propriétés, de l'enrichissement de données (utilisation d'autres sources d'information) et de l'utilisation des contraintes pour répondre à des questions précises. Cet article présente aussi une application sur des données d'expression de gènes chez la Drosophile en montrant comment le cadre des bases de données inductives est adapté à la ré-itération des processus d'extraction. En fait, ces scénarios sont des abstractions des différentes analyses de données que nous avons réalisées.

Nous avons développé un logiciel d'extraction de connaissances [10] (avec d'autres doctorants travaillant sur l'analyse de données d'expression de gènes), appelé Bio++, dédié à l'analyse de données d'expression de gènes. Ce logiciel regroupe les fonctionnalités nécessaires à un processus d'extraction de connaissances ainsi que les extracteurs et les méthodes présentés précédemment. A court terme, il sera mis à la disposition de la communauté scientifique.

	Hs.101174	Hs.10283	Hs.105656
M00075	0	0	1
M00076	1	1	1
M00271	0	1	1

TAB. 1 – Matrice booléenne avec en ligne des sites de fixation, en colonne des gènes et indiquant si un site de fixation est présent en amont d'un gène.

Application à l'insulino-résistance

L'équipe de l'UMR INSERM/INRA 1235 "Régulation nutritionnelle de l'expression de gènes" travaille en particulier sur la compréhension des mécanismes de régulation des gènes en réponse à l'insuline. Pour avancer sur cette problématique, l'équipe dispose de données de puces à ADN qui mesurent la variation d'expression de gènes dans le muscle squelettique humain avant et après injection d'insuline chez des personnes saines. Ces expériences ont été réalisées afin de comprendre la régulation transcriptionnelle de l'insuline, pour ensuite découvrir et mieux appréhender les altérations de la régulation chez les personnes insulino-résistantes. Nous avons d'abord réalisé (voir la section 4.2.1) de nombreuses étapes de pré-traitement sur ces données (normalisation, sélection de gènes, etc). Ensuite, nous avons essayé d'extraire les ensembles de gènes qui varient simultanément dans différentes conditions expérimentales. Ces associations entre gènes, donnent des pistes de travail aux biologistes et peuvent par exemple indiquer qu'ils partagent une même fonction biologique ou qu'ils interviennent dans un même processus de régulation. Malheureusement sur nos données, ni l'utilisation du clustering hiérarchique ni des concepts formels n'a permis d'obtenir de nouvelles hypothèses biologiques sur les mécanismes de régulation des gènes en réponse à l'insuline (voir la section 4.2.2). En effet, ces données ne permettent qu'une analyse globale de la réponse à l'insuline. Pour aller plus loin dans l'analyse, nous avons alors utilisé d'autres informations. Nous avons décidé d'enrichir nos données [16] en ajoutant des informations sur les sites de fixation de facteurs de transcription, éléments clés de la régulation génique (voir la section 4.3.2). Les facteurs de transcription se fixent sur des sites de fixation particuliers en amont des gènes (région promotrice) en stimulant ou en inhibant le complexe d'initiation de la transcription. En analysant les associations entre ensembles de gènes et ensembles de facteurs de transcription, il devient alors possible de mieux appréhender les mécanismes de la régulation transcriptionnelle.

Ainsi en utilisant différents logiciels et bases de données (SAM, SOURCE, TF-SEARCH), nous avons construit une nouvelle matrice booléenne associant des gènes régulés par l'insuline et leurs sites de fixation de facteurs de transcription lorsqu'ils étaient connus. La table précédente donne un exemple d'une telle matrice.

Nous avons ensuite centré notre étude sur le facteur de transcription SREBP1 (Sterol-responsive-element binding protein 1) qui est connu pour être impliqué dans la réponse transcriptionnelle de l'insuline [72]. En réalité, la régulation des gènes se produit très souvent par l'intermédiaire de plusieurs facteurs de transcription appelés alors co-facteurs. Ainsi, regarder les associations "un gène un site de fixation" n'est pas du tout satisfaisant pour appréhender la complexité des mécanismes de régulation. Il faut alors être capable de découvrir des associations plus pertinentes associant des ensembles de gènes et des ensembles de facteurs de transcription. Nous appellerons ces associations des "modules de régulation". Les concepts formels sont alors de bons motifs pour capturer les modules de régulation. Il est connu que SREBP1 a une faible affinité pour son site de reconnaissance SRE et a besoin de co-facteurs pour agir efficacement. En particulier, les facteurs de transcription SP1 (Stimulatory protein) et NF-Y (nuclear factor-Y) sont des co-facteurs de SREBP1. Nous avons alors regardé quels sont les gènes qui étaient potentiellement régulés par l'association de ces trois facteurs de transcription. En utilisant D-Miner nous avons extrait les modules de régulation contenant ces trois facteurs. Finalement, 1477 motifs ont été extraits (voir la section 4.3.3). Nous nous sommes intéressés plus particulièrement à un concept composé de 6 sites de fixation de facteurs de transcription : GATA-1 (M00075), GATA-2 (M00076), AML-1a/Runx1 (M00271) et SREBP1/NF-Y/SP1, et de 13 gènes : SPOP, SF1 (transport et processing de l'ARN) MORF4L2 (régulation de la transcription) MAPRE1, SDC1 (cytosquelette), VPS29, ARF4 (trafic vésiculaire et réseau trans-golgien), ABCA7 (transporteurs), PGRMC2 (récepteur membranaire), FEM1B (induction d'apoptosis), HK2 (glycolyse), HIG1 et CRYBA4 (fonctions inconnues). Une validation expérimentale par chromatin immunoprecipitation (ChIP [71]) a été réalisée sur 11 des 13 gènes. Elle montre que SREBP1 se fixe effectivement sur 8 des 11 gènes [68]. Nous avons donc découvert de nouveaux gènes cibles de SREBP1 qui, rappelons le, est un facteur de transcription connu comme étant impliqué dans la réponse à l'insuline. Cette expérience valide aussi toute la démarche que nous avons mise en place : en partant de données d'expression de gènes, en passant par l'enrichissement et l'extraction de concepts formels sous contraintes, le processus conduit à des découvertes de connaissances dans le domaine d'application, connaissances qui peuvent être publiées dans des journaux et des conférences spécialisés en biologie moléculaire.

Notons que les types de bi-ensembles plus élaborés qui ont été étudiés n'ont pas encore été utilisés dans des processus d'extraction de connaissances biologiques complets. En fait, la validation biologique de la pertinence d'une hypothèse fournie par un ou des motifs extraits "in silico", est un processus qui reste coûteux. Les concepts formels ont déjà montré leurs potentiels [12, 68, 18].

Nous résumons maintenant la structure du mémoire. Le premier chapitre présente d'abord le cadre des bases de données inductives, puis des méthodes de bi-clustering pour l'analyse des données d'expression de gènes et enfin notre approche basée sur l'utilisation de motifs locaux ensemblistes dans des données booléennes. Le chapitre

2 est consacré à l'extraction de concepts formels sous contraintes. Dans le chapitre 3, nous abordons le problème de la sensibilité au bruit de nos méthodes d'extraction de connaissances et présentons deux contributions visant à pallier ce problème. Le chapitre 4 expose le travail que nous avons réalisé pour essayer de mieux appréhender les mécanismes de régulation des gènes liés à la réponse à l'insuline.

Chapitre 1

Méthodes d'analyse du transcriptome

1.1 Enjeux et objectifs de l'analyse du transcriptome

L'activité cellulaire repose sur des mécanismes de production et de dégradation de protéines. Dans la vision classique de la biologie moléculaire, la synthèse des protéines est le résultat de deux étapes : une étape de traduction d'un gène en un ARN messager et d'une étape de transcription de cet ARN messager en protéine. Ainsi l'ARNm n'est qu'une forme intermédiaire entre le génome (le monde de l'information) et le protéome (le monde de la fonction) et constitue l'un des trois grands niveaux de régulation de l'activité cellulaire. L'étude du transcriptome, i.e., de l'ensemble des ARNm présents dans une cellule à un instant donné, est l'un des points d'accès à la compréhension des mécanismes cellulaires. Le niveau d'expression d'un gène est associé à la quantité d'ARNm présents dans une cellule. De nombreuses technologies permettent de mesurer ce niveau d'expression [82]. Certaines sont à haut-débit et permettent de mesurer simultanément l'activité de plusieurs milliers de gènes, comme les puces à ADN ou la méthode SAGE. D'autres comme la PCR temps réel permettent l'analyse d'un petit nombre de transcrits (ARNm). La caractérisation et la quantification du transcriptome d'un tissu donné placé dans des conditions expérimentales spécifiques, peuvent permettre d'identifier des gènes actifs, de déterminer des mécanismes de régulation d'expression des gènes et finalement de découvrir des réseaux de régulation de gènes. Ces nouvelles connaissances trouvent de nombreuses applications en médecine, en pharmacie et dans les activités agro-alimentaires. Les enjeux vont de la découverte de nouveaux gènes impliqués dans des maladies à la thérapie génique en passant par la découverte de nouveaux médicaments et le développement de méthodes de sélection ou de génotypage dans le domaine végétal ou animal. Pour plus de détails, le lecteur peut se référer au chapitre 1 de [43] pour découvrir "Le transcriptome : le

nouveau monde ?".

L'avènement des récentes technologies à haut-débit modifie profondément la façon dont les biologistes peuvent étudier le vivant. La biologie moléculaire peut maintenant être étudiée à l'échelle du transcriptome entier. En revanche, cette grande quantité d'information, pour être exploitable, nécessite le développement de nouvelles technologies. On peut citer par exemple la nécessité de définir de vrais modèles de données, d'améliorer les outils d'interrogation dans ces données, les méthodes de simulation, les outils de gestion et de découverte de connaissances.

En particulier, il est primordial de pouvoir extraire dans les données d'expression de gènes des régularités. En effet, il n'est plus envisageable d'analyser seulement ces données "à la main" à l'aide de tableurs car nous devons travailler sur des matrices contenant des centaines de milliers voir des millions de valeurs réelles. De très nombreuses méthodes existent pour analyser ces données comme celles de classification supervisée et non supervisée. Les récents développements en Data Mining ont permis d'apporter des solutions à l'extraction de régularités dans les données. On va s'intéresser plus particulièrement à l'extraction de groupes de synexpression [70]. Ce sont des ensembles maximaux de gènes co-exprimés associés à toutes les conditions expérimentales dans lesquelles ces gènes sont co-exprimés. Ces groupes de synexpression jouent un rôle très particulier. En effet, pour réussir à découvrir ou enrichir des réseaux de régulation, il est d'abord nécessaire de connaître quels gènes fonctionnent ensemble et dans quelles conditions. On peut alors faire l'hypothèse qu'ils appartiennent à une même voie ou à une même cascade de régulation. Ces groupes de gènes forment les premières briques vers les réseaux de régulation.

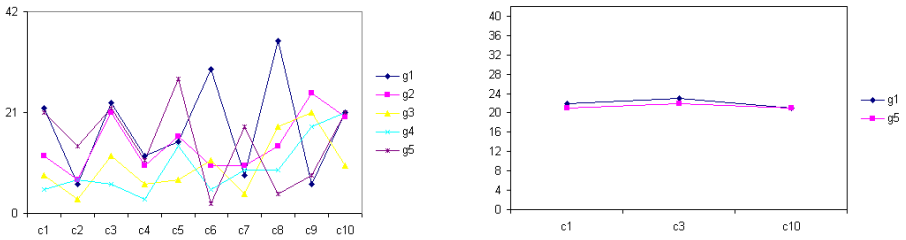
Par la suite, nous allons nous concentrer sur les méthodes de bi-partitionnement qui sont des méthodes non-supervisées. Ces méthodes permettent d'extraire des motifs appelés bi-partitions formés d'un ensemble de colonnes et d'un ensemble de lignes qui sont liés par une certaine propriété. Le bi-partitionnement est l'une des approches phares en classification conceptuelle [35].

Exemple. Le tableau 1.1 représente le niveau d'expression de 5 gènes (colonnes) dans 10 conditions expérimentales (lignes). La figure 1.1 à gauche montre les niveaux d'expression des 5 gènes pour toutes les conditions expérimentales. Il apparaît que les gènes n'ont pas de profils d'expression identiques sur les 10 conditions. En revanche, si l'on considère l'ensemble des conditions $\{c_1 c_3 c_{10}\}$ et l'ensemble de gènes $\{g_1 g_5\}$ (voir la figure 1.1 à droite), une régularité apparaît.

Notre conviction est que des informations intéressantes, par exemple surprenantes pour le biologiste car valides dans les données mais ne faisant pas partie de la connaissance du domaine, ne peuvent être découvertes que si l'on s'intéresse à des sous-ensembles de lignes et de colonnes dans la matrice d'expression, autrement dit à des motifs locaux.

	g_1	g_2	g_3	g_4	g_5
c_1	22	12	8	5	21
c_2	6	7	3	7	14
c_3	24	2	12	6	22
c_4	12	10	6	3	11
c_5	15	16	7	14	28
c_6	30	10	11	5	2
c_7	8	10	4	9	18
c_8	36	14	18	9	4
c_9	6	25	21	18	8
c_{10}	21	20	10	21	21

TAB. 1.1 – Données d'expression de gènes



Motif global Motif local

FIG. 1.1 – Exemple de motifs dans les données de la table 1.1

1.2 Approche Base de Données Inductives

La théorie des bases de données inductives (BDI) a été suggérée en 1996 par Imielinski et Mannila [53] puis développée à partir de [25, 26, 63]. C'est l'un des cadres formels les plus prometteurs pour mieux comprendre et assister de nombreux processus d'extraction de connaissances. Le projet européen cInQ (IST-2000-26469) a apporté de nombreuses contributions à cette problématique.

Un processus de découverte de connaissances dans des bases de données peut être vu comme une succession de tâches qui manipulent à la fois des données et des motifs. Par exemple, durant de tels processus, on peut avoir besoin de sélectionner certaines données, d'extraire diverses collections de motifs, de comparer des collections de motifs entre elles, de reprendre les étapes de préparation des données au regard de motifs extraits, etc. En clair, il faut pouvoir traiter des collections de données, extraire des motifs dans des données, et traiter des collections de motifs. Ainsi, la section 1.5.1 présente des exemples de questions biologiques et montre comment l'on peut essayer

d'y répondre à l'aide d'une séquence de tâches visant soit à sélectionner certaines données transcriptomiques soit à réaliser des extractions de motifs dans ces données. De plus, la section 4 présente tout le processus de découverte que nous avons réalisé sur notre problématique de l'insulino-résistance. Le cadre BDI considère qu'il s'agit là de processus d'interrogations et que les tâches peuvent être spécifiées au moyen de requêtes. Les traitements sur les données et sur des collections de motifs déjà extraites peuvent être confiés à des requêtes classiques au sens des bases de données. Les requêtes sur les motifs (extraction) demandent la conception de langages pour spécifier déclarativement les propriétés des motifs recherchés. Seules quelques propositions très préliminaires existent (voir, e.g., [65] pour un état de l'art récent) comme par exemple, des langages de requêtes pour l'extraction de règles d'association. A long terme, un véritable langage de requête BDI devrait proposer une intégration conceptuelle de ces deux mécanismes d'interrogation et même des primitives pour la gestion de requêtes spécifiant des traitements simultanés sur les données et les motifs (e.g., la sélection du sous-ensemble des données qui violent certains motifs). On pourrait alors formaliser des processus de découverte de connaissances comme des séquences de requêtes qui satisfont la propriété de clôture : chaque requête prend une instance de la base de données inductive (i.e., des données et des motifs, éventuellement définis en intention) et renvoie une nouvelle instance. Autremet dit, il s'agit de retrouver les formalisations qui ont fait le succès des bases de données relationnelles (algèbre et calculs relationnels) mais dans un cadre élargi à la découverte de connaissances. C'est clairement un objectif très ambitieux et discuter plus précisément de ces problèmes sort du cadre de ce mémoire. Nous allons nous limiter à une abstraction utile des tâches d'extraction de motifs.

Considérons une base de données $\mathbf{r}$ (e.g., une matrice booléenne), un langage de motifs $\mathcal{L}$ (e.g., le langage des bi-ensembles ou des bi-partitions), et un prédicat de sélection $\mathcal{C}$. Une tâche d'extraction proprement dite peut être formalisée comme le calcul de l'ensemble $\mathcal{TH}(\mathbf{r}, \mathcal{L}, \mathcal{C}) = \{l \in \mathcal{L} \mid \mathcal{C}(l, \mathbf{r}) \text{ est vrai}\}$ [63]. Le prédicat $\mathcal{C}$ est utilisé pour dire si oui ou non, une phrase l de $\mathcal{L}$ doit être considérée comme intéressante sur $\mathbf{r}$ (e.g., le bi-ensembles est une association maximale d'objets et d'attributs, la bi-partition est optimale au regard de la fonction objectif retenue). C'est donc la spécification déclarative des propriétés recherchées sur les motifs et l'on parle de la requête inductive pour la contrainte $\mathcal{C}$.

Deux des principales directions de recherche pour la communauté émergente des bases de données inductives sont (a) de déterminer les contraintes primitives pertinentes pour un langage de motifs $\mathcal{L}$ mais aussi les moyens utilisés pour les combiner et ainsi formuler une requête inductive (e.g., conjonctions seulement ou combinaison booléennes arbitraires), et (b) d'identifier des algorithmes efficaces pour évaluer les requêtes inductives. Ce dernier point est en effet crucial puisque, généralement, le langage $\mathcal{L}$ qui définit l'espace de recherche est très grand (voir même infini), et la base de données $\mathbf{r}$ peut être également de très grande taille (i.e., le coût de la vérification des contraintes demandant un accès aux données peut être très élevé).

Il faut bien voir que lorsque l'utilisateur formule une requête inductive, il s'agit d'une spécification déclarative et que l'idéal serait qu'il n'ait pas à se soucier des méthodes d'évaluation utilisées. Autrement dit, il faut développer des Systèmes de Gestion de Bases de Données Inductives qui seront en mesure d'élaborer des plans d'exécutions et de choisir de bonnes stratégies pour l'évaluation des requêtes inductives.

Exemple. Par exemple, un utilisateur souhaite les motifs qui satisfont une certaine propriété P_1 dans le jeu de données D_1 mais qui ne la satisfait pas dans un autre jeu de données D_2. Pour répondre à cette requête, différentes stratégies peuvent être utilisées :

- les motifs sont générés directement grâce à un algorithme qui sait extraire les motifs satisfaisant la propriété P_1 dans un jeu de données mais pas dans un autre.
- les motifs satisfaisant P_1 dans D_1 sont générés ainsi que ceux qui satisfont P_1 dans D_2. La solution est obtenue en faisant une différence ensembliste entre la première et la seconde collection.
- la base de données inductive contient déjà la collection C_1 des motifs qui satisfont une propriété P_2 dans D_1 mais pas dans D_2. Or, la propriété P_2 est moins stringente que P_1, i.e., si un motif satisfait P_2 alors il satisfait P_1. Ainsi, il suffit de ne conserver que les motifs de C_1 qui satisfont P_1 pour obtenir la solution.

Lorsque l'on dit que l'évaluation d'une requête inductive $\mathcal{C}$ doit retourner la collection $\{l \in \mathcal{L} \mid \mathcal{C}(l, \mathbf{r}) \text{ est vrai}\}$, on voit clairement que certaines requêtes ne pourront pas être évaluées. Par exemple, une requête qui demande tous les ensembles d'attributs de taille 15 parmi 30000 ne pourra pas être évaluée : il y en a trop. Un autre exemple classique est que l'on ne sait pas calculer les partitions disjointes d'attributs (chaque attribut n'appartient qu'à un seul motif) qui minimisent l'inertie intra-classe dès lors que les données contiennent plus d'une dizaine d'attributs. Ces deux exemples nous permettent d'illustrer deux problèmes importants. La première requête inductive n'est pas suffisamment sélective et il est difficile d'imaginer soit son optimisation ou même une approximation intéressante de la collection demandée. Classiquement, on peut aller vers des évaluations faisables en formulant une contrainte plus sélective, par exemple en ajoutant une contrainte qui impose de plus que les ensembles d'attributs soient fréquents au sens de [2]. Lorsque certaines des contraintes primitives utilisées sont sélectives et/ou que l'on utilise des algorithmes sachant exploiter efficacement leurs propriétés, alors on peut disposer de techniques d'évaluation justes et complètes. L'intérêt est clair, même si l'utilisateur a dû revoir sa formulation initiale, il dispose d'une caractérisation formelle des motifs extraits (i.e., ce sont tous ceux qui satisfont le prédicat $\mathcal{C}$ et seulement ceux-ci). En revanche, pour le second exemple utilisé, on ne peut pas faire une évaluation correcte et complète (i.e., seules les partitions optimales et toutes les partitions optimales sont extraites) mais des techniques d'optimisation locale permettent souvent de calculer de bonnes solutions. Ainsi, les

méthodes de type K-MEANS ou de clustering hiérarchique génèrent des collections de partitions disjointes qui peuvent être "proches" des partitions optimales. Notons cependant qu'aucune information ne permet de caractériser précisément la qualité des collections extraites. Par la suite, nous parlerons d'extractions heuristiques quand les motifs extraits ne satisfont pas exactement les prédicats de sélection, et d'extractions complètes dans le cas contraire.

Dans notre travail, nous nous sommes concentrés sur l'utilisation de motifs construits sur des bi-ensembles. Un bi-ensemble est un couple de lignes et de colonnes dans une matrice. Les bi-ensembles permettent de capturer des associations entre des éléments situés sur les deux dimensions d'un tableau de données. Une collection de bi-ensembles issue d'un jeu de données est appelée une bi-partition. La section 1.3 présente différents types de bi-ensembles qui permettent de capturer des régularités dans les données d'expression de gènes.

1.3 Introduction au bi-partitionnement

L'objectif des méthodes de bi-partitionnement est d'extraire des couples d'ensembles de lignes et d'ensembles de colonnes d'un tableau de données qui sont pertinents pour un objectif d'analyse donné. Dans la suite de ce chapitre, nous serons en présence de données matricielles (i.e., $\mathbf{r}$ est soit une matrice de nombres soit une matrice booléenne). La question de la pertinence est clairement dépendante de l'objectif d'analyse à un instant donné. Nous allons voir dans l'exemple suivant comment assister la découverte de groupes de synexpression.

Exemple. Nous avons vu dans la figure 1.1 un premier exemple de modèle pour les groupes de synexpression. Ce sont des bi-ensembles que l'on pourrait appeler bi-ensembles presque "constants". En effet, ils ne contiennent que des valeurs presque identiques (autour de 21 dans l'exemple). L'hypothèse biologique qui est faite pour ces motifs est que chaque gène répond de la même façon dans chaque condition expérimentale pour le même processus biologique. Or, cette hypothèse n'est pas complètement satisfaisante. Par exemple, on peut s'intéresser à des bi-ensembles ayant des valeurs identiques à une constante près, constante liée au gène et à la condition expérimentale. Ces constantes permettent d'exprimer le fait qu'un gène peut répondre différemment dans deux conditions et qu'une condition peut induire des réponses différentes au sein d'un même mécanisme biologique. Les figures 1.2 (a) et (b) sont des exemples de ces modèles dit additifs. Ce différentiel d'expression peut aussi s'exprimer à l'aide de facteurs multiplicatifs (voir la figure 1.2 (c)). La figure 1.2 (d) montre un quatrième type de motifs. Dans cet exemple, l'ensemble des gènes $\{g_2, g_3, g_4\}$ ont des profils assez similaires dans les conditions $\{c_1, c_6, c_9\}$ mais surtout leurs profils sont très différents de ceux des gènes g_1 et g_5. Cette contrainte particulière permet de définir les motifs à la fois par rapport aux données dans le

bi-ensemble mais aussi par rapport aux données extérieures. Cette contrainte fait référence à une forme de maximalité des motifs, il satisfait le modèle et aucun autre élément ne peut être ajouté au motif sans violer le modèle.

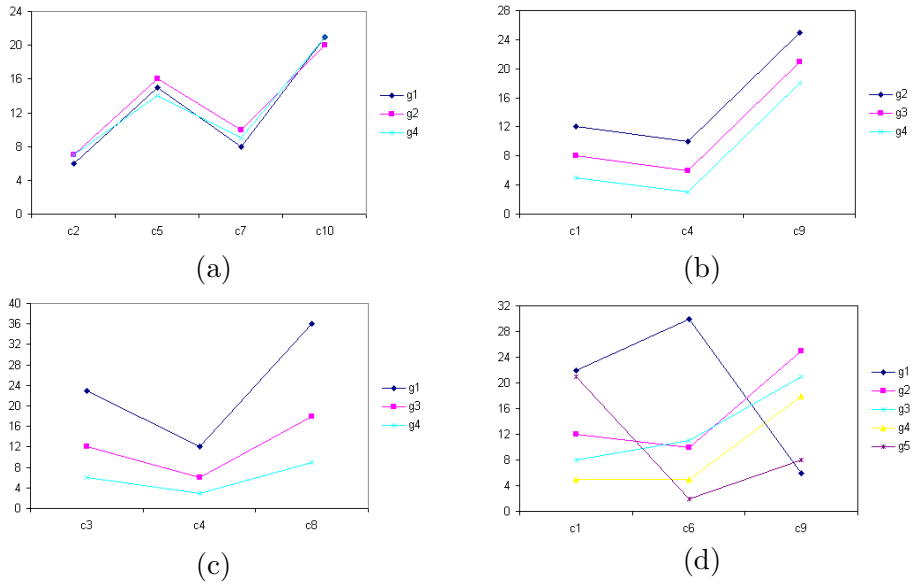


FIG. 1.2 – Exemples de modèles de bi-partitionnement

Le second problème majeur posé par les méthodes de bi-partitionnement est lié à la combinatoire de la recherche. Effectivement, si l'on travaille sur un jeu de données contenant n conditions expérimentales et m gènes, il y a 2^{n+m} bi-ensembles possibles. Par exemple, en prenant 1000 gènes et 24 conditions ($n = 24$ et $m = 1000$), il y a 2^{1024} bi-ensembles possibles, c'est-à-dire plus de 10^{308} bi-ensembles possibles. Il n'est pas envisageable d'énumérer l'ensemble de ces candidats. Les algorithmes de bi-partitionnement doivent donc être capables d'extraire les motifs sans parcourir tout l'espace de recherche. En revanche, nous ne sommes pas dans une problématique de type "data stream" car les données que l'on manipule sont pérennes au sens où les mécanismes que l'on souhaite étudier existent et existeront encore pendant longtemps. Il faut néanmoins que l'extraction des motifs soit faisable ou du moins soit assez rapide pour préserver la dynamique des processus d'extraction. Certains algorithmes utilisent des méthodes d'optimisation locale, c'est-à-dire qui cherchent de bonnes solutions mais sans être sûr d'atteindre une solution optimale. Une telle méthode est utilisée lorsque les motifs sont définis à partir d'une contrainte dont aucune solution exacte n'est connue ou parce qu'elle est trop coûteuse à calculer. Dans ce cas, l'algorithme va chercher à s'approcher le plus près possible d'une (de) solution(s).

Il y a quatre grandes classes de méthodes heuristiques pour le bi-partitionnement :

les méthodes agglomératives, divisives, par permutation ou par approximation de paramètres. La première méthode commence avec une partition discrète des lignes et/ou des colonnes puis les éléments de cette partition sont rassemblés successivement dans différents groupes. Ces différents groupes forment finalement le bi-partitionnement. Les méthodes divisives débutent avec un motif formé de toutes les lignes et de toutes les colonnes puis le découpent successivement en bi-ensembles plus petits. Les méthodes par permutation déplacent des éléments entre groupes afin d'améliorer le résultat obtenu. Ces trois premières méthodes essayent d'améliorer la qualité globale de la collection en effectuant l'opération qui améliore le plus cette qualité. Cette amélioration est bien souvent locale, de sorte que la collection finale n'est souvent qu'un optimum local. La dernière méthode est assez différente, elle cherche à calculer des paramètres (moyenne, écart type, ...). Elle s'applique souvent aux modèles probabilistes ou statistiques [64].

Certains algorithmes calculent un seul motif, K motifs (avec K un entier fixé par l'utilisateur) ou une collection de taille a priori indéfini. Le fait de calculer un seul motif se réfère souvent à l'enrichissement d'un motif déjà connu. C'est un point particulièrement intéressant. Effectivement, les biologistes possèdent souvent des informations sur un phénomène précis. Cette information peut alors être enrichie. Par exemple certains gènes peuvent être connus comme étant régulés dans des conditions expérimentales particulières ; le système peut alors fournir d'autres gènes qui varient de la même façon dans ces conditions ou d'autres conditions pour lesquelles ces gènes varient simultanément. Le fait de calculer K motifs ou les K meilleurs motifs peut être vu comme une contrainte sur la collection à extraire pour les méthodes complètes alors que pour les méthodes heuristiques c'est plutôt un paramètre nécessaire pour pouvoir calculer les solutions.

La définition des régularités à extraire et la combinatoire de l'extraction sont deux points importants des algorithmes de bi-partitionnement. Mais il faut aussi s'intéresser à la "forme" des collections extraites. Le terme "forme" fait référence à la façon dont les motifs sont disposés entre eux. Cette forme peut être assimilée à une contrainte sur la collection des motifs. Quatre contraintes principales peuvent définir la forme de la collection : l'exclusivité, le recouvrement, la structuration en arbre et le partitionnement complet des données. Une collection est exclusive si tous les éléments (gènes ou conditions) appartiennent à au plus un motif. Une collection est avec recouvrement si au moins deux motifs ont des gènes ou des conditions en commun. Une collection a une structuration en arbre si pour tout couple de bi-ensembles (X, Y) et (X', Y') alors $X \cap X' \in \{\emptyset, X, X'\}$ et $Y \cap Y' \in \{\emptyset, Y, Y'\}$. Finalement, nous parlons de partitionnement complet des lignes et/ou des colonnes ou de partitionnement complet des cases de la matrice si elles appartiennent toutes à au moins un motif de la collection extraite. La figure 1.3 présente cinq exemples de collections de bi-ensembles (bi-partitions). Pour chaque exemple, le rectangle principal représente le jeu de données et les rectangles en gris des bi-ensembles.

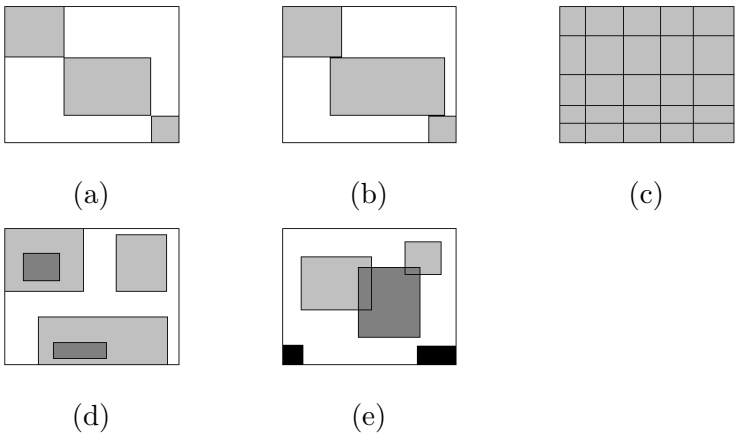


FIG. 1.3 – Exemples de collections de bi-ensembles

Le premier exemple (a) représente une collection de motifs exclusifs en ligne et en colonne, donc sans recouvrement, qui partitionne l'ensemble des lignes et des colonnes mais pas l'ensemble des cases de la matrice. La seconde collection (b) est quant à elle exclusive seulement en ligne. La troisième collection (c) est non exclusive, sans recouvrement mais l'ensemble des cases de la matrice est partitionné, le terme "checkerboard" est souvent employé pour ce type de collections. La collection suivante (d) illustre la structuration en arbre. La dernière est une collection quelconque. Il faut noter que la plupart des méthodes considèrent qu'il n'y a pas d'ordre sur les lignes et les colonnes, c'est-à-dire que le résultat de l'extraction ne dépend pas de l'ordre initial des éléments.

Avant de décrire différentes méthodes de bi-partitionnement, on va s'intéresser aux articles de Ihmels et al. [52] et de Getz et al. [45].

Ihmels et al. [52] proposent un algorithme très simple pour construire en deux étapes un bi-ensemble candidat à être un groupe de synexpression à partir d'un (petit) ensemble de gènes G fourni. Ce n'est pas à proprement parler un algorithme de bi-partitionnement mais il permet une bonne introduction à ces algorithmes et aux motifs à extraire. A partir de G et de la matrice d'expression centrée et réduite par rapport aux gènes, l'algorithme calcule d'abord un score pour chaque condition. Ce score est la moyenne des variations d'expression des gènes de G pour la condition considérée. Les conditions qui ont un score élevé (en absolu) sont sélectionnées. Ensuite, il calcule les gènes qui ont une variation d'expression significative pour les conditions en question. Pour cela, il calcule pour chaque gène la moyenne des variations d'expression pour les conditions sélectionnées, pondérée par le score. Si la moyenne est élevée, le gène est retenu. Finalement, cet algorithme essaye d'extraire un groupe de synexpression contenant un ensemble donné de gènes. Le développement

de cette approche se poursuit dans [7]. L'intérêt de cette méthode est qu'elle est très simple et que son temps de calcul est linéaire par rapport au nombre de gènes et par rapport au nombre de conditions expérimentales.

Getz et al. [45] proposent d'utiliser les méthodes de clustering sur une dimension, qui produisent des clusters de gènes ou de conditions expérimentales, pour extraire les bi-partitions. Cette méthode permet de faire un lien entre les méthodes classiques de clustering et celles de bi-partitionnement. Le principe est de calculer itérativement à l'aide d'une méthode de clustering quelconque les clusters de gènes et de conditions expérimentales contenus dans les sous-matrices définies à partir des groupes de gènes et de conditions expérimentales déjà identifiés aux étapes précédentes. Le processus débute avec la matrice complète : l'ensemble des lignes et des colonnes. Un clustering est réalisé sur les lignes et sur les colonnes. A l'étape suivante, toutes les sous-matrices (X, Y) telles que X est un cluster de lignes et Y un cluster de colonnes sont calculées. Afin d'améliorer l'efficacité de l'extraction, seules les sous-matrices qui satisfont un certain critère comme la stabilité ou une taille minimale sont considérées. Ensuite, le processus est réitéré : des clusters de lignes et de colonnes sont extraits à partir de ces sous-matrices etc. Cette méthode permet d'extraire des clusters de gènes et de conditions expérimentales et pas directement des bi-partitions. Par contre, ces motifs ont étés calculés sur des sous-ensembles de matrices et offrent donc bien des modèles locaux même si des méthodes de clustering sur une dimension (modèles globaux) ont été utilisées. Deux approches permettent néanmoins de mettre en relation les clusters de gènes et de conditions produites. Tout d'abord, si une partition de conditions expérimentales semble particulièrement intéressante d'un point de vue biologique, alors l'ensemble des gènes du tableau de données ayant conduit à cette partition peut lui être associé pour former un bi-ensemble. Une seconde approche consiste à rechercher les clusters de gènes qui n'apparaissent qu'une seule fois et l'ensemble des conditions expérimentales du tableau sur lequel ce cluster a été produit. Cet ensemble de conditions apparaît alors comme une signature de l'ensemble des gènes. La construction des bi-ensembles à proprement parler doit être faite en post-traitement. En effet, dans le premier cas, on associe à une partition de conditions un seul cluster de gènes. Dans le second, on associe bien à un ensemble de gènes un ensemble de conditions.

Nous allons nous concentrer dans la section suivante sur les algorithmes de bi-partitionnement permettant la découverte de groupes de synexpression potentiels. L'hypothèse forte qui est faite est que les données contiennent effectivement des groupes de synexpression c'est-à-dire que les variations d'expression des gènes peuvent être expliquées par les conditions expérimentales. C'est l'hypothèse du monde clos qui est appliqué au niveau des données.

Pour mettre en avant le problème de la définition des régularités à extraire et la combinatoire de l'extraction, nous présentons d'abord les algorithmes heuristiques (voir section 1.4.1) puis les algorithmes complets (voir section 1.4.2). Dans chaque

section, les méthodes présentées sont ordonnées en fonction du modèle utilisé pour définir les bi-partitions.

1.4 Algorithmes de bi-partitionnement

1.4.1 Méthodes heuristiques

Bi-ensemble presque "constant"

Une première façon de définir les groupes de synexpression consiste à dire que ce sont des bi-ensembles presque "constants", c'est-à-dire tels que les valeurs dans chaque motif sont presque identiques. Cette définition informelle peut en fait se décliner sous la forme de différentes définitions visant à définir formellement le terme "presque". Nous allons ainsi voir que les trois méthodes présentées dans cette section utilisent des définitions différentes du terme "presque". Ces trois méthodes s'expriment toutes en terme de minimisation de l'inertie intra-classe de telle sorte que chaque motif et le modèle qui le définit soient les plus proches possibles, au sens d'une certaine distance. Une métrique qui définit la qualité globale d'une collection doit aussi être définie. Elle est d'ailleurs souvent définie à partir des distances entre le modèle et les bi-ensembles.

Pour la suite, nous désignerons par Moy_{XY} la moyenne des valeurs des lignes de X sur les colonnes de Y et par M_{xy} la valeur pour la ligne x et la colonne y.

Hartigan a été le premier à proposer dès 1972 [51] une méthode appelée "Block Clustering" cherchant à extraire des bi-ensembles "constants". Il définit ainsi le modèle de base : les bi-ensembles doivent contenir des valeurs proches de la moyenne des valeurs contenues dans le motif. Le bi-ensemble "parfait" ne contient que des valeurs identiques. Ensuite, il utilise pour mesurer la distance entre le modèle et les bi-ensembles la variance des valeurs du bi-ensemble. En effet, plus la variance est faible et plus le motif est constant. La variance est la somme des différences au carré entre les valeurs et la moyenne du bi-ensemble (X,Y) :

$Variance(X,Y) = \sum_{x \in X, y \in Y} (M_{xy} - \text{Moy}_{X,Y})^2$

Ensuite, la mesure de qualité d'une collection va être simplement la somme des variances des différents motifs.

Cette mesure pose néanmoins un problème récurrent dans ce type de méthodes : la collection composée de tous les motifs de taille $1*1$ (avec une seule ligne et une seule colonne) a une mesure optimale. Pour pallier ce problème, la solution habituellement utilisée est d'ajouter une autre contrainte visant à fixer a priori le nombre de motifs que doit contenir la collection extraite. Hartigan propose donc d'extraire les K bi-

ensembles les plus "constants". La matrice initiale est découpée successivement en plusieurs sous-matrices et s'arrête lorsque la collection formée des K bi-ensembles a une distance globale inférieure à un seuil fixé par l'utilisateur. Une autre méthode vise à normaliser la mesure de qualité par la taille de la collection extraite.

Les approches développées autour du K-MEANS et du clustering hiérarchique ascendant proposent une autre façon de définir les bi-ensembles presque "constants". Elles sont telles que chaque vecteur ligne (x,Y) (resp. chaque vecteur colonne (X,y)) d'un bi-ensemble presque "constant" (X,Y) doit être le plus proche possible du barycentre des lignes (resp. des colonnes) du motif. Un barycentre est associé aux lignes et un autre barycentre est associé aux colonnes. La distance entre le modèle "parfait" (toutes les valeurs sont identiques) et un motif est obtenu à partir des distances entre les vecteurs colonne et leur barycentre et les vecteurs ligne et leur barycentre. La différence majeure entre le K-MEANS et le clustering hiérarchique ascendant réside dans l'heuristique employée. En effet, le K-MEANS adopte une méthode de permutation alors que le clustering hiérarchique ascendant adopte une méthode d'agglomération.

Busygin et al. [30] proposent d'utiliser des cartes auto-organisatrices de Kohonen (SOM) qui peuvent être considérées comme une généralisation de la méthode K-MEANS. Cette méthode consiste à partitionner l'ensemble des conditions expérimentales et l'ensemble des gènes à l'aide des cartes auto-organisatrices de Kohonen (SOM) [57] et à forcer le lien entre les deux partitions par l'intermédiaire d'une bijection associant à chaque nœud (le vecteur représentant chaque classe) d'un des deux espaces (conditions ou gènes) un nœud de l'autre espace appelé conjugué. Le procédé itératif consiste à construire l'une des deux partitions à l'aide de la méthode SOM, e.g., la partition des conditions expérimentales. Ensuite, les coordonnées des nœuds de la partition de l'autre espace, e.g., des gènes, sont calculées par le produit matriciel de la matrice d'expression gènes$\times$ conditions, dont chaque ligne a été normalisée pour être un vecteur unité, et de son conjugué qui vient d'être estimé. Une partition des gènes est alors construite à partir des coordonnées des nœuds à nouveau calculées avec la méthode SOM. On réestime ensuite les coordonnées des vecteurs associés aux nœuds de la partition des conditions par le produit matriciel entre la matrice conditions$\times$ gènes dont chaque ligne a été normalisée. Le procédé est réitéré jusqu'à ce que les partitions se stabilisent. Cette méthode fournit alors une collection de bi-ensembles formant une partition des gènes et une partition des conditions expérimentales. Les classes des gènes discriminent les classes de conditions expérimentales et réciproquement. Cette méthode a comme avantage de converger relativement rapidement. Les collections de motifs extraits sont exclusives et partitionnent l'ensemble des gènes et des conditions (voir exemple en haut à gauche de la figure 1.3). La figure 1.4 illustre comment ces motifs sont calculés. Les ronds représentent des conditions, les rectangles des gènes et les croix les représentants des classes. Dans cet exemple, on cherche deux bi-ensembles. D'abord, les deux représentants des conditions sont choisis aléatoirement sur $\mathbb{R}^2$ et les conditions leurs

sont associées, noir pour le premier représentant et gris pour le second (voir deuxième figure). Ensuite les représentants pour les gènes sont calculés (voir troisième figure). Ils sont recalculés (voir quatrième figure). Ils sont de nouveau projetés sur l'espace des conditions. Cette configuration est un point fixe, la méthode s'arrête et deux bi-ensembles sont ainsi obtenus.

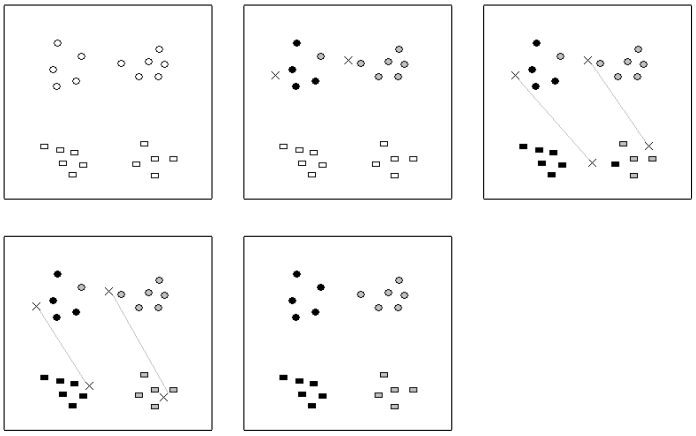


FIG. 1.4 – Bi-clustering avec SOM

Eisen et al. [41] proposent une méthode basée sur le clustering hiérarchique ascendant. Cette méthode est très largement utilisée sur les données d'expression de gènes. Le clustering hiérarchique ascendant regroupe successivement les classes d'éléments les plus proches en formant ainsi un dendogramme. Malheureusement, dans l'approche proposée par les auteurs, les conditions expérimentales et les gènes sont partitionnés de manière complètement indépendante. L'avantage principal de cette méthode réside dans son passage à l'échelle c'est-à-dire que des données contenant des dizaines de milliers de gènes et des dizaines de colonnes peuvent être utilisées. Il faut noter aussi que le succès de cette méthode est principalement dû à sa faculté à offrir une visualisation simple et intuitive des motifs extraits. La figure 1.5 montre un exemple de cette méthode. On peut voir (en haut) un dendogramme issu de la classification des colonnes (des gènes) et un autre (sur le côté) issu des lignes.

Bi-ensemble plus réaliste

Cheng et Church [34] proposent un modèle un peu plus satisfaisant pour extraire les groupes de synexpression. Cette méthode est une amélioration du modèle proposé par Hartigan. En effet, les réponses transcriptionnelles des gènes ne sont pas identiques dans toutes les conditions biologiques. Deux gènes peuvent répondre dans une condition biologique mais à des niveaux différents. Deux conditions peuvent induire des réponses très différentes pour un même gène. Ainsi, si l'on s'intéresse à des bi-

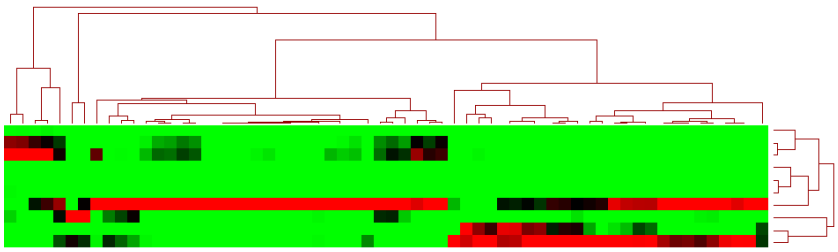


FIG. 1.5 – Bi-partitionnement pour le clustering hiérarchique sur les deux dimensions

ensembles représentant des groupes de synexpression, le modèle simple de Hartigan n'est pas complètement satisfaisant. Cheng et Church proposent d'ajouter à la valeur moyenne dans un bi-ensemble deux autres valeurs : une liée à l'influence du gène et l'autre liée à la condition expérimentale. Ils proposent d'utiliser pour ces deux valeurs la moyenne Moy_{Xy} sur le gène y (resp. Moy_{xY} sur la condition expérimentale x) des valeurs d'expression pour toutes les conditions expérimentales (resp. pour tous les gènes) contenues dans le bi-ensemble (X,Y). Ils utilisent la distance H suivante pour mesurer la qualité d'un bi-ensemble :

$$H(X,Y) = \frac{\sum_{x \in X, y \in Y} (Moy_{xy} - D_{x,Y} - D_{X,y} + D_{X,Y})^2}{\mid X \mid\mid Y \mid}$$

avec $D_{x,Y} = \frac{Moy_{x,Y}}{|Y|}$, $D_{X,y} = \frac{Moy_{X,y}}{|X|}$ et $D_{X,Y} = \frac{Moy_{X,Y}}{|Y||X|}$.

La qualité globale d'une collection est la somme des distances pour chaque motif de la collection. Les auteurs emploient le terme de résidu pour la différence entre la valeur attendue et celle qui est dans la matrice. Le modèle de Hartigan peut être retrouvé en réalisant un simple pré-traitement sur la matrice. Comme les moyennes sont calculées sur l'ensemble des lignes et/ou des colonnes, il suffit d'enlever à chaque valeur du tableau la moyenne des valeurs de sa ligne et la moyenne des valeurs de sa colonne pour retrouver exactement le résultat de [51].

En revanche, Cheng et Church proposent plusieurs heuristiques pour extraire les motifs. L'une d'entres elles consiste à enlever itérativement des gènes et des conditions expérimentales un à un jusqu'à ce que la mesure de distance soit inférieure à δ, c'est une approche divisive. Une limite de cette approche est que le nombre de bi-ensembles à rechercher est fixé par l'utilisateur tout comme le seuil δ utilisé pour la mesure de qualité. [100] généralise le travail réalisé par [34] en permettant la prise en compte des valeurs manquantes.

Lazzeroni et al. [61] proposent d'améliorer le modèle précédent. L'hypothèse qui est faite est que si un gène peut intervenir dans différents phénomènes biologiques

alors son niveau d'expression est lié à la combinaison de ces phénomènes. Pour capturer ces phénomènes, il faut non pas chercher des bi-ensembles presque "constants", mais ceux dont les valeurs résultent de cette combinaison. La figure 1.6 montre un exemple de deux bi-ensembles dont la valeurs (5) qui appartient aux deux bi-ensembles est la somme de la valeur du premier (2) et du deuxième (3).

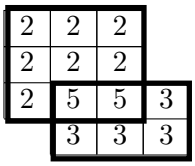


FIG. 1.6 – Exemple de deux bi-ensembles

Ainsi, d'une manière algébrique le niveau d'expression d'un gène i dans une condition expérimentale j est modélisé par

$$Y_{ij} = \mu_0 + \sum_k (\mu_k + \alpha_{ik} + \beta_{jk})\, \rho_{ik}\psi_{jk}$$

où μ_0 représente le bruit de fond, μ_k la couleur du calque k, ρ_{ik} vaut 1 si i appartient au bi-ensemble k et 0 sinon, ψ_{jk} vaut 1 si la condition expérimentale j appartient au bi-ensemble k, et vaut 0 sinon, α_{ik} est le facteur correctif pour le gène i, et β_{jk} est un facteur correctif pour la condition expérimentale j. La méthode consiste alors à rechercher le modèle minimisant la distance euclidienne entre les valeurs d'expression observées et celles modélisées. L'estimation des paramètres se fait itérativement et ne produit qu'une valeur approchée. Les expérimentations fournies dans l'article semblent produire des résultats intéressants. Cette méthode est similaire aux méthodes de décomposition en valeurs singulières [56], mais ici les vecteurs ne sont pas contraints à être orthogonaux entre eux.

Califano et al. [33] présentent une autre façon de définir des bi-ensembles presque "constants". Les bi-clusters doivent contenir des valeurs comprises dans un intervalle de taille σ. Cette fois-ci, la mesure utilisée pour mesurer la qualité des motifs est la probabilité d'apparition "par chance" d'un motif. Cette probabilité est calculée à partir d'un ensemble contrôle. Cette méthode est une méthode supervisée.

Méthodes basées sur la théorie de l'information

Deux variantes d'une même méthode de bi-partitionnement ont été développées de manière indépendante par Dhillon et al. [39] et Robardet et al. [86, 88]. Cette méthode consiste à considérer les deux partitions cherchées comme des variables aléatoires à valeurs discrètes et à concevoir la recherche d'une bi-partition comme un problème de maximisation de l'association entre ces deux variables. Il existe différentes mesures

d'association qui évaluent le lien entre deux variables aléatoires à partir (a) d'un tableau de contingence si les deux variables ont même domaine, ou (b) d'un tableau de co-occurrence comme dans le cas de la recherche d'une bi-partition. La construction de ce tableau nécessite que les variables soient à valeurs booléennes, ce qui peut être généralisé au cas où les variables sont discrètes [85]. L'idée est d'associer à chaque colonne une classe de la partition des conditions expérimentales, et à chaque ligne une classe de gènes à laquelle est associé un niveau d'expression. Un élément (i, j) du tableau est alors égal au nombre de co-occurrences d'une condition expérimentale de la classe correspondant à la colonne j et du niveau d'expression d'un gène de la classe i. Ce tableau permet d'estimer empiriquement la distribution de la probabilité jointe entre les deux variables représentant les partitions. Dhillon et al. [39] utilisent la mesure de divergence entre distributions de probabilités de Kullback et Leibler. Cependant, il a été montré, d'un point de vue théorique et expérimental [85], que les mesures de connexion étaient mieux adaptées que les mesures de divergences pour la recherche d'une bi-partition optimale. Robardet utilise dans [85] la mesure de connexion τ de Goodman et Kruskal [49] pour évaluer la qualité de la bi-partition. Chacune des deux méthodes produit une partition par un processus d'optimisation locale : [39] propose de fixer a priori le nombre de classes de chacune des deux partitions et optimisent localement la fonction en estimant itérativement une partition en fonction de l'autre jusqu'à convergence ; [85] ne fixe pas a priori le nombre de classes des deux partitions et utilise alors un algorithme d'optimisation local stochastique qui procède également par ajustement itératif d'une partition en fonction de l'autre.

L'article [4] présente la recherche de bi-partitions (X,Y) optimales comme la recherche de bi-ensembles contenant le maximum d'information à l'intérieur par rapport à l'information contenu à l'extérieur du bi-ensemble. Il montre comment utiliser la divergence de Bregman pour essayer de résoudre ce problème. Cette divergence est de plus une généralisation d'un grand nombre de mesures habituellement utilisées dans ce type de problème.

1.4.2 Méthodes complètes

Wang et al. [96] proposent une méthode exacte pour extraire des bi-ensembles presque "constants" appelés *pCluster* (pattern Cluster). Le modèle impose que les moyennes des valeurs des colonnes et/ou des lignes doivent appartenir à un intervalle de taille σ. Pour pouvoir extraire tous les motifs sans parcourir tout l'espace de recherche des bi-ensembles, ils proposent d'ajouter une contrainte supplémentaire aux bi-ensembles : ils imposent que tous les bi-ensembles de taille 2*2 inclus dans un *pCluster* doivent aussi satisfaire le modèle. Ainsi si l'un de ces bi-ensembles ne le satisfait pas alors il n'est pas nécessaire de regarder ses sur-ensembles. Plus précisément, tous les bi-ensembles de taille 2*2 $(\{o_1, o_2\}, \{a_1, a_2\})$ inclus dans un *pCluster* doivent satisfaire la contrainte suivante :

$M_{o_1a_1} + M_{o_2a_2} - M_{o_1a_2} - M_{o_2a_1} < \sigma$

Ils cherchent de plus à extraire les bi-ensembles qui sont maximaux au niveau des colonnes pour un ensemble d'objets donné. Cet algorithme permet en plus d'imposer que les *pCluster* aient une taille minimale sur les lignes et les colonnes, contrainte exploitée efficacement pendant l'extraction. L'algorithme permet de ne pas calculer toutes les paires de lignes et de colonnes.

Pour extraire ces motifs, ils utilisent une méthode complète basée sur la recherche des motifs contenant 2 gènes et 2 conditions expérimentales et les éléments de l'autre dimension qui satisfont la contrainte précédente. Ces bi-ensembles sont ensuite utilisés pour générer des bi-ensembles plus grands.

[93] présente un algorithme basé sur la théorie des graphes. Les données sont représentées sous la forme d'un graphe biparti. Un graphe biparti est un graphe tel que les nœuds sont découpés en deux ensembles disjoints U et V tels qu'il n'y ait pas d'arêtes entre les nœuds de U (respectivement de V). Pour les données d'expression, U représente l'ensemble des gènes et V l'ensemble des conditions. Une arête existe entre un gène g et une condition c ssi le niveau d'expression de g est significatif pour c. Ils cherchent en fait dans les données des bi-ensembles maximaux. Ce problème peut être formalisé comme la recherche de bi-cliques maximales dans les graphes bipartis. Les bi-cliques (X, Y) sont des sous-graphes bipartis tels qu'il existe une arête entre chaque nœud de X et chaque nœud de Y. Les auteurs présentent ce problème comme étant trop compliqué a résoudre. Ils se limitent alors aux graphes ayant que des nœuds avec un petit degré maximal. Le degré maximal est le nombre maximum d'arêtes issues d'un nœud.

La recherche de toutes les bi-cliques maximales dans un graphe biparti peut être réalisable en pratique, mais elle nécessite une approche non naïve de l'énumération des candidats. Ce problème est équivalent a la recherche de tous les concepts formels contenus dans une matrice booléenne. C'est d'ailleurs sur ce problème qu'ont porté nos travaux de recherche. La prochaine section présente notre approche de l'extraction de motifs dans les données d'expression de gènes.

1.5 Notre approche de la fouille de données

1.5.1 Exemples de questions biologiques

Nous présentons 4 exemples de questions (requêtes) biologiques. Nous montrons ensuite comment les motifs locaux ensemblistes permettent d'apporter des réponses à ces questions. Les questions sont les suivantes :

– Quels sont les ensembles de gènes qui varient significativement et simultanément dans au moins 2 conditions en réponse à l'insuline ? (Requête 1)

- Est-ce que la présence de certains sites potentiels de facteurs de transcription dans les séquences promotrices des gènes ne pourraient pas expliquer les variations de certains ensembles de gènes en réponse à l'insuline ? (Requête 2)
- Peut-on trouver des ensembles de facteurs de transcription associés à des ensembles de gènes participant à une même fonction cellulaire ? (Requête 3)
- Quels sont les ensembles de gènes qui sont régulés chez l'homme en présence d'insuline et qui ont un homologue chez la souris ? (Requête 4)

De nombreuses informations peuvent être utilisées pour répondre à ces questions : données d'expression, facteurs de transcription, fonctions des gènes, localisation chromosomique, séquences homologues dans d'autres espèces, etc . De très nombreuses bases de données disponibles sur internet contiennent ces données : séquences (PIR-NBRF, Swissprot, EMBL-GenBank-DDBJ, Flybase-Drosophile, MGD-souris, GDB-humain, PROSITE et eMOTIF), métabolisme (KEGG, BRENDA, EMP, Enzyme et EcoCyc), Régulation transcriptionnelle (RegulonDB), interaction protéine-protéine (PDB et Ec to TDB) et données structurales (PKR et 5s Ribosomal RNA Database).

Pour illustrer notre approche, nous allons utiliser 5 jeux de données artificielles.

Jeux de données

Le premier jeu de données noté $\mathbf{r}_{D1}$(figure 1.7 à gauche) représente des expériences de puces à ADN. A chaque gène g_i est associé sa variation d'expression (en log_2) en réponse à l'insuline (avant et après injection d'insuline) dans le muscle pour 5 individus I_j. Pour cette expérience, 6 gènes ont été testés. Le deuxième jeu de données noté $\mathbf{r}_{D2}$ (figure 1.7 à droite) est composé de données SAGE et représente le nombre de copies pour 4 TAG (un TAG représente un ARNm associé à un gène) dans 4 librairies. Les librairies S_j sont relatives à des tissus musculaires pour 4 individus. Le troisième jeu de données $\mathbf{r}_{D3}$ (figure 1.8 à gauche) est composé de 5 lignes correspondant à des facteurs de transcription et 5 colonnes correspondants à des gènes. Un "1" dans la matrice entre un gène g_i et un facteur de transcription FT_j indique que FT_j peut s'accrocher sur la région promotrice du gène g_i. Ce facteur de transcription peut alors potentiellement réguler la transcription du gène g_i. Le quatrième jeu de données $\mathbf{r}_{D4}$ (figure 1.8 à droite) est composé de 4 lignes, des fonctions moléculaires, et de 4 colonnes, des gènes. Ce tableau indique la fonction de chaque gène. Le dernier jeu de données $\mathbf{r}_{D5}$ (figure 1.9) indique si un gène humain (en colonne) est homologue à un gène d'une espèce donnée (en ligne) en indiquant si c'est le cas le numéro du chromosome puis la position de début et de fin du gène homologue sur ce chromosome.

Puces à ADN						
ID	g_1	g_2	g_3	g_4	g_5	g_6
P_1	-2.2	1.8	-0.5	1.7	0.1	0.2
P_2	2.01	1.6	-2.3	-0.78	1.58	2.6
P_3	2.1	-1.62	-1.71	2.1	-0.25	-1.2
P_4	2.2	1.2	-0.2	0.3	-0.25	-
P_5	0.1	-1.63	-0.4	0.21	-3.25	-

SAGE				
ID	g_1	g_2	g_3	g_4
S_1	80	75	39	10
S_2	12	85	42	20
S_3	35	80	40	92
S_4	100	10	45	113

FIG. 1.7 – Données $\mathbf{r}_{D1}$ (gauche) et données $\mathbf{r}_{D2}$ (droite)

Facteurs de Transcription				
ID	g_1	g_2	g_3	g_5
FT1	1	1	0	0
FT2	0	0	0	0
FT3	1	1	1	0
FT4	1	0	1	1
FT5	0	0	1	1

Fonctions				
ID	g_1	g_2	g_3	g_5
F_1	0	0	0	1
F_2	0	1	1	0
F_3	1	1	0	0
F_4	1	1	1	0

FIG. 1.8 – Données $\mathbf{r}_{D3}$ (gauche) et données $\mathbf{r}_{D4}$ (droite)

Requête 1

A partir de ces données, nous allons essayer de répondre à la première question : "Quels sont les ensembles de gènes qui varient significativement et simultanément dans au moins 2 conditions en réponse à l'insuline ?". La propriété qui nous intéresse pour cette question est "un gène varie significativement". Il faut alors encoder cette propriété dans le jeu de données $\mathbf{r}_{D1}$. Or, biologiquement, la variation significative n'a pas un sens absolu. Il est simplement admis qu'au delà d'une variation de 1.5 en valeur absolue, la variation est significative du point de vue de la technologie des puces à ADN. Nous allons ainsi utiliser ce seuil pour coder cette propriété et obtenir un nouveau jeu de données $\mathbf{r}_{D1bis}$ (figure 1.10). Cette première étape soulève le point

Homologies				
ID	g1	g2	g3	g4
Souris	(3,232,800)	(4,400,550)	-	(8,2000,2156)
Rat	-	-	(3,1000,1235)	(1,500,663)
singe	(4,5087,4078)	-	(3,1001,1523)	-

FIG. 1.9 – Données $\mathbf{r}_{D5}$

cruciale de l'encodage de propriétés. [6] et [80] ont proposé différentes solutions pour des données d'expression. [6] propose un ensemble de méthodes de discrétisation pour les données SAGE. [80] propose une approche originale visant à choisir parmi un ensemble de discrétisations (méthodes et paramètres) celle qui conserve au mieux les structures globales présentes dans les données. La méthode propose de faire un clustering hiérarchique sur les données brutes puis sur chaque jeu de données discrétisé, la discrétisation retenue est celle qui conserve au mieux le dendogramme associé au clustering hiérarchique.

Un autre problème concerne la présence de données manquantes. Effectivement, le jeu de données ne contient pas d'information pour le gène g_6 sur les puces $P4$ et $P5$. Pour cette requête, nous décidons de considérer seulement les gènes qui ne contiennent aucune valeur manquante. Le gène g_6 va donc être écarté de l'analyse. Il faut noter que l'on aurait pu choisir d'écarter les puces P_4 et P_5 au lieu du gène g_6. [84] présente une première approche visant à prendre en compte au cours de l'extraction de motifs les valeurs manquantes.

Propriété "varie significativement" pour $\mathbf{r}_{D1}$					
ID	g_1	g_2	g_3	g_4	g_5
P_1	1	1	0	1	0
P_2	1	1	1	0	1
P_3	1	1	1	1	0
P_4	1	0	0	0	0
P_5	0	1	0	0	1

FIG. 1.10 – Données $\mathbf{r}_{D1bis}$ qui encodent la variation significative pour le jeu de données $\mathbf{r}_{D1}$ pour la requête 1

A partir de $\mathbf{r}_{D1bis}$, les ensembles des gènes qui varient significativement et simultanément dans au moins 2 conditions en réponse à l'insuline peuvent être extraits au moyen de concepts formels. En revanche, nous n'allons nous intéresser qu'à certains concepts formels, ceux qui contiennent au moins 2 conditions expérimentales et 2 gènes. C'est un exemple de contraintes que l'on doit être capable d'exploiter au cours de l'extraction. La collection suivante E est la collection des concepts formels (X,Y) de $\mathbf{r}_{D1bis}$ tels que $\mid X \mid > 1$ et $\mid Y \mid > 1$:

$E = \{(\{P_1, P_2, P_3\}, \{g_1, g_2\}), (\{P_2, P_3\}, \{g_1, g_2, g_3\})$

Les gènes g_1 et g_2 (respectivement g_1, g_2 et g_3) ont effectivement une variation d'expression significative dans les conditions P_1, P_2 et P_3 (resp. P_2 et P_3).

Cette collection de motifs permet d'apporter des réponses à la première question en fournissant des associations présentes dans les données et donc des groupes de synexpression potentiels.

Requête 2

La deuxième question à laquelle nous allons nous intéresser est : "Est-ce que la présence de certains sites potentiels de facteurs de transcription dans les séquences promotrices des gènes ne pourrait pas expliquer les variations de certains ensembles de gènes en réponse à l'insuline ?". Pour répondre à cette deuxième requête, il va falloir exploiter à la fois les informations contenues dans la table $\mathbf{r}_{D1bis}$ et $\mathbf{r}_{D3}$ qui sont en fait toutes les deux des propriétés relatives aux gènes. Nous allons ainsi fusionner ces deux tables afin d'obtenir un contexte d'extraction unique appelé $\mathbf{r}_{req2}$ (voir figure 1.11). La table $\mathbf{r}_{D3}$ ne nécessite aucun encodage car elle contient déjà les informations nécessaires sous une forme booléenne.

Requête 2				
ID	g_1	g_2	g_3	g_5
P_1	1	0	1	0
P_2	1	1	1	1
P_3	1	1	1	0
P_4	1	0	0	0
P_5	0	1	0	1
FT_1	1	1	0	0
FT_2	0	0	0	0
FT_3	1	1	1	0
FT_4	1	0	1	1
FT_5	0	0	1	1

FIG. 1.11 – Contexte d'extraction $\mathbf{r}_{req2}$ pour la requête 2

Cette nouvelle table est particulière par rapport aux précédentes car la sémantique associée à un "1" dans la table diffère en fonction des lignes. Cette différence va bien évidemment se répercuter sur la manière dont doit être interprété un motif qui en découle. Il faut noter que cette fois-ci les gènes g_4 et g_6 ne sont pas présents dans $\mathbf{r}_{req2}$. En effet, aucune information sur leurs facteurs de transcription n'est disponible dans nos données. C'est un deuxième cas où l'on enlève des gènes de l'analyse.

Comme pour la requête 1, nous allons utiliser les concepts formels. Cette fois-ci, nous allons extraire les concepts E de $\mathbf{r}_{req2}$ contenant au moins 2 conditions expérimentales et 2 facteurs de transcription.

$E = \{(\{P_2, P_3, FT_1, FT_3\}, \{g_1, g_2\})\}$

Les gènes g_1 et g_2 varient significativement dans les conditions expérimentales P_2 et P_3 et cette co-variation peut être expliquée par la présence des facteurs de transcription FT_1 et FT_3. En effet, on peut suspecter que g_1 et g_2 sont en fait co-

régulés par FT_1 et FT_3.

Requête 3

La troisième question est : "Peut-on trouver des ensembles de facteurs de transcription associés à des ensembles de gènes participant à une même fonction moléculaire ?". Pour répondre à la requête 3, nous allons utiliser les règles d'association sur les tables $\mathbf{r}_{D3}$ et $\mathbf{r}_{D4}$. Nous avons fusionné les deux tables et transposé la matrice. La table $\mathbf{r}_{req3}$ est ainsi obtenue (voir figure 1.12).

Requête 3									
ID	FT_1	FT_2	FT_3	FT_4	FT_5	F_1	F_2	F_3	F_4
g_1	1	0	1	1	0	1	0	1	1
g_2	1	0	1	0	0	0	1	1	1
g_3	0	0	1	1	1	0	1	0	1
g_5	0	0	0	1	1	1	0	0	0

FIG. 1.12 – Contexte d'extraction $\mathbf{r}_{req3}$ pour la requête 3

Nous allons extraire les règles d'association R avec une fréquence minimale de 2/4 et une confiance minimale de 50% ayant comme partie gauche seulement des facteurs de transcription et comme partie droite des fonctions. Une règle d'association $A \Longrightarrow B$ avec une fréquence $freq \in [0, 100]$ et une confiance $conf \in [0, 100]$ indique que $freq$% des lignes n'ont que des "1" pour les colonnes de $A \cup B$ et que $conf$% des lignes n'ont que des "1" pour les colonnes B quand il y a des "1" pour les colonnes de A. Une règle avec une confiance de 100% est une règle logique.

$\{FT_1, FT_3\} \Longrightarrow \{F_3, F_4\}$ est une règle contenue dans $\mathbf{r}_{req3}$ ayant une fréquence de 2 et une confiance de 100%. Cette règle permet de suspecter que les facteurs de transcription FT_1 et FT_3 pourraient jouer un rôle particulier dans les fonctions F_3 et F_4. La règle $\{FT_3\} \Longrightarrow \{F_3, F_4\}$ avec une fréquence de 2 et une confiance de 2/3 est un autre exemple de règles. Rappelons que l'on extrait toutes les règles d'association ayant une fréquence de 2 et une confiance de 100%.

Pour cette requête, plusieurs points importants ont été soulevés. D'abord, il a fallu transposer la matrice pour considérer des règles contenant des facteurs de transcription et des fonctions portées cette fois-ci par les gènes. La transposition de la matrice n'est pas seulement une réorganisation graphique de la table mais a pour conséquence une modification de la faisabilité de l'extraction et de l'utilisation des contraintes. Dans cet exemple d'autres contraintes que la taille minimale ont été utilisées. Des contraintes particulières comme la présence de tels ou tels éléments dans tels ou tels ensembles de fréquence et de confiance minimale ont été utilisées pour pouvoir extraire des motifs plus pertinents de notre point de vue vis-à-vis de la question posée.

Requête 4

La quatrième requête est la suivante : "Quels sont les ensembles de gènes qui sont régulés chez l'homme en présence d'insuline et qui ont un homologue chez la souris ?". Pour répondre à cette requête, deux jeux de données vont être utilisés : $\mathbf{r}_{D2}$ et $\mathbf{r}_{D5}$. Pour la première table, il faut encoder la propriété "un gène est activé dans une condition expérimentale", pour la seconde table l'information "un gène humain est homologue à un gène de souris" doit aussi être encodée. Pour les données SAGE, nous allons utiliser la méthode appelée $Max(1-X)$ présentée dans l'article [6]. Un gène est dit activé si son nombre de copies est supérieur à $Max(1-X)$ avec Max le nombre maximum de copies de ce gène pour toutes les librairies et $X \in [0;1]$ un paramètre. La table de gauche de la figure 1.13 représente cet encodage avec $X = 0.25$. Pour cette requête, nous avons décidé d'encoder cette propriété sur l'ensemble des données puis de sélectionner celles qui concernent la souris. Pour la table $\mathbf{r}_{D5}$, seule l'information sur la présence d'un gène homologue est nécessaire. La table de droite $\mathbf{r}_{D5bis}$ de la figure 1.13 encode cette information. On parle bien d'encodage et non uniquement de discrétisation car les informations dans les tables peuvent prendre de nombreuses formes (pas seulement des réels). Il faut noter que dans la table $\mathbf{r}_{D5}$, le symbole "-" signifie "pas homologue" alors que dans la table $\mathbf{r}_{D1}$ il signifie que c'est une valeur manquante. Cela illustre simplement la disparité des formats des données que l'on doit manipuler.

gènes activés				
ID	g_1	g_2	g_3	g_4
S_1	1	1	1	0
S_2	0	1	1	0
S_3	0	1	1	1
S_4	1	0	1	1

homologue				
ID	g_1	g_2	g_3	g_4
$Souris$	1	1	0	1

FIG. 1.13 – $\mathbf{r}_{D2bis}$: encodage de $\mathbf{r}_{D2}$ à gauche et $\mathbf{r}_{D5bis}$: encodage de l'homologie à droite

Pour répondre à notre question, il faut en fait générer un contexte d'extraction contenant les deux premières lignes de $\mathbf{r}_{D2bis}$ qui concernent la souris et la première ligne de la table $\mathbf{r}_{D5bis}$. La figure 1.14 présente cette table.

Nous pouvons extraire tous les concepts formels contenant la ligne "souris", c'est-à-dire dont les gènes humains sont homologues à ceux de la souris. En fait, le contexte d'extraction précédent peut être simplifié car seuls les gènes ayant un "1" sur la troisième ligne vont être considérés. La figure 1.15 montre le nouveau contexte d'extraction ainsi obtenu. C'est un exemple typique de simplification de matrice.

L'ensemble des concepts formels contenus dans ce contexte est :

contexte 1				
ID	g_1	g_2	g_3	g_4
S_1	1	1	1	0
S_2	0	1	1	0
$Souris$	1	1	0	1

FIG. 1.14 – Contexte d'extraction pour la requête 4

contexte 1			
ID	g_1	g_2	g_4
S_1	1	1	0
S_2	0	1	0

FIG. 1.15 – Contexte d'extraction simplifié

$\{(\{S_1\}, \{g_1, g_2\}), (\{S_1, S_2\}, \{g_2\})\}$

Cette collection de motifs permet d'apporter des réponses potentielles à la requête 4. Par exemple, g_1 et g_2 sont des gènes qui sont activés en présence d'insuline et homologues à des gènes de souris.

1.5.2 Retour sur les scénarios d'extraction

Les questions précédentes ont permis de mettre en avant le potentiel des motifs locaux ensembliste dans des données. Pour répondre à ces questions, différentes primitives ont été utilisées qui peuvent être classées en deux groupes : la préparation des contextes d'extraction et l'extraction de motifs sous contraintes. Pour la préparation des contextes, les primitives suivantes ont été utilisées : encodage de propriétés d'expression, fusion de jeux de données, transposition de matrices, sélection d'éléments suivant différents critères. Pour les extractions, on a recherché des motifs satisfaisant certaines contraintes. Les concepts formels et les règles d'association ont été utilisés mais d'autres types de motifs auraient pu être utilisés. Sur des applications réelles, il faudrait en plus pouvoir par exemple pré-traiter les données, comparer différentes collections de motifs issus de traitements différents et sélectionner certains motifs selon des critères. Il est encore très difficile de pouvoir définir un langage de requêtes pour les bases de données inductives permettant de répondre à toutes les tâches d'extraction de connaissances. Malgré cela, les avancées qui ont été réalisées à la fois sur la résolution de tâches primitives, l'étude des scénarios d'extraction [79, 76] et l'optimisation de requêtes [37, 38, 50, 66, 67] indiquent clairement l'émergence des bases de données inductives. Même si un système opérationnel intégrant un véritable langage de requêtes utilisant une multitude de primitives est encore hors de portée

actuellement, un système plus limité peut être mis en place. Par exemple pour les données transcriptomiques, un jeu de données intégrant une multitude d'informations disponibles dans les bases de données peut être produit et les différents outils de prétraitement, d'extraction de motifs et de post-traitement déjà développés, permettent déjà de répondre à des questions non triviales. Nous avons d'ailleurs développé un logiciel appelé Bio++ [10] dédié à l'extraction de connaissances dans les données d'expression de gènes. Il regroupe différentes méthodes de discrétisaton, d'extracteurs de motifs, de post-traitement de motifs et finalement un outil de visualisation des motifs.

Chapitre 2

Extraction de concepts formels

2.1 Définitions et propriétés des motifs locaux

2.1.1 Introduction

Nous allons dans cette section présenter le cadre de la recherche des motifs locaux dans les données booléennes. Cet axe de recherche est aussi appelé "extraction d'itemsets fréquents" même si depuis son émergence [2] il ne se résume plus au calcul des itemsets fréquents, d'autres types de motifs ayant été proposés. Les données représentent une relation binaire $\mathbf{r}$ entre un ensemble d'objets noté $\mathcal{O}$ et un ensemble d'attributs noté $\mathcal{A}$. La relation $\mathbf{r}$ encode un lien ou une association entre un attribut et un objet. La sémantique de cette relation peut différer en fonction des attributs et objets considérés, ainsi que du type d'association étudié par l'utilisateur. Toute propriété entre des attributs et des objets qui peut être représentée sous la forme d'une relation peut bénéficier des avancées technologiques relatives à l'extraction des motifs locaux dans les données booléennes. Nous utiliserons le terme données booléennes ou données transactionnelles pour désigner ce type de jeux de données. Une représentation sous forme matricielle est souvent adoptée, les objets (resp. les attributs) peuvent alors être appelés des lignes (resp. des colonnes). Pour simplifier les notations, $\mathbf{r}$ sera utilisé pour désigner à la fois la relation et les données.

Exemple. Soit un jeu de données $\mathbf{r}_1$ composé de 3 objets $\mathcal{O} = \{o_1 o_2 o_3\}$, de 4 attributs $\mathcal{A} = \{a_1 a_2 a_3 a_4\}$ tels que $\mathbf{r}_1 = \{(o_1, a_1),\ (o_1, a_2), (o_1, a_3), (o_2, a_1), (o_2, a_2),$-$(o_3, a_2), (o_3, a_3), (o_3, a_4)\}$. La table 2.1 est la représentation matricielle de $\mathbf{r}_1$.

Dans le cas des données d'expression (voir par exemple [6, 81]), les colonnes représentent généralement des gènes, alors que les lignes représentent des informations relatives aux gènes. Suivant la nature des lignes, la relation $\mathbf{r}$ peut exprimer par exemple la variation d'un gène dans une condition expérimentale, le fait que le gène

	a_1	a_2	a_3	a_4
o_1	1	1	1	0
o_2	1	1	0	0
o_3	0	1	1	1

TAB. 2.1 – Jeu de données $\mathbf{r}_1$

est régulé par un facteur de transcription ou l'appartenance d'un gène à une famille de fonctions.

Mannila et Toivonen ont proposé une abstraction utile de nombreux travaux en fouille de données [63]. Considérons une base de données $\mathbf{r}$, un langage $\mathcal{L}$ pour l'expression de propriétés dans les données et un prédicat de sélection $\mathcal{C}$. Le prédicat $\mathcal{C}$ est utilisé pour dire si oui ou non, une phrase $l \in \mathcal{L}$ doit être considérée comme intéressante sur $\mathbf{r}$. Une tâche d'extraction peut alors être formalisée comme le calcul de la théorie de $\mathbf{r}$ pour $\mathcal{L}$ et $\mathcal{C}$, i.e., l'ensemble $\mathcal{TH}(\mathbf{r}, \mathcal{L}, \mathcal{C}) = \{l \in \mathcal{L} \mid \mathcal{C}(l, \mathbf{r}) \text{ est vrai}\}$. On peut parler de la requête inductive $\mathcal{C}$ sur $\mathbf{r}$.

2.1.2 Bi-ensemble et 1-rectangle

Pour notre part, nous allons nous intéresser au calcul de théories de la forme $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C})$ avec $\mathbf{r} \subseteq \mathcal{O} \times \mathcal{A}$. Nous allons ainsi rechercher tous les bi-ensembles (voir définition 2.1) satisfaisant $\mathcal{C}$ avec $\mathbf{r}$ une relation binaire entre $\mathcal{O}$ et $\mathcal{A}$.

Définition 2.1 (bi-ensemble) *Un bi-ensemble (X, Y) est un couple appartenant à $2^{\mathcal{O}} \times 2^{\mathcal{A}}$. Le terme de rectangle peut aussi être utilisé dans la mesure où il n'y pas d'ordre a priori sur les lignes et les colonnes. Les bi-ensembles sont des rectangles dans la matrice à un réarrangement près des lignes et des colonnes.*

L'extraction de motifs locaux dans des données transactionnelles consiste donc à extraire des bi-ensembles satisfaisants certaines contraintes. Deux grandes classes de contraintes peuvent être considérées : les contraintes qui fixent le type de motifs à extraire et les contraintes qui imposent "la forme" des motifs à extraire. Ces contraintes seront appelées respectivement contraintes de type (voir section 2.1.4) et contraintes syntaxiques (voir section 2.1.3). Par exemple, les contraintes de type permettent de spécifier que l'on cherche des 1-rectangles, des itemsets, des ensembles fermés, des ensembles libres ou des concepts formels (voir section 2.1.4). Les contraintes syntaxiques permettent par exemple de fixer la taille minimale (nombre d'objets et nombre d'attributs) des bi-ensembles à extraire ou la présence ou non de certains éléments (voir 2.1.3). Il faut noter que les contraintes de type sont liées aux données alors que

les contraintes syntaxiques ne dépendent souvent que des éléments contenus dans le bi-ensemble.

Nous cherchons ainsi à extraire des théories de la forme $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C})$ avec $\mathcal{C} \equiv \mathcal{C}_t(l, \mathbf{r}) \wedge \mathcal{C}_s(l)$. $\mathcal{C}_t$ est une contrainte de type et $\mathcal{C}_s$ est une contrainte syntaxique.

Nous allons nous intéresser plus particulièrement à certains motifs appelés 1-rectangles (voir définition 2.2).

Définition 2.2 (1-rectangle) *Un bi-ensemble (X, Y) est un 1-rectangle dans $\mathbf{r}$ s'il satisfait la contrainte $\mathcal{C}_{\mathbf{r}-1r} \equiv \forall x \in X,\ \forall y \in Y,\ (x, y) \in \mathbf{r}$.*

Exemple. Le bi-ensemble $(\{o_1o_2\}, \{a_1a_2\})$ est un 1-rectangle dans $\mathbf{r}_1$. Par contre, le bi-ensemble $(\{o_1o_2\}, \{a_1a_2a_3\})$ n'est pas un 1-rectangle dans $\mathbf{r}_1$ car $(o_2, a_3) \notin \mathbf{r}_1$.

Ces motifs sont particulièrement intéressants car ils permettent d'identifier des ensembles d'objets X et d'attributs Y qui sont en relation c'est-à-dire tels que tous les objets de X soient en relation avec tous les attributs de Y et inversement. Par exemple dans les données d'expression booléennes (un "1" exprime une variation d'expression pour un gène dans une condition), un 1-rectangle (X,Y) exprime le fait que tous les gènes de Y varient dans toutes les conditions expérimentales de X : les gènes de Y sont co-exprimés dans les conditions de X.

Nous allons munir les bi-ensembles de relations de spécialisation.

Définition 2.3 (Relations de spécialisation) *La relation d'inclusion ensembliste $\subseteq$ sera utilisée comme relation de spécialisation sur les ensembles. La même notation sera adoptée pour les bi-ensembles : $(X_1, Y_1) \subseteq (X_2, Y_2)$ ssi $X_1 \subseteq X_2$ et $Y_1 \subseteq Y_2$. Nous utilisons aussi une autre relation de spécialisation notée $\preceq$ sur les bi-ensembles : $(X_1, Y_1) \preceq (X_2, Y_2)$ ssi $X_2 \subseteq X_1$ et $Y_1 \subseteq Y_2$.*

Le point central pour calculer les théories de la forme $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C}_t \wedge \mathcal{C}_s)$ est l'aspect combinatoire de la recherche. Dès que $2^{\mathcal{O}} \times 2^{\mathcal{A}}$ est très grand, il est important d'avoir des stratégies pour parcourir l'ensemble des phrases possibles, i.e., l'espace de recherche. Depuis la formalisation de l'apprentissage comme un problème de recherche dans un espace d'états [69], on sait qu'il est souvent possible d'optimiser le parcours de l'espace de recherche pourvu qu'il soit associé à une relation de spécialisation $\prec$ (ordre partiel) et que les motifs recherchés soient définis à partir de contraintes que l'on sache exploiter [91]. En particulier, deux types de contraintes jouent un rôle particulier dans les extractions : les contraintes dites anti-monotones et monotones (voir définition 2.4).

Définition 2.4 (Contraintes anti-monotones et monotones) *Soit un ensemble partiellement ordonné $(\mathcal{E}, \leq_{\mathcal{E}})$ et une contrainte $\mathcal{C}$ alors :*

- $\mathcal{C}$ *est anti-monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$ *ssi* $\forall\, X,\, Y \in \mathcal{E}$ *tel que* $X \leq_{\mathcal{E}} Y$ *alors* $\mathcal{C}(Y) \Rightarrow \mathcal{C}(X)$. *La contraposée de cette définition est* $\forall\, X,\, Y \in \mathcal{E}$ *tel que* $X \leq_{\mathcal{E}} Y$ *alors* $\neg\mathcal{C}(X) \Rightarrow \neg\mathcal{C}(Y)$
- $\mathcal{C}$ *est monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$ *ssi* $\forall\, X,\, Y \in \mathcal{E}$ *tel que* $X \leq_{\mathcal{E}} Y$ *alors* $\mathcal{C}(X) \Rightarrow \mathcal{C}(Y)$. *La contraposée de cette définition est* $\forall X, Y \in \mathcal{E}$ *tel que* $X \leq_{\mathcal{E}} Y$ *alors* $\neg\mathcal{C}(Y) \Rightarrow \neg\mathcal{C}(X)$

Propriété 2.1 *Si* $\mathcal{C}$ *est monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$ *alors* $\neg\mathcal{C}$ *est anti-monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$. *Si* $\mathcal{C}$ *est anti-monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$ *alors* $\neg\mathcal{C}$ *est monotone sur* $\mathcal{E}$ *par rapport à* $\leq_{\mathcal{E}}$. *Une disjonction de conjonction de contraintes monotones (resp. anti-monotones) est monotone (resp. anti-monotone).*

Il faut noter que la contrainte $\mathcal{C}_{\mathbf{r}-1r}$ est une contrainte anti-monotone sur $\subseteq$.

Exemple 2.1 *Soit les bi-ensembles* $B_1 = (\{o_1o_2\}, \{a_1a_2a_3\})$ *et* $B_2 = (\{o_1o_2\}, \{a_1a_2\})$ *dans* $\mathbf{r}_1$, *alors* B_2 *satisfait* $\mathcal{C}_{\mathbf{r}1-1r}$ *mais pas* B_1. *Comme* $\mathcal{C}_{\mathbf{r}1-1r}$ *est monotone pour* $\subseteq$ *alors tout sur-ensemble de* B_1 *suivant* $\subseteq$ *ne satisfait pas* $\mathcal{C}_{\mathbf{r}-1r}$ *et tout sous-ensemble de* B_2 *suivant* $\subseteq$ *satisfait* $\mathcal{C}_{\mathbf{r}-1r}$. *En effet, comme* B_1 *contient un "0" alors tous ses sur-ensembles en contiennent aussi et donc ne satisfont pas* $\mathcal{C}_{\mathbf{r}-1r}$. *Comme* B_2 *ne contient pas de "0" alors tous ses sous-ensembles n'en contiennent pas non plus.*

2.1.3 Contraintes syntaxiques

Les définitions 2.5 et 2.6 donnent des exemples de contraintes syntaxiques respectivement monotones et anti-monotones par rapport à $\subseteq$.

Définition 2.5 (Exemples de contraintes syntaxiques monotones) *Un bi-ensemble* (X, Y) *satisfait :*

- $\mathcal{C}_{\sigma_1\sigma_2-mis}$ *(taille minimale) ssi* $\sharp X \geq \sigma_1$ *et* $\sharp Y \geq \sigma_2$.
- $\mathcal{C}_{AB-sur}$ *(sur-ensemble) ssi* $A \subseteq X$ *et* $B \subseteq Y$
- $\neg\mathcal{C}_{AB-sous}$ *(pas un sous-ensemble) ssi* $X \not\subseteq A$ *ou* $Y \not\subseteq B$
- $\mathcal{C}_{AB-ec}$ *(des éléments en commun) ssi* $X \cap A \neq \emptyset$ *ou* $Y \cap B \neq \emptyset$
- $\mathcal{C}_{\sigma-area}$ *(aire minimale) ssi* $\mid X \mid \times \mid Y \mid \geq \sigma$.

Définition 2.6 (Exemples de contraintes syntaxiques anti-monotones) *Un bi-ensemble* (X, Y) *satisfait :*

- $\mathcal{C}_{\sigma_1\sigma_2-mas}$ *(taille maximale) ssi* $\sharp X \leq \sigma_1$ *et* $\sharp Y \leq \sigma_2$.
- $\neg\mathcal{C}_{AB-sur}$ *(pas un sur-ensemble) ssi* $A \not\subseteq X$ *et* $B \not\subseteq Y$
- $\mathcal{C}_{AB-sous}$ *(sous-ensemble) ssi* $X \subseteq A$ *et* $Y \subseteq B$
- $\neg\mathcal{C}_{AB-ec}$ *(aucun élément en commun) ssi* $X \cap A = \emptyset$ *et* $Y \cap B = \emptyset$

La contrainte de fréquence minimale habituellement utilisée pour extraire les motifs locaux s'exprime grâce à $\mathcal{C}_{\sigma 0-mis}$ où σ est le seuil de fréquence minimale.

Exemple 2.2 *Dans* $\mathbf{r}_1$*, on peut rechercher la collection des 1-rectangles ayant une aire supérieure à 3 et contenant l'attribut* a_3*. Cette collection correspond à la théorie* $\mathcal{T}$ *suivante :*

$\mathcal{T} = \mathcal{TH}(\mathbf{r}_1, 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C}_{\mathbf{r}-1r} \wedge \mathcal{C}_{3-area} \wedge \mathcal{C}_{\emptyset\,\{a_3\}-ec})$

$\mathcal{T}$ *est égale à* $\{(\{o_1\}, \{a_1a_2a_3\}), (\{o_1o_3\}, \{a_2a_3\}), (\{o_3\}, \{a_2a_3a_4\})\}$.

2.1.4 Contraintes de type

La connexion de Galois (correspondance de Galois) est un outil clé pour définir, manipuler et utiliser les motifs locaux tels que les itemsets et les concepts formels. Nous donnons quelques rappels sur la connection de Galois (voir notamment [98]).

Définition 2.7 (Connection de Galois) *Soit* $\phi : \mathcal{O} \rightarrow \mathcal{A}$ *et* $\psi : \mathcal{A} \rightarrow \mathcal{O}$ *deux opérateurs entre deux ensembles partiellement ordonnés* $(\mathcal{O}, \leq_{\mathcal{O}})$ *et* $(\mathcal{A}, \leq_{\mathcal{A}})$*. Ces opérateurs forment une connection de Galois si :*

1 $\forall v, w \in \mathcal{O}$, *si* $v \leq_{\mathcal{O}} w$ *alors* $\phi(w) \leq_{\mathcal{A}} \phi(v)$,
2 $\forall i, j \in \mathcal{A}$, *si* $i \leq_{\mathcal{A}} j$ *alors* $\psi(j) \leq_{\mathcal{O}} \psi(i)$,
3 $\forall v \in \mathcal{O}, \forall i \in \mathcal{A}$, $v \leq_{\mathcal{O}} \psi(\phi(v))$ *et* $i \leq_{\mathcal{A}} \phi(\psi(i))$

où $\leq_{\mathcal{O}}$ *et* $\leq_{\mathcal{A}}$ *sont deux relations de spécialisation respectivement sur* $\mathcal{O}$ *et* $\mathcal{A}$*.*

Définition 2.8 (Connections ϕ et ψ) *Si* $X \subseteq \mathcal{O}$ *et* $Y \subseteq \mathcal{A}$*,* ϕ *et* ψ *peuvent être définis ainsi :* $\phi(X, \mathbf{r}) = \{y \in \mathcal{A} \mid \forall x \in X,\ (x, y) \in \mathbf{r}\}$ *et* $\psi(Y, \mathbf{r}) = \{x \in \mathcal{O} \mid \forall y \in Y,\ (x, y) \in \mathbf{r}\}$*.* ϕ *renvoie l'ensemble des attributs qui sont portés par tous les objets de* X*.* ψ *fournit l'ensemble des objets qui sont portés par l'ensemble des attributs de* Y*.* (ϕ, ψ) *forme une connection de Galois entre* $\mathcal{O}$ *et* $\mathcal{A}$ *munis de l'inclusion ensembliste* $\subseteq$*. Nous utilisons les notations classiques* $h = \phi \circ \psi$ *et* $h' = \psi \circ \phi$ *pour désigner les opérateurs de fermeture de Galois. Quand* $h(Y, \mathbf{r}) = Y$ *(resp.* $h'(X, \mathbf{r}) = X$*) on parle d'ensemble fermé.*

Exemple 2.3 *Dans* $\mathbf{r}_1$*, on a* $\phi(\{o_1o_2\}, \mathbf{r}_1) = \{a_1a_2\}$ *et* $\psi(\{a_2\}, \mathbf{r}_1) = \{o_1o_2o_3\}$.

A partir de cette connection (ϕ, ψ), les itemsets et les concepts formels peuvent être définis.

Définition 2.9 (itemsets et concepts formels) *Un bi-ensemble* (X, Y) *est un itemset s'il satisfait la contrainte* $\mathcal{C}_{\mathbf{r}-it} \equiv X = \psi(Y, \mathbf{r})$. *Une notation sous forme de bi-ensemble est adoptée ici pour les itemsets. Un bi-ensemble* (X, Y) *est un concept dans* $\mathbf{r}$ *s'il satisfait la contrainte* $\mathcal{C}_{\mathbf{r}-cf}$ *qui est équivalente à chacune des contraintes suivantes :*

- $X = \psi(Y, \mathbf{r}) \wedge Y = \phi(X, \mathbf{r})$
- $X = h'(X, \mathbf{r}) \wedge Y = \phi(X, \mathbf{r})$
- $Y = h(Y, \mathbf{r}) \wedge X = \psi(Y, \mathbf{r})$

Propriété 2.2 *Si* (X, Y) *est un concept formel alors* X *et* Y *sont des ensembles fermés.*

Une propriété importante de la connection de Galois est que chaque ensemble fermé sur l'une des deux dimensions est associé à un unique ensemble fermé de l'autre dimension. C'est pour cela, que l'on parlera essentiellement des concepts formels car les itemsets fermés sont contenus dans la notion de concepts formels.

Une autre propriété très importante de la connection de Galois est que les fonctions ϕ et ψ sont décroissantes. Cette propriété est cruciale pour l'exploration de l'espace de recherche et la propagation des contraintes. En effet, les concepts formels contenus dans un jeu de données peuvent être représentés sous la forme d'un treillis de concepts muni de la relation de spécialisation $\preceq$. La figure 2.1 montre le treillis de concepts associé à $\mathbf{r}_1$.

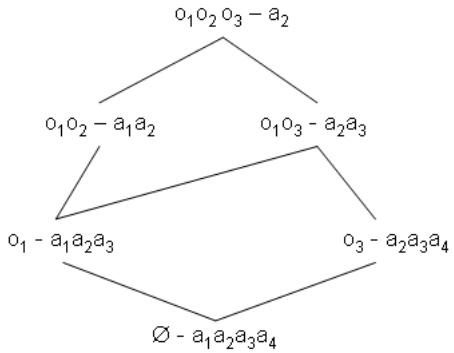


FIG. 2.1 – Treillis de concepts pour $\mathbf{r}_1$

Les ensembles libres [24] sont aussi des motifs intéressants dans les données booléennes. Les ensembles libres sont aussi appelés motifs clefs [5].

Définition 2.10 (Ensemble libre) *Un ensemble d'attributs* X *est libre si et seulement si* X *n'est pas inclus dans la fermeture de l'un de ses sous-ensembles stricts. On*

dira qu'un ensemble est libre s'il satisfait la contrainte $\mathcal{C}_{\mathbf{r}-libre}$. *D'autres formulations sont possibles :* X *est libre si sa fréquence dans* $\mathbf{r}$ *est strictement inférieure à celle de tous ses sous-ensembles stricts ou encore,* X *est libre s'il n'existe pas de règles d'association de confiance* 1 *contenant seulement des éléments de* X. *Ces ensembles sont aussi appelés ensembles clés dans [5].*

Définition 2.11 *Soit la relation d'équivalence* $\sim$ *sur les ensembles d'attributs telle que* $X \sim Y$ *avec* $X, Y \in \mathcal{A}$ *si et seulement si* X *et* Y *ont le même support [5]. Le support d'un ensemble d'attribut* Y *est* $\psi(Y, \mathbf{r})$. *Alors les ensembles libres et les ensembles fermés d'attributs sont des éléments particuliers de ces classes. En effet, ces classes d'équivalence sont formées d'un unique ensemble maximal, un ensemble fermé* F, *et d'un ou plusieurs éléments minimaux qui sont les ensembles libres ayant comme fermeture* F.

La figure 2.2 présente un exemple de classes d'équivalences. Les "patatoïdes" (figure 2.2 droite) représentent les classes. Tous les itemsets contenus dans une même classe ont un support identique. La taille du support est indiqué en dessous de chaque classe d'équivalence.

	G_1	G_2	G_3	G_4
T_1	1	0	1	0
T_2	1	1	1	1
T_3	0	1	1	1
T_4	0	0	0	1
T_5	1	0	0	0

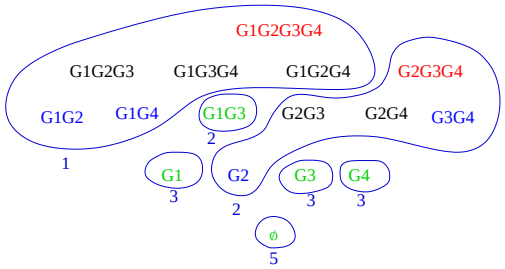


FIG. 2.2 – Un contexte booléen $\mathbf{r}$ (gauche) et ses classes d'équivalences associées (droite)

Exemple. $\{G_1G_2\}$ et $\{G_1G_4\}$ sont deux ensembles libres (éléments minimaux) et $\{G_1G_2G_3G_4\}$ correspond à leur fermeture (l'ensemble maximal). Ces trois ensembles définissent une classe d'équivalence.

Propriété 2.3 (ensembles libres et fermés) *La fermeture d'un ensemble libre donne un ensemble fermé et tous les ensembles fermés peuvent être obtenus par calcul des fermetures de tous les ensembles libres [23]. Les ensembles libres peuvent être utilisés comme des générateurs d'ensembles fermés.*

Les ensembles libres et leur fermeture comme les concepts formels sont des représentations condensées exactes des ensembles d'attributs. C'est-à-dire qu'à partir de la collection complète de ces motifs, il est possible de retrouver la fréquence de tous les ensembles d'attributs sans accéder aux données. Un autre type de motifs appelé δ-libres a été proposé comme une représentation approximative des ensembles d'attributs [31, 54, 5, 23]. Ils permettent sans accéder aux données de retrouver avec une erreur bornée la fréquence de tous les ensembles d'attributs.

Définition 2.12 (Ensemble δ-libre) *Un ensemble d'attributs X est un δ-libre si et seulement si $\forall Z \subset X$, $\sharp(\psi(Z) - \psi(X)) > \delta$.*

A chaque δ-libre K peut être associé un ensemble d'attributs C appelé δ-fermeture de K et défini par

$$C = \{g \in \mathcal{P} \mid \sharp(\psi(K) \setminus \psi(g)) \leq \delta\}$$

Quand $\delta = 0$, on a $C = h(K, \mathbf{r})$, les ensembles libres sont des cas particuliers des δ-libres.

2.2 Transposition de matrices

2.2.1 Introduction

Les technologies à haut-débit comme les puces à ADN ou SAGE fournissent le niveau d'expression de (dizaines de) milliers de gènes dans différentes conditions expérimentales. Ces technologies restent assez onéreuses et difficiles à mettre en œuvre. Il est ainsi très rare de pouvoir disposer pour une même problématique biologique de milliers de conditions expérimentales. En pratique, il est courant d'avoir une dizaine ou une vingtaine de conditions. Ainsi, les données peuvent être représentées sous la forme d'une matrice contenant une dizaine de lignes (les conditions) et des (dizaines de) milliers de colonnes (les gènes). Les données dont nous disposons pour étudier les mécanismes de la réponse à l'insuline sont typiquement de cette forme (matrice de taille $5 * 22069$, voir section 4.2.1).

Etant donné une matrice à fouiller, le problème de l'extraction de tous les itemsets satisfaisant une certaine contrainte syntaxique est très difficile dès lors que le nombre d'attributs dépasse quelques dizaines. En effet, la taille de la collection des itemsets, et donc de l'espace de recherche, s'exprime en une fonction exponentielle du nombre d'attributs et il n'est possible d'en explorer qu'une infime partie. Ensuite, la taille de la solution, i.e., la collection des ensembles satisfaisant la contrainte syntaxique peut être énorme, de sorte qu'aucun algorithme ne pourra les trouver tous. Lorsque la seule contrainte utilisée est une contrainte de fréquence minimale, on peut élever le seuil de fréquence pour diminuer a priori la taille de la solution et, pourvu que l'on sache

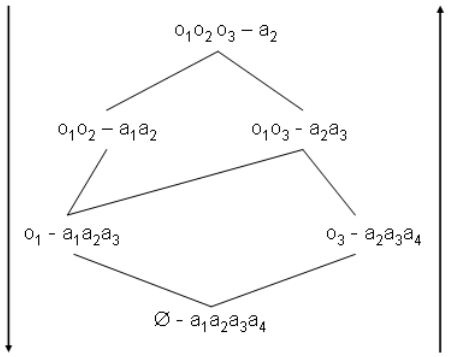


FIG. 2.3 – Deux sens de parcours du treillis de concepts

explorer intelligemment l'espace de recherche, calculer tous les ensembles fréquents. Le problème est alors que le seuil de fréquence utilisé peut ne pas être satisfaisant pour le biologiste. Par exemple, concernant le problème de la recherche de groupes de synexpression, avec une fréquence minimale importante seules des associations déjà bien connues peuvent être extraites (car trop fréquentes pour être inconnues).

Puisque les données d'expression qui nous intéressent concernent un nombre élevé de gènes, nous sommes clairement dans un cas difficile. Les espaces de recherche atteignent des tailles astronomiques (e.g., pour 900 gènes, il y a environ 10^{300} ensembles possibles). D'autre part, et c'est l'une des spécificités des données d'expression que nous voulons analyser, les tailles des collections des itemsets fréquents dans des matrices d'expression sont également énormes. Ceci provient du faible nombre de situations généralement considérées. En présence d'un petit nombre de situations (e.g., quelques dizaines), le nombre de fréquences possibles pour un ensemble de gènes est réduit et, quel que soit le seuil retenu, la collection des ensembles fréquents atteint elle aussi de très grandes tailles. Ainsi, tout algorithme qui, pour calculer des ensembles fréquents, doit, pour le moins, évaluer la fréquence de tous les ensembles fréquents (cas de toutes les variantes de l'algorithme APRIORI) ne pourra donc pas être utilisé dans ce contexte. Il est ainsi indispensable de se tourner vers les ensembles fermés [92] et les concepts formels qui sont des représentations condensées des itemsets. Or, même si ces approches ont donné d'excellents résultats dans de nombreux contextes d'extraction, elles échouent dans le cas des données d'expression de gènes. Ainsi, si l'on veut être capable d'extraire par exemple les groupes de synexpression potentiels dans de telles matrices (voir requête 1 de la section 1.5.1), il faut apporter une solution à ce problème.

L'idée centrale pour résoudre ce problème est de pouvoir extraire les concepts formels de façon duale. En effet, le treillis de concepts peut être parcouru de deux façons. La figure 2.3 montre les deux parcours du treillis de concepts de $\mathbf{r}_1$.

Nous proposons maintenant une nouvelle méthodologie d'extraction d'ensembles fermés et de concepts qui tire pleinement parti des propriétés de la connexion de Galois. L'idée est de réaliser des extractions depuis la matrice transposée tout en exploitant la connexion de Galois pour inférer les résultats qui seraient obtenus dans la matrice initiale. Si l'idée est ancienne et a déjà été utilisée dans les algorithmes de calcul de concepts formels et donc d'ensembles fermés (e.g., [21]), son application dans le contexte de l'extraction d'ensembles sous contraintes est originale.

On considère que l'on travaille avec un algorithme classique d'extraction d'ensembles fermés qui utilise la relation de spécialisation sur les attributs. Ce type d'algorithme est capable d'exploiter les contraintes anti-monotones sur les attributs et donc de calculer les théories suivantes :

$\mathcal{TH}(\mathbf{r}, (o,a) \in 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C}_{\mathbf{r}-cf}((o,a)) \wedge \mathcal{C}_s(a))$ avec $\mathcal{C}_s$ une contrainte syntaxique anti-monotone sur $\mathcal{A}$.

Nous allons montrer comment réussir à extraire la même théorie mais en travaillant sur la matrice transposée, c'est-à-dire en calculant la théorie :

$\mathcal{TH}({}^t\mathbf{r}, (a,o) \in 2^{\mathcal{A}} \times 2^{\mathcal{O}}, \mathcal{C}_{\mathbf{r}-cf}(a,o) \wedge {}^t\mathcal{C}_s(o))$.

${}^t\mathbf{r}$ représente la matrice transposée de $\mathbf{r}$ et la contrainte ${}^t\mathcal{C}_s$ sur $\mathcal{O}$ représente la contrainte transposée de $\mathcal{C}_s$ sur ${}^t\mathbf{r}$. ${}^t\mathbf{r}$ et ${}^t\mathcal{C}_s$ vont être définies par la suite.

2.2.2 Transposition de relations et de contraintes

Nous allons définir la transposition d'une relation (voir définition 2.13) et d'une contrainte (voir définitions 2.14).

Définition 2.13 (Relation transposée) *Soit* $\mathbf{r}$ *une relation sur* $\mathcal{O} \times \mathcal{A}$ *alors* ${}^t\mathbf{r}$ *sur* $\mathcal{A} \times \mathcal{O}$ *est la relation transposée de* $\mathbf{r}$ *ssi* $(a,s) \in {}^t\mathbf{r} \iff (s,a) \in \mathbf{r}$.

Si la transposition des données est naturelle, il n'en est pas de même pour une contrainte. Nous allons définir la transposition de contraintes dans le cadre de l'extraction de concepts. Par exemple, dans le cas de la contrainte de fréquence minimale, la notion duale de la fréquence pour les ensembles fermés d'attributs est celle de la taille des ensembles fermés d'objets correspondants.

Définition 2.14 (prédicat transposé) *Soit* $\mathcal{C}$ *une contrainte sur* $\mathcal{A}$, ${}^t\mathcal{C}$ *sur* $\mathcal{O}$ *est la contrainte transposée de* $\mathcal{C}$ *ssi pour tout concept formel (X,Y) :*

${}^t\mathcal{C}(X) \equiv \mathcal{C}(Y)$

Propriété 2.4 *La contrainte transposée* ${}^t\mathcal{C}$ *d'une contrainte* $\mathcal{C}$ *est :*

$^t\mathcal{C} \equiv \mathcal{C} \circ \phi$

avec ϕ l'opérateur de Galois [83].

Propriété 2.5 *Soit $\mathcal{C}$ une contrainte anti-monotone sur $\mathcal{A}$ selon une relation d'ordre sur les attributs alors $^t\mathcal{C}$ est monotone sur les objets selon cette même relation d'ordre.*

A partir de la transposition de la relation et de la contrainte, la théorie suivante peut être calculée : $\mathcal{TH}(^t\mathbf{r}, 2^{\mathcal{A}} \times 2^{\mathcal{O}}, \mathcal{C}_{\mathbf{r}-it} \wedge^t \mathcal{C})$. Par contre, la propriété 2.5 indique que la contrainte $^t\mathcal{C}$ est monotone suivant la relation de spécialisation sur les objets car la contrainte $\mathcal{C}$ est anti-monotone suivant la relation de spécialisation sur les attributs. Ainsi, avec un algorithme classique d'extraction d'ensembles fermés qui exploite les contraintes anti-monotones il n'est pas possible d'extraire la collection suivante à part en utilisant la contrainte $^t\mathcal{C}$ en post-traitement. Malgré cela, dans des contextes pathologiques c'est-à-dire des jeux de données contenant beaucoup plus de colonnes que de lignes, typiquement les données d'expression de gènes, il peut être beaucoup plus avantageux d'extraire les fermés ou les concepts sur la matrice transposée quitte à utiliser la contrainte en post-traitement. La section 2.2.3 illustre cette idée. En revanche, pour certains contextes pathologiques il n'est pas envisageable d'utiliser la contrainte en post-traitement car soit la collection extraite est trop grande (le coût du post-traitement est trop important) soit l'utilisation de la contrainte durant l'extraction est nécessaire afin de rendre l'extraction faisable. Ainsi, il est nécessaire de développer de nouveaux algorithmes capables d'exploiter des contraintes anti-monotones sur les attributs et monotones sur les objets ou en d'autres termes des contraintes monotones suivant $\subseteq$ sur les bi-ensembles. L'algorithme D-Miner présenté dans la section 2.3 répond exactement à cette attente. De façon plus générale, il est tout à fait important de pouvoir exploiter d'autres types de contraintes que ce soit sur les objets ou les attributs. La section 3.3.3 présente un cadre plus général pour l'extraction de bi-ensembles sous contraintes.

2.2.3 Validation expérimentale

Nous allons montrer que la transposition permet, pour certains jeux de données, d'améliorer sensiblement la faisabilité des extractions ou au moins de les accélérer. Les extractions ont été réalisées par F. Rioult avec `mv-miner` qui est une implémentation d'AC-Miner étendue simplement à la génération de l'ensemble support. Nous avons aussi développé notre propre extension appelé AC-like. Cet algorithme utilise les ensembles libres pour extraire les ensembles fermés. Les extractions ont été réalisées avec un seuil de fréquence absolu de 1 (au moins une condition ou un gène). Ces expériences montrent l'intérêt pour extraire les fermés ou les concepts de passer par la transposée. Le gain en temps de calcul de l'extraction de tous les ensembles libres

en passant par la matrice transposée par rapport à l'approche habituelle valide le potentiel de cette méthode pour des contextes d'extraction pathologiques.

Données biopuces utilisées

Les deux jeux de données utilisés proviennent de puces à ADN de même type (cDNA microarrays) réalisées au Stanford Genome Technology Center (Paolo Alto, CA 94306, USA).

Le premier jeu utilisé est celui présenté dans la section 4.2.1, il est composé de 5 lignes (les conditions expérimentales) et de 22069 colonnes (les gènes). Le second provient d'une expérience d'analyse du transcriptome de la drosophile durant son développement du stade de l'embryon à celui de la mouche [3]. Ce développement a été suivi sur 40 jours. Pour chaque temps, les ARN messagers ont été extraits et ont tous été comparés aux mêmes ARN messagers provenant d'un mélange de différentes mouches adultes. Il s'agissait de déterminer dans le temps les différentes séquences d'activation et de répression des gènes nécessaires au développement de la drosophile. Ces données sont disponibles sur le site du SMD[1]. Nous avons obtenu in fine une matrice d'expression composée de 162 lignes et 1 230 colonnes. A partir de ces données d'expression, nous avons procédé à une discrétisation des données afin d'obtenir un contexte booléen pour l'extraction des ensembles fréquents. Les données sont obtenues par la fonction log_2(Cy5/Cy3) et donc chaque valeur est positive (resp. négative) si l'expression d'un gène a augmenté (resp. diminué) entre les deux situations étudiées (e.g., avant et après insuline). Pour n'étudier que les fortes variations d'expression, nous avons travaillé sur les valeurs absolues des données d'expression et, pour chaque gène i, nous avons fixé un seuil noté $seuil_i$. Il permet pour chaque valeur du gène i de fixer dans la matrice booléenne la valeur 1 (resp. 0) si la valeur est supérieure (resp. inférieure) à $seuil_i$. Différentes méthodes sont utilisables pour fixer $seuil_i$ et nous avons utilisé la suivante [3] : $seuil_i = max_i \times (1 - seuil\_discr)$ où max_i est la valeur d'expression maximum du gène i dans les différentes conditions et $seuil\_discr$ est un paramètre de discrétisation commun à tous les gènes. La matrice initiale, notée M dans les figures est sous la forme situations $\times$ gènes (gènes en colonne). La matrice booléenne transposée dont les colonnes correspondent aux conditions expérimentales est notée ${}^tM$ dans les figures. Dans les expérimentations, nous avons fait varier le paramètre $seuil\_disc$ pour étudier l'évolution des extractions par rapport à la densité des matrices booléennes produites.

[1] genome-www5.stanford.edu/cgi-bin/SMD/publication/viewPublication.pl ?pub_no=183)

Données "drosophile"

La table 2.2 et l'Annexe A (figures A.1 et A.2) montrent les temps d'extraction des motifs sur les données drosophile et sur sa transposée en faisant varier *seuil_disc*. Il apparaît que le temps d'extraction est bien plus faible sur la matrice transposée que sur la matrice d'origine. Par exemple, pour $seuil\_disc = 0.11$, nous avons 1,19s pour la transposée contre 245s pour la matrice initiale. Un résultat très intéressant est que l'extraction devient faisable pour des matrices de densité plus importante. Cette méthode permet donc, non seulement de réduire le temps d'extraction mais surtout de rendre l'extraction faisable dans certains cas. L'Annexe A (figures A.1 et A.2) contient une représentation graphique des données de la table 2.2.

	seuil discr	densité	temps (ms)	nb libres	nb fermés
${}^tM$	0.02	0.008	160	965	434
M	0.02	0.008	1622	5732	434
${}^tM$	0.05	0.011	240	1975	879
M	0.05	0.011	9633	23225	879
${}^tM$	0.075	0.015	420	3667	1508
M	0.075	0.015	35390	60742	1508
${}^tM$	0.085	0.016	520	4760	1879
M	0.085	0.016	59856	89554	1879
${}^tM$	0.1	0.019	721	6890	2569
M	0.1	0.019	146861	162907	2569
${}^tM$	0.11	0.021	1191	9303	3299
M	0.11	0.021	245442	255363	3299
${}^tM$	0.15	0.032	4526	36309	10447
M	0.15	0.032	échec	-	-
${}^tM$	0.2	0.047	36722	410666	46751
M	0.2	0.047	échec	-	-
${}^tM$	0.25	0.067	455575	1330099	259938
M	0.25	0.067	échec	-	-
${}^tM$	0.3	0.09	échec	-	-
M	0.3	0.09	échec	-	-

TAB. 2.2 – Résultats des extractions sur les données "drosophile"

Données "humaines"

Les extractions sur les données humaines (voir Table 2.3 et Annexe A, figures A.3 et A.4) confirment les résultats obtenus sur les données de la drosophile (voir Table 2.2). La différence est même encore plus spectaculaire entre les extractions sur la matrice

initiale et sur sa transposée. En effet, pour la matrice transposée le temps d'extraction peut devenir négligeable par rapport à celui de la matrice initiale (e.g., 1,2s vs. 368s). De plus, le nombre de libres extraits sur la matrice initiale peut être très grand en comparaison du nombre de fermés à trouver, e.g., 51 881 libres pour seulement 34 fermés. Précisons que dans l'Annexe A, les courbes qui correspondent à l'extraction sur la transposée se confondent avec l'axe des abscisses et, de fait, ne sont pas visibles.

	seuil discr	densité	temps (ms)	nb libres	nb fermés
${}^tM$	0.01	0.193	80	19	19
M	0.01	0.193	16183	11039	19
${}^tM$	0.02	0.197	80	25	25
M	0.02	0.197	21150	13363	25
${}^tM$	0.03	0.203	90	26	26
M	0.03	0.203	27840	16725	26
${}^tM$	0.05	0.21	90	29	29
M	0.05	0.21	42991	23377	29
${}^tM$	0.075	0.223	100	32	31
M	0.075	0.223	64723	31850	31
${}^tM$	0.1	0.24	120	36	34
M	0.1	0.24	368409	51881	34
${}^tM$	0.15	0.268	120	40	38
M	0.15	0.268	échec	-	-

TAB. 2.3 – Résultats des extractions sur les données "humaines"

2.2.4 Conclusion

Les expériences précédentes ont montré l'intérêt de travailler sur les matrices transposées pour les contextes contenant beaucoup de colonnes mais peu de lignes en utilisant les contraintes en post-traitement. Par contre, lorsque la matrice contient beaucoup moins de lignes que de colonnes mais avec un nombre de lignes important, il est primordial d'être capable d'exploiter efficacement ces contraintes. Pour cela, il faut posséder des algorithmes capables de pousser des contraintes sur les deux dimensions. La transposition de matrice ne devenant alors qu'une heuristique particulière pour accélérer le calcul des ensembles fermés et des concepts formels.

2.3 D-Miner

2.3.1 Introduction

Les algorithmes d'extraction de concepts formels ou d'ensembles fermés comme Closet [75], Ac-Miner [22, 24] et Charm [101] peuvent exploiter durant l'extraction des contraintes anti-monotones sur les attributs et monotones sur les objets. Par exemple, avec ces algorithmes, on peut extraire tous les concepts formels (X, Y) tels que $\sharp(X) \geq \sigma_1 \wedge \sharp(Y) < \sigma_2$. Dans les contextes difficiles c'est-à-dire quand l'extraction de tous les concepts est impossible, ces contraintes permettent de rendre les extractions faisables. Malheureusement, les motifs qui sont alors extraits sont composés de beaucoup d'objets mais de peu d'attributs. Or, dans beaucoup de domaines d'application, les utilisateurs finaux souhaitent obtenir des motifs composés d'un nombre minimal d'objets et d'attributs. Par exemple, pour les données d'expression de gènes, les biologistes souhaitent des groupes de synexpression composés d'un certain nombre de conditions expérimentales mais aussi d'un certain nombre de gènes. Ainsi, en utilisant les algorithmes d'extraction de concepts traditionnels, on ne pourra extraire qu'un sous-ensemble des associations que souhaitent vraiment les utilisateurs finaux. Effectivement, il faut fixer une contrainte de taille minimale sur les objets (fréquence minimale) puis post-traiter la collection finale afin de ne conserver que les motifs ayant un nombre minimal d'attributs. De plus, la section précédente a montré que si l'on veut pouvoir extraire les concepts formels "fréquents" sur une matrice booléenne et sur sa transposée, il faut alors être capable d'exploiter la contrainte de taille minimale à la fois sur les objets et sur les attributs.

Ainsi, nous avons proposé un nouvel algorithme d'extraction de concepts formels appelé D-Miner [15, 12] qui permet d'exploiter, durant l'extraction, des contraintes monotones sur les deux dimensions. Il est de plus très efficace dans les types de contextes qui nous intéressent c'est-à-dire les contextes relativement petits mais très denses.

D-Miner permet d'extraire les théories suivantes : $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{O}} \times 2^{\mathcal{A}}, \mathcal{C})$ avec $\mathcal{C}$ une contrainte monotone sur $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$.

Parmi ces contraintes monotones, certaines nous intéressent particulièrement. La première permet de fixer une taille minimale sur les deux dimensions des concepts. En effet, nous allons pouvoir nous intéresser seulement aux motifs contenant au moins un certain nombre de lignes et un certain nombre de colonnes. La deuxième contrainte intéressante permet d'imposer que l'aire des concepts extraits (X, Y) soit supérieure à une valeur donnée, par exemple $|X| \times |Y| > 20$. Cette dernière a aussi la particularité d'utiliser conjointement les deux dimensions $\mathcal{O}$ et $\mathcal{A}$ c'est-à-dire qu'elle ne peut être décomposée sous la forme de deux contraintes l'une sur $\mathcal{O}$ et l'autre sur $\mathcal{A}$. Enfin, on peut aussi imposer la présence de certains éléments dans les concepts extraits.

2.3.2 Principe de D-Miner

Un concept (T, G) est un bi-ensemble dont tous les objets et tous les attributs sont en relation dans $\mathbf{r}$. Ainsi, l'absence de relation entre un objet o et un attribut a induit la présence dans $\mathbf{r}$ de deux concepts, un contenant o mais pas a et un autre contenant a mais pas o. D-Miner est basé sur cette observation. Nous allons utiliser une autre définition des concepts formels pour expliquer le fonctionnement de D-Miner .

Définition 2.15 (Concept formel) *Un bi-ensemble (X, Y) est un concept dans $\mathbf{r}$ ssi :*

1. $\mathcal{C}_{\mathbf{r}-1-r}((X, Y))$
2. $\forall x \in \mathcal{O} \setminus X \; \exists y \in y$ *tel que* $(x, y) \notin \mathbf{r}$
3. $\forall y \in \mathcal{A} \setminus Y \; \exists x \in X$ *tel que* $(x, y) \notin \mathbf{r}$

La contrainte (1) permet d'imposer que les concepts formels sont des 1-rectangles. Les contraintes (2) et (3) sont des contraintes de "maximalité". Un élément n'appartient pas à un concept si et seulement s'il introduirait un "0" dans le motif s'il en faisait partie.

D-Miner énumère les bi-ensembles en débutant avec $(\mathcal{O}, \mathcal{A})$ et utilise la relation d'ordre $\subseteq$ pour générer les candidats suivants. C'est une énumération binaire sur les objets qui est utilisée. Pour chaque candidat (X, Y), un objet o de X est choisi et deux nouveaux candidats sont générés l'un contenant o et l'autre pas. Pour chaque candidat ainsi généré, il faut ensuite appliquer les propriétés de la définition 2.15.

Définition 2.16 *Nous notons $H \in \mathcal{O} * 2^{\mathcal{A}}$ l'ensemble tel que :*

$\{(x, Y) \mid x \in \mathcal{O}$ *et* $\forall y \in Y (x, y) \notin \mathbf{r}$ *et* $\forall y \in \mathcal{A} \setminus Y (x, y) \in \mathbf{r}\}$

H représente les vecteurs de "0" pris ligne par ligne dans $\mathbf{r}$.

D-Miner débute avec le couple $(\mathcal{O}, \mathcal{A})$ et ensuite le découpe récursivement en utilisant tous les éléments de H. Un élément (a, b) de H est utilisé pour découper un bi-ensemble (X, Y) si $\{a\} \cap X \neq \emptyset$ et $\{b\} \cap Y \neq \emptyset$. Si c'est le cas, deux bi-ensembles sont obtenus $(X \setminus \{a\}, Y)$ et $(X, Y \setminus \{b\})$. Par convention le premier est appelé le fils gauche et le second le fils droit. La figure 2.4 illustre le découpage. Tous les descendants du fils droit possèdent l'objet a car chaque élément de H est utilisé au plus une fois. D'après (1) il faut aussi que tous les descendants du fils droit ne contiennent aucun des attributs qui ne sont pas en relation avec l'objet a. Le fils droit est ainsi $(X, Y \setminus b)$. De façon récursive, la propriété (1) de la définition 2.15 est vérifiée pour les feuilles de l'arbre d'énumération. Tous les descendants du

fils droit ne contiennent pas l'objet a. D'après (2), tous les descendants du fils droit doivent posséder au moins un attribut qui n'est pas en relation avec a. Il faut ainsi vérifier que les attributs du fils droit ont une intersection non vide avec tous les éléments de H utilisés pour générer ses ascendants qui sont des fils gauches. Si ce n'est pas le cas, il est élagué. De façon récursive, la propriété (2) de la définition 2.15 est satisfaite pour les feuilles de l'arbre d'énumération. La propriété (3) de la définition 2.15 est quant à elle toujours vérifiée. En effet, l'énumération étant réalisée sur les objets, un attribut est absent d'un bi-ensemble si et seulement si un objet qui n'est pas en relation avec lui a déjà été mis dans le bi-ensemble. Finalement, les bi-ensembles qui sont les feuilles de l'arbre d'exécution sont les concepts formels de $\mathbf{r}$. L'exemple 2.4 montre une premier exemple d'exécution de D-MINER et l'exemple 2.5 est un deuxième exemple qui montre comment la propriété (2) est satisfaite.

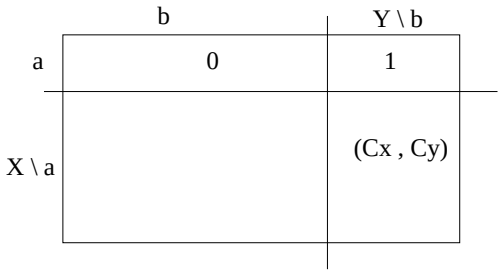


FIG. 2.4 – Découpage d'un bi-ensemble (X, Y) candidat

Exemple 2.4 *Dans cet exemple, nous allons utiliser une relation* $\mathbf{r}_2$ *présentée dans la table 2.4. Dans cet exemple,* H *est égal à* $\{(o_1, \{a_1\}), (o_3, \{a_2\})\}$*. L'arbre d'exécution de* D-MINER *pour* $\mathbf{r}_2$ *est représenté dans la figure 2.5. Les bi-ensembles représentent les bi-ensembles candidats et les boîtes représentent les éléments de* H *utilisés pour générer les nouveaux candidats. Finalement, les quatre concepts de* $\mathbf{r}_2$ *sont obtenus :*

$$\{(\{o_2\}, \{a_1a_2a_3\}), (\{o_2o_3\}, \{a_1a_3\}), (\{o_1o_2\}, \{a_2a_3\}), (\{o_1o_2o_3\}, \{a_3\})\}$$

	a_1	a_2	a_3
o_1	0	1	1
o_2	1	1	1
o_3	1	0	1

TAB. 2.4 – Contexte $\mathbf{r}_2$ pour l'exemple 2.4

Exemple 2.5 *Soit maintenant la relation* $\mathbf{r}_3$ *représentée dans la table 2.5. L'ensemble* H *est égal à* $\{(o_1, \{a_1a_2\}),\ (o_2, \{a_2\}),\ (o_3, \{a_1a_2\})\}$*. La Figure 2.6 illustre*

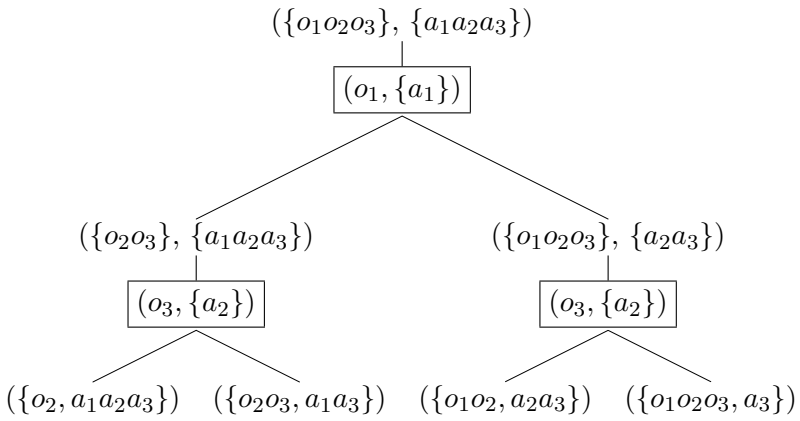


FIG. 2.5 – Arbre d'énumération pour $\mathbf{r}_2$ (Exemple 2.4)

l'arbre d'énumération de D-MINER *pour* $\mathbf{r}_3$*. Deux bi-ensembles sont soulignés. Ces deux bi-ensembles ne sont pas des concepts car* $(\{o_3\}, \{a_3\}) \subseteq (\{o_1o_2o_3\}, \{a_3\})$ *et* $(\{o_2o_3\}, \{a_3\}) \subseteq (\{o_1o_2o_3\}, \{a_3\})$*. Ceux sont des 1-rectangles qui ne sont pas maximaux. Effectivement, pour le bi-ensemble* $(\{o_3\}, \{a_3\})$*, il faut regarder les éléments de H qui ont été utilisés pour générer ses ascendants qui sont des fils gauches. Il a trois ascendants* $(\{o_1o_2o_3\}, \{a_1a_2a_3\})$*,* $(\{o_2o_3\}, \{a_1a_2a_3\})$ *et* $(\{o_3\}, \{a_1a_2a_3\})$*. Parmi ceux la, seuls les deux derniers sont des fils gauches et les éléments de H utilisés pour les obtenir sont respectivement* $(o_1, \{a_1a_2\})$ *et* $(o_2, \{a_2\})$*. Or, l'ensemble des attributs du bi-ensemble* $(\{o_3\}, \{a_3\})$ *a une intersection vide avec l'ensemble des attributs des éléments de H pré-cités. Ainsi, ce bi-ensemble ne satisfait pas la propriété (2) de 2.15 et donc est élagué. Pour* $(\{o_2o_3\}, \{a_3\})$*, il faut utiliser* $(o_1, \{a_1a_2\})$ *de H pour vérifier la propriété (2). De la même façon, ce bi-ensemble doit être élagué. Finalement, la collection des concepts de la table 2.5 est :*

$$\{(\{o_1o_2o_3\}, \{a_3\}),\ (\{o_2\}, \{a_1a_3\}),\ (\emptyset, \{a_1a_2a_3\})\}$$

	a_1	a_2	a_3
o_1	0	0	1
o_2	1	0	1
o_3	0	0	1

TAB. 2.5 – Contexte $\mathbf{r}_3$ pour l'exemple 2.5

Concernant la propagation des contraintes, dans la mesure où les candidats sont générés en utilisant la relation d'ordre $\subseteq$ alors toutes les contraintes monotones sur $\subseteq$ peuvent être exploitées activement. Il suffit pour cela de vérifier la contrainte sur

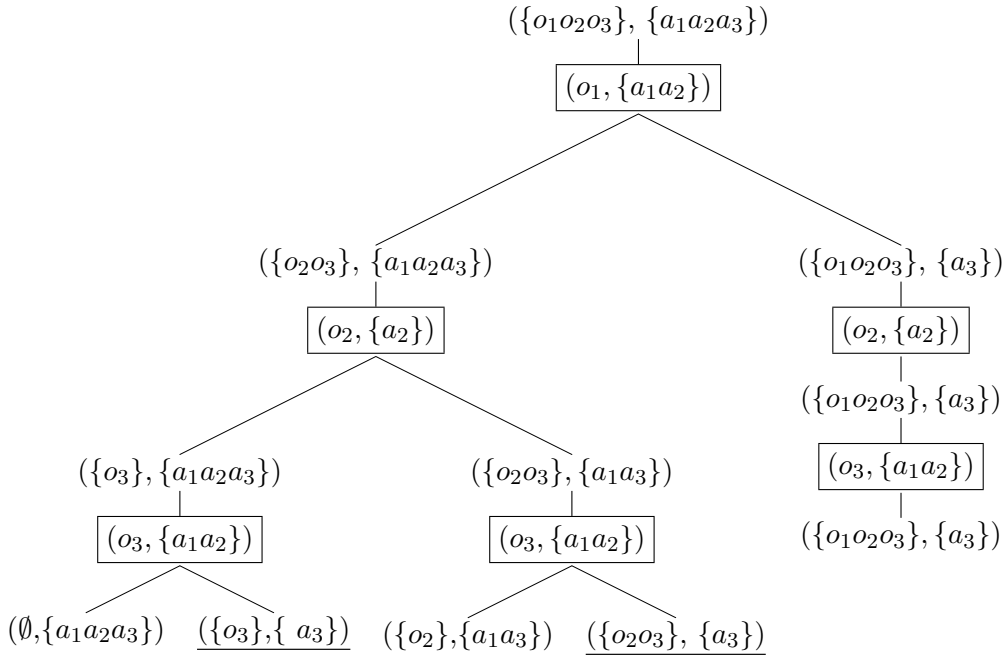


FIG. 2.6 – Arbre d'énumération pour $\mathbf{r}_3$ (Exemple 2.5)

les bi-ensembles candidats et si elle n'est pas vérifiée alors le candidat est élagué, c'est-à-dire qu'il ne doit pas être utilisé pour générer de nouveaux candidats.

L'annexe B contient les preuves de consistance et de complétude de D-MINER.

2.3.3 Algorithme

Avant de découper un bi-ensemble (X, Y), deux types de contraintes doivent être vérifiées, d'abord la contrainte monotone puis la contrainte relative à la propriété (2) de la définition 2.15. Pour cela, un ensemble H_L contenant les éléments de H utilisés pour générer les fils gauches précédents est utilisé.

D-MINER est un algorithme qui énumère les candidats en profondeur. Les algorithmes 1 et 2 contiennent le pseudo-code de D-MINER . D'abord, l'ensemble H est calculé. Ensuite la fonction récursive *decoupage()* est appelée.

La fonction *decoupage* coupe le bi-ensemble (X, Y) avec l'élément $(a, b) = H[i]$ s'il satisfait la contrainte suivante : (X, Y) doit avoir une intersection non vide avec $H[i]$ ($a \cap X \neq \emptyset \wedge b \cap Y \neq \emptyset$). Si ce n'est pas le cas, la fonction *decoupage* est rappelée avec le prochain $H(i+1)$. Avant de découper (X, Y) en $(X \backslash a, Y)$, il faut vérifier la contrainte monotone. Si c'est le cas, la fonction *decoupage* est appelée sur $(X \backslash a, Y)$

Algorithme 1 : D-MINER

Input : Une relation **r** avec n objets et m attributs,
$\mathcal{O}$ l'ensemble des objets, $\mathcal{A}$ l'ensemble des attributs et
$\mathcal{C}_m$ une contrainte monotone sur $2^{\mathcal{O}} \times 2^{\mathcal{A}}$.
Output : $\mathcal{Q}$ l'ensemble des concepts qui satisfont $\mathcal{C}_m$
$H_L \leftarrow \emptyset$
H est calculé à partir de **r**
If $(\mathcal{C}_m((X, Y)))$
  $\mathcal{Q} \leftarrow$ decoupage$((\mathcal{O}, \mathcal{A}), H, 0, H_L, \mathcal{C}_m)$

Algorithme 2 : decoupage

Input : (X, Y) un bi-ensemble de $2^{\mathcal{O}} \times 2^{\mathcal{A}}$, l'ensemble H,
i le niveau de profondeur de l'énumération,
l'ensemble H_L, $\mathcal{C}_m$ une contrainte monotone sur $2^{\mathcal{O}} \times 2^{\mathcal{A}}$.
Output : $\mathcal{Q}$ l'ensemble des concepts qui satisfont $\mathcal{C}_m$
$(a, b) \leftarrow$ H[i]
If $(i \leq |H| - 1)$
  If $((a \cap X = \emptyset)$ or $(b \cap Y = \emptyset))$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup decoupage((X, Y), H, i + 1, H_L, \mathcal{C}_m)$
  Else
    If $\mathcal{C}_m((X \setminus a, Y))$ est vérifiée
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup\ decoupage((X \backslash a, Y), H, i + 1, H_L \cup (a, b), \mathcal{C}_m)$
    If $\mathcal{C}_m((X, Y \backslash b))$ est vérifiée $\wedge\ \forall (a', b') \in H_L,\ b' \cap Y \backslash b \neq \emptyset$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup decoupage((X, Y \backslash b), H, i + 1, H_L, \mathcal{C}_m)$
Else
  $\mathcal{Q} \leftarrow (X, Y)$
Return $\mathcal{Q}$

---

en ajoutant (a, b) à H_L. Pour le second découpage de (X, Y) c'est-à-dire $(X, Y \backslash b)$, deux vérifications de contraintes sont nécessaires. D'abord, la contrainte monotone doit être vérifiée et ensuite il faut utiliser l'ensemble H_L pour supprimer les motifs redondants comme expliqué précédemment.

Trois points permettent d'optimiser l'algorithme. D'abord, l'ordre des éléments de H est crucial. L'objectif est de couper le plus rapidement possible les branches qui gênèrent des 1-rectangles non-maximaux. Ainsi, les éléments (a, b) de H sont triés par ordre décroissant de la taille de b. De plus, pour réduire la taille de H, les éléments de

H qui ont le même ensemble d'attributs sont fusionnés. Finalement, pour les raisons évoquées dans la section 2.1.1 si $|\mathcal{O}| > |\mathcal{A}|$, l'extraction est effectuée sur la matrice transposée.

Nous avons utilisé, pour l'implémentation de D-MINER, des vecteurs de bits pour représenter les ensembles de lignes et de colonnes, un 1 représente la présence d'un élément dans l'ensemble et un 0 l'absence. En effet, l'opérateur principal dont nous avons besoin est l'intersection entre deux ensembles. Avec des vecteur de bits, cette intersection correspond à un "et bit à bit" qui est très rapide. De plus, comme 32 éléments (lignes et/ou colonnes) sont représentés par un seul entier, l'espace mémoire nécessaire pour représenter les ensembles est réduit et à chaque "et bit à bit" 32 tests sont réalisés simultanément. Avec cette structure de données, seul le comptage de la taille des ensembles est un peu coûteux.

2.3.4 Évaluation du délai

Nous désignons dans la suite par n le nombre de lignes et par m le nombre de colonnes du tableau de données.

Kuznetsov [58] a montré que l'extraction des concepts formels est un problème NP-complet.

La taille de la collection à extraire est potentiellement exponentielle dans la plus petite dimension du jeu de données. Pour comparer la complexité des algorithmes, il peut alors être intéressant de s'abstraire de la taille de la collection. C'est ce que permet de faire le délai :

Définition 2.17 (Délai) *Un algorithme qui énumère des structures combinatoires a un délai d s'il effectue au plus d opérations élémentaires entre deux solutions générées [55].*

Dans la suite, nous allons évaluer le délai de D-MINER dans le pire des cas et en moyenne, et nous donnerons quelques éléments de comparaison avec les algorithmes de référence.

Délai dans le pire cas

Nous allons montrer que D-MINER n'explore pas de chemin de longueur supérieure à 1 qui ne mène pas à un concept formel.

Propriété 2.6 *Soit (X, Y) un nœud de l'arbre alors s'il satisfait la partie (2) de la propriété 2.15 (i.e. l'intersection avec tous les éléments de la liste H_L est non vide)*

alors (X, Y) contient au moins un concept. Ainsi, si un candidat ne contient pas de concepts alors il ne satisfait pas la propriété 2.15(2).

Preuve. Si (X, Y) satisfait la propriété 2.15(2) alors son fils gauche est généré, il satisfait aussi la propriété. Récursivement (X, Y) a un descendant qui est une feuille et qui est le descendant des fils gauches successifs.

Dans le pire cas, le chemin le plus long (nombre d'arêtes dans l'arbre) entre deux concepts consécutifs est de longueur $2n$. Une fois le premier concept généré, D-MINER va suivre ce chemin et pour chaque nœud de ce chemin il va tester au plus un candidat supplémentaire (d'après la propriété précédente). Ainsi au plus $4n$ candidats seront générés. Dans le pire cas il faut tester pour chaque candidat la propriété 2.15 qui est en $O(nm)$. Ainsi, le délai entre la génération de deux concepts est en $O(n^2m)$.

L'algorithme de Chein a un délai polynomial en $O(n^3m)$, alors que ceux de Bordat et Ganter ont un délai polynomial identique à celui de D-MINER [59].

Délai en moyenne

Nous voulons maintenant évaluer le délai de D-MINER en considérant l'écart moyen entre les concepts formels. L'écart entre deux concepts correspond au nombre de feuilles de l'arbre d'énumération complet entre ces deux concepts. Comme D-MINER réalise une énumération sur les objets, la profondeur de l'arbre d'énumération est n et donc son nombre de feuille est de 2^n. Nous allons considérer les collections de concepts formels de taille K. Ainsi, l'écart moyen d entre deux concepts est de : $d = \frac{2^n}{K}$. Nous allons maintenant calculer le délai moyen pour des concepts formels distants de d.

Nous allons considérer que K est une puissance de 2. Soient n_1 et n_2 deux concepts formels distants de d, nous appelons nc le plus petit ancêtre commun de n_1 et n_2. Ce nœud est à une hauteur $h = \lfloor log_2(d) \rfloor + 1$ ($\lfloor a \rfloor$ est l'entier arrondi inférieurement de a). Les deux fils de nc définissent deux sous-arbres. Comme K est une puissance de 2, chaque sous-arbre contient d feuilles, alors si l'on considère que n_1 est la x^{ieme} feuille du premier sous-arbre alors n_2 est aussi la x^{ieme} feuille du second sous-arbre. La figure 2.7 résume ces informations.

Ainsi, pour calculer le délai moyen pour des concepts formels distants de d, il suffit de le calculer sur ces sous-arbres. Le coût de l'évaluation d'un fils gauche est en $O(m)$ et d'un fils droit est en $O(mn)$. Il faut calculer pour les chemins entre deux concepts distants de d le nombre de fils gauche et de fils droit considérés et ensuite calculer leur délai. La figure 2.8 montre un exemple de l'énumération réalisée par D-MINER entre deux concepts n_1 et n_2 et leur coût associé. Sur ce chemin, le délai est de $4 * O(mn) + O(m)$.

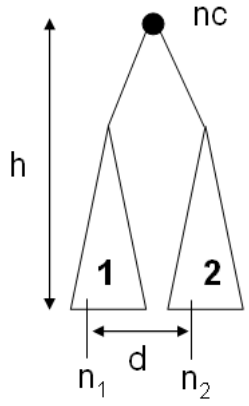


FIG. 2.7 – Résumé des notations pour le délai

Il faut d'abord calculer la moyenne des fils droits considérés sur le sous-arbre gauche de *nc* pour tous les chemins allant des feuilles de ce sous-arbre à *nc* (cas 1 de la figure 2.7).

Propriété 2.7 *Le nombre de chemins qui passent par le fils droit d'un nœud est égal au nombre de feuilles associées au fils gauche.*

Effectivement, seuls les chemins arrivant du fils gauche du nœud passent par le fils droit. Or, le nombre de ces chemins est égal au nombre de feuilles du fils gauche.

Finalement, pour atteindre un nœud de hauteur h, tous les chemins partant des feuilles du sous-arbre gauche considèrent D fils droits avec

$$D = \sum_{i=1}^{h} 2^{i-1}\, 2^{h-i}$$

En effet, il suffit de calculer pour chaque niveau i, le nombre de nœuds de ce niveau (2^{h-i}) et pour chacun de ces nœuds le nombre de fois où son fils droit a été considéré, i.e. le nombre de feuilles associées à son fils gauche (2^{i-1}). En simplifiant, on obtient finalement

$$D = \sum_{i=1}^{h} 2^{i-1}\, 2^{h-i} = \sum_{i=1}^{h} 2^{h-1} = h\, 2^{h-1}$$

En moyenne, un chemin partant d'une feuille du sous-arbre gauche (contenant $2^h - 1$ feuilles) et allant à *nc*, considère h nœuds droits en moyenne :

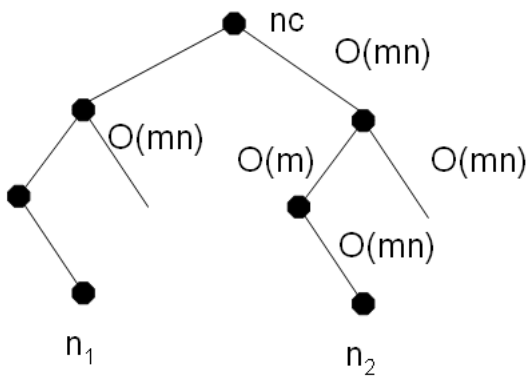


FIG. 2.8 – Exemple d'un chemin et du calcul du délai

$$\frac{D}{2^{h-1}} = \frac{h\,2^{h-1}}{2^{h-1}} = h$$

Il faut noter que sur le sous-arbre gauche aucun chemin n'évalue de fils gauche, effectivement on effectue seulement des remontées récursives.

Il faut ensuite calculer la moyenne des fils droit et des fils gauche considérés sur le sous-arbre droit de nc pour tous les chemins allant de nc aux feuilles de ce sous-arbre (cas 2 de la figure 2.7). En moyenne, les chemins vont considérer $\frac{(h-1)}{2} + 1$ fils droit et $\frac{(h-1)}{2}$ fils gauche.

Finalement, le délai moyen pour deux concepts distants de d est de :

$$\begin{aligned} delais &= \left(\frac{h-1}{2} + 1 + h\right) O(nm) + \frac{h-1}{2} O(m) \\ &= \left(\frac{\log_2(d)}{2} + 1 + \log_2(d) + 1\right) O(nm) + \frac{\log_2(d)}{2} O(m) \\ &\approx (\log_2(d) + 1)\, O(nm) \\ &\approx (n - \log_2(K) + 1)\, O(mn) \end{aligned}$$

Si l'on considère un jeu de données contenant les deux concepts triviaux ($\bot$ et $\top$), on retrouve le délai dans le pire des cas : $O(mn^2)$. Lorsque l'on considère un jeu de données contenant 2^n concepts (nombre maximum) le délai est alors en $O(mn)$. On peut remarquer que dans des jeux de données très denses (contenant beaucoup de concepts) le délai est alors très intéressant.

2.3.5 Validation expérimentale

Nous avons comparé le temps d'exécution de D-Miner avec ceux de Closet [75], Ac-Miner [22, 24] et Charm [101]. Closet, Ac-Miner et Charm sont des algorithmes efficaces qui calculent les ensembles fermés. Ces algorithmes sont basés sur différentes techniques. Closet et Charm réalisent une recherche en profondeur des itemsets fermés. Ac-Miner calcule par niveau les itemsets libres et leurs fermetures, i.e., les itemsets fermés. Les algorithmes qui calculent les itemsets fermés peuvent calculer les concepts formels en ajoutant simplement le support de l'ensemble fermé. Nous avons réalisé des expérimentations sur des jeux de données artificielles, sur des jeux de données de l'UCI repository[2] et sur des données d'expression de gènes. Comme expliqué précédemment, il est facile d'obtenir les concepts à partir des ensembles fermés et dans ces expérimentations, nous comparons directement le temps d'exécution des algorithmes Closet, Charm et Ac-Miner sans considérer le temps de génération des concepts formels à partir des ensembles fermés.

Nous avons utilisé l'implémentation de Zaki de Charm [101] et celles de Bykowski d'Ac-Miner [22, 24] et de Closet [75]. Pour réaliser une comparaison juste, pour chaque expérience nous avons transposé les matrices de telle sorte à avoir le plus petit nombre de colonnes pour Closet, Ac-Miner et Charm et le plus petit nombre de lignes pour D-Miner. En effet, la complexité en temps dépend principalement du nombre de colonnes pour Closet, Ac-Miner et Charm et du nombre de lignes pour D-Miner. Le seuil de fréquence minimale est donné en relatif par rapport au nombre de lignes. Toutes les extractions ont été réalisées sur un Pentium III (450 MHz, 128 Mb).

Données artificielles et benchmarks

Dans l'Annexe C, nous présentons le temps d'exécution des 4 algorithmes sur plusieurs jeux de données artificielles générés avec le générateur IBM[3]. Nous avons généré 60 jeux de données denses avec différents nombres de colonnes (300, 700 et 900 colonnes), de lignes (100 et 300 lignes) et de densité de valeurs "1" (15 % et 35 %). Pour chaque combinaison de paramètres, nous avons généré 5 jeux de données. La table C.1 de l'annexe C fournit la moyenne et l'écart-type du temps d'exécution (en secondes) pour chaque algorithme pour les jeux de données avec une densité de 15%. Nous pouvons remarquer que D-Miner réussit à extraire les ensembles fermés (concepts formels) alors que certains autres algorithmes échouent. La table C.2 de l'Annexe C donne la moyenne et l'écart-type du temps d'exécution (en secondes) pour chaque algorithme sur les jeux de données avec une densité de 35%. Sur ces jeux de données, l'efficacité de D-Miner par rapport aux autres algorithmes est encore plus

[2]http ://www.ics.uci.edu/ mlearn/databases/
[3]http ://miles.cnuce.cnr.it/ palmeri/datam/DCI/datasets.php

visible.

Dans la figure 2.9, nous montrons le temps d'extraction des motifs sur le benchmark "Mushroom". Ce jeu de données contient 8 124 lignes et 120 colonnes.

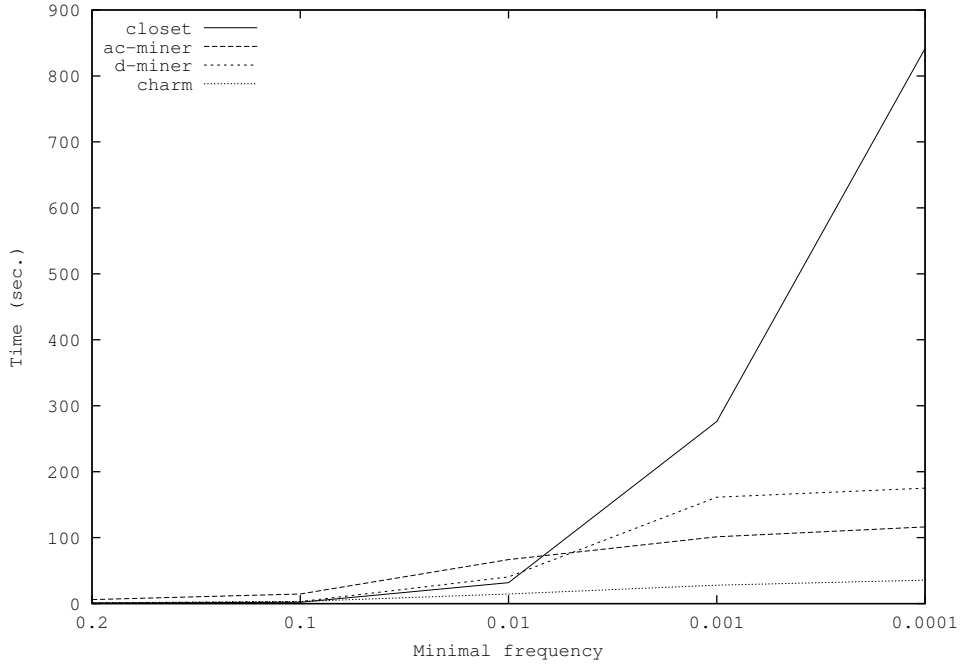


FIG. 2.9 – Extraction de concepts sur Mushroom

Sur "Mushroom", les quatre algorithmes réussissent à extraire tous les motifs (avec une fréquence minimale de 0.0001) en quelques minutes. La fréquence minimale correspond aux motifs contenant au moins 1 objet. Le temps d'exécution de CLOSET augmente beaucoup plus rapidement en comparaison des trois autres.

Ensuite, nous considérons le benchmark "Connect4". Ce jeu de données contient 67 557 lignes et 149 colonnes. Le temps d'extraction des motifs est montré dans la figure 2.10. Seuls CHARM et D-MINER réussissent à les extraire avec une fréquence minimale de 0.1. D-MINER est à peu près deux fois plus rapide que CHARM sur ce jeu de données. Il faut noter aussi que l'extraction de tous les motifs (i.e. avec la fréquence minimale la plus petite) reste infaisable sur "Connect4".

Une application aux données d'expression de gènes

Pour cette expérience, nous avons utilisé le jeu de données présenté dans la section 4.3. Les lignes représentent des facteurs de transcription et les colonnes des gènes.

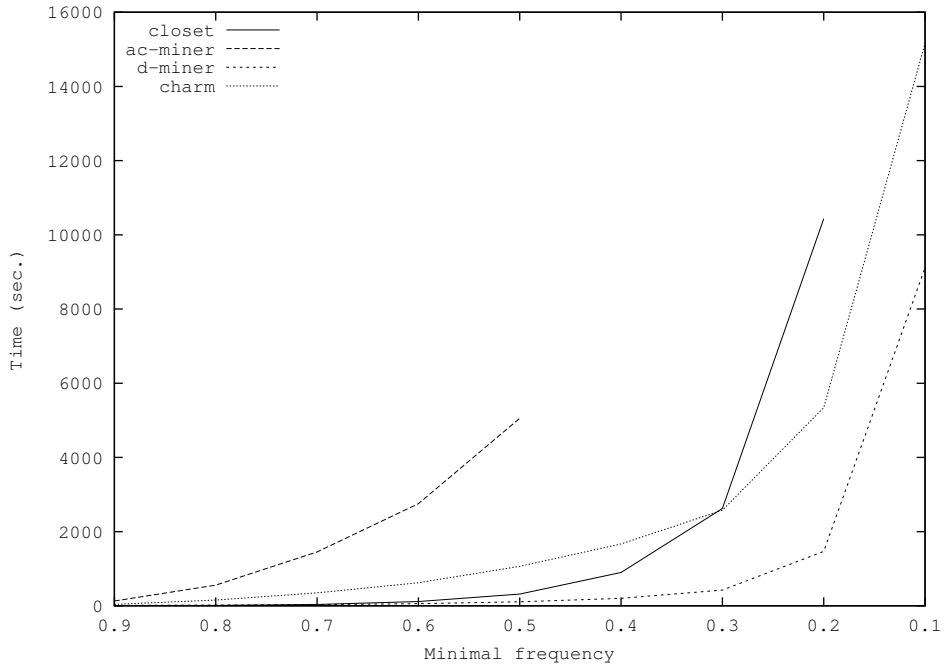


FIG. 2.10 – Extraction de concepts sur Connect4

Un "1" dans la matrice signifie qu'un facteur de transcription peut potentiellement s'accrocher sur la région promotrice du gène et ainsi le réguler. Cette matrice contient 155 lignes et 356 colonnes. L'extraction de concepts formels dans ce type de contextes booléens correspond à la Requête 2 de la section 1.5.1.

Le temps d'extraction des motifs en fonction de la fréquence minimale (sur les gènes) sur ce jeu de données biologiques pour CLOSET, AC-MINER, CHARM et D-MINER est présenté dans la figure 2.11.

D-MINER est le seul algorithme qui réussit à extraire tous les concepts formels. Ce jeu de données est particulier : il y a peu de concepts formels avant le seuil de fréquence de 0.1 sur les gènes (5 534 concepts) et ensuite le nombre de concepts augmente très rapidement. Ce jeu de données contient en fait plus de 5 millions de concepts formels. Dans ce contexte, il est nécessaire d'extraire les concepts formels avec un seuil de fréquence minimale relativement petit, sinon presque aucun concept n'est extrait. Ainsi, D-MINER est plus pertinent que les autres algorithmes pour ce type de jeux de données car il réussit à extraire les concepts formels quelle que soit la fréquence minimale.

Nous allons maintenant utiliser un jeu de données d'expression de gènes qui contient 5 conditions expérimentales (les lignes) et 1065 gènes (les colonnes). Nous

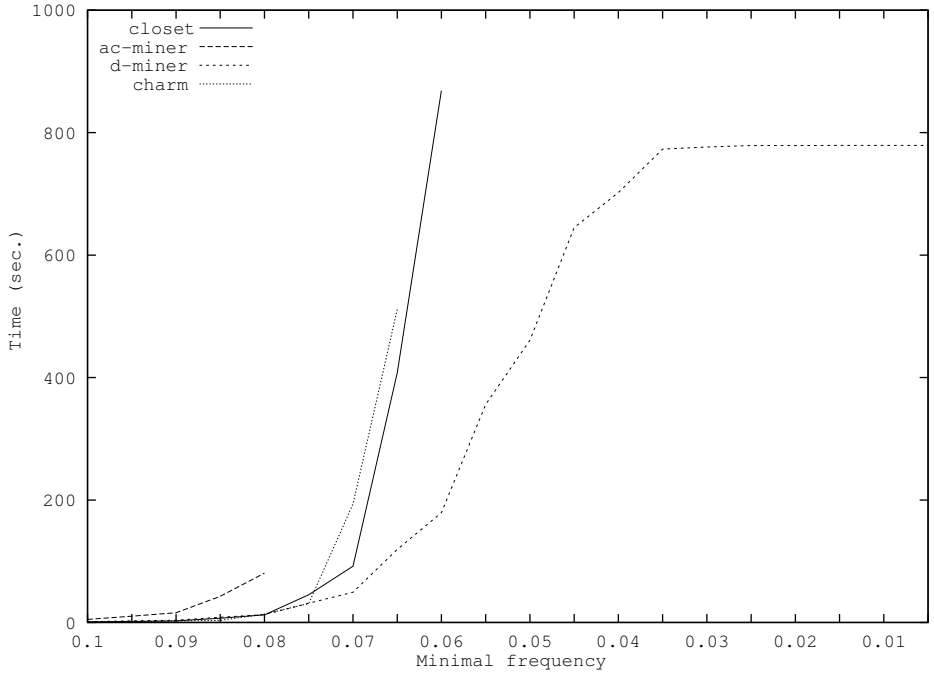


FIG. 2.11 – Jeu de données biologiques

souhaitons extraire les concepts qui contiennent au moins 4 parmi les 5 conditions expérimentales et au moins γ gènes. Cette extraction est similaire à la Requête 1 de la section 1.5.1. La figure 2.12 présente le nombre de concepts formels qui satisfont $\mathcal{C}_{4\gamma-mis}$ lorsque γ varie.

En utilisant des contraintes qui imposent une taille minimale pour l'ensemble des lignes et des colonnes, le nombre de motifs extraits peut diminuer de façon importante tout en augmentant la pertinence des motifs extraits pour l'utilisateur final. De plus, dans d'autres jeux de données, l'utilisation simultanée de ces deux contraintes peut rendre l'extraction faisable alors que l'extraction utilisant une seule des deux contraintes et un post-traitement pour la seconde est infaisable. Pour illustrer ce point, nous allons utiliser le jeu de données contenant 155 lignes et 356 colonnes présenté précédemment. Nous avons utilisé la contrainte $\mathcal{C}_{\gamma_1\gamma_2-mis}$ qui impose une taille minimale γ_1 pour l'ensemble des objets et γ_2 pour les attributs. La figure 2.13 indique le nombre de concepts extraits en fonction de γ_1 et γ_2.

Il apparaît qu'en utilisant seulement une des deux contraintes, le nombre de concepts ne diminue pas de façon importante (voir les valeurs avec $\gamma_1 = 0$ or $\gamma_2 = 0$). Alors qu'en utilisant simultanément les deux contraintes, la taille de la collection diminue énormément (la surface de la fonction forme un bassin). Par exemple, le nombre de concepts extraits avec la contrainte $\mathcal{C}_{21\,10\,-mis}$ est de 142 279. Alors que

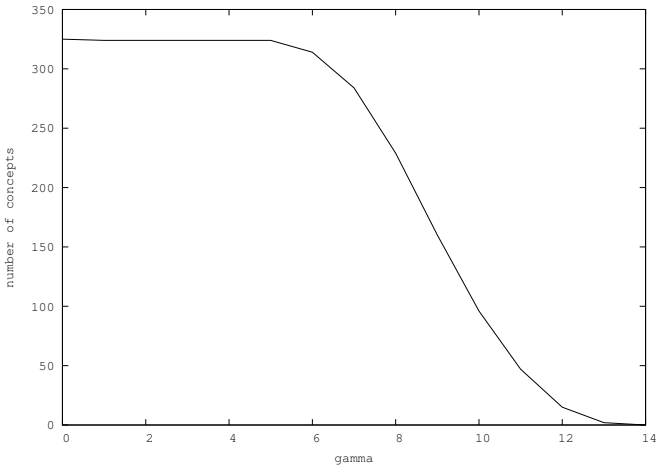


FIG. 2.12 – Nombre de concepts en fonction de γ

si l'on utilise la contrainte $\mathcal{C}_{0\,10\,-mis}$ 5 422 514 concepts sont extraits. En utilisant $\mathcal{C}_{21\,0\,-mis}$ 208 746 concepts sont obtenus.

2.3.6 Conclusion

Même si l'extraction de concepts formels est étudiée depuis une vingtaine d'années, elle reste difficile dans la plupart des applications réelles qui nous intéressent. Outre la combinatoire de cette approche, le problème de la pertinence des motifs est toujours ouvert. Surtout que la pertinence des motifs peut être associée à différents points de vue : pertinence par rapport au jeu de données, par rapport au modèle des données ou par rapport à la problématique étudiée par l'utilisateur. Dans les données booléennes d'expression de gènes, on peut extraire des milliers de motifs (concepts formels) dont beaucoup d'entres eux ne sont pas intéressants car, par exemple n'associant que quelques conditions expérimentales à des milliers de gènes. Il faut alors offrir aux utilisateurs un moyen d'exprimer précisément le type de motifs qui leur permettrait de répondre à leurs questions. Il est alors indispensable de travailler sur l'extraction sous contraintes. Les contraintes permettent alors d'augmenter la pertinence des motifs extraits. Ainsi, nous avons proposé un algorithme d'extraction de concepts formels sous contraintes appelé D-MINER qui exploite efficacement les contraintes monotones sur $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$. En particulier, il est capable d'extraire tous les concepts qui ont au moins un certain nombre de lignes et un certain nombre de colonnes, une aire minimale et qui contiennent tels ou tels éléments. Ces contraintes permettent à la fois de rendre les extractions faisables mais surtout de pouvoir améliorer considérablement la pertinence des motifs extraits par rapport aux attentes de l'utilisateur. La section 4 montre clairement l'intérêt d'exploiter ce type de contraintes alors même que l'ana-

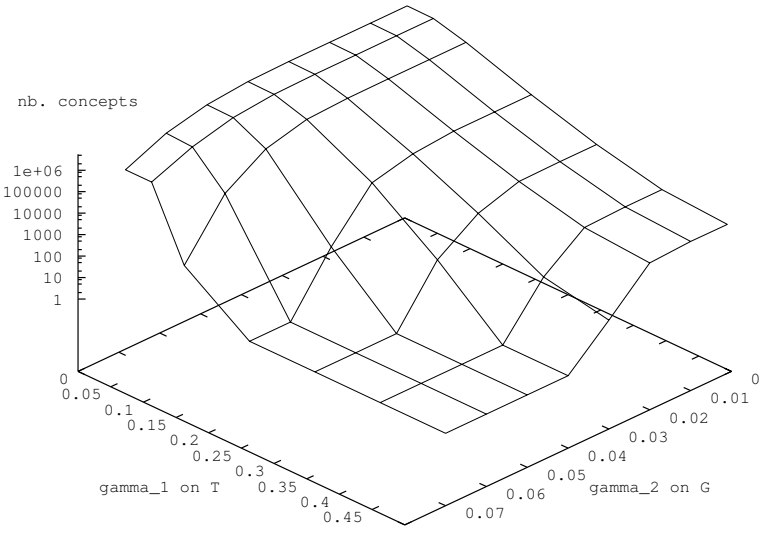


FIG. 2.13 – Nombre de concepts en fonction de γ_1 et γ_2

lyse de la collection complète (>5 millions de motifs) n'est pas envisageable. Même si D-MINER apporte des solutions pour l'extraction de concepts sous contraintes, il faut garder en tête qu'un des objectifs finaux est de réussir à proposer un extracteur générique de bi-ensembles sous contraintes qui est une brique des bases de données inductives.

Chapitre 3

Concepts formels avec exceptions

3.1 Introduction au problème des données bruitées

3.1.1 Introduction

Dans la section 2.1, nous avons discuté et présenté des travaux sur la recherche de concepts formels. Ce sont des 1-rectangles maximaux c'est-à-dire des ensembles maximaux de lignes et de colonnes qui sont (complètement) associés,i.e. , ils ne contiennent aucun "0". Or cette association très forte pose des problèmes dans le cas des données bruitées. Par exemple, on peut imaginer un phénomène biologique réel représenté par le tableau booléen de la figure 3.1 à gauche. Deux groupes de synexpression valides y sont représentés : $(\{o_1o_2o_3\}, \{a_3\})$ et $(\{o_1o_2\}, \{a_1a_2a_3\})$. Le tableau booléen de la figure 3.1 à droite représente le résultat d'une expérience (de puces à ADN) qui arrive à capturer ce phénomène. Le concept $(\{o_1o_2\}, \{a_1a_2a_3\})$ est un module de transcription valide mais qui ne peut être extrait en utilisant les concepts formels à partir de la table des données réelles. Dans les contextes bruités, l'extraction, le

	a_1	a_2	a_3	a_4
o_1	1	1	1	0
o_2	1	1	1	0
o_3	0	0	1	0

	a_1	a_2	a_3	a_4
o_1	1	1	1	0
o_2	1	0	1	1
o_3	0	0	1	0

FIG. 3.1 – Phénomène réel (gauche) et contexte à étudier (droite)

post-traitement et l'interprétation des concepts formels deviennent très difficiles. En d'autres termes, nous sommes en présence d'une très grande sensibilité au bruit. Or,

non seulement les données d'expression numériques sont bruitées du fait de la complexité des techniques de mesure, mais aussi l'encodage des propriétés booléennes à partir des données numériques peut également introduire du bruit.

3.1.2 Travaux connexes

Les récentes techniques de bi-partitionnement tendent à fournir des rectangles plus robustes au bruit mais au moyen de recherches heuristiques (optimisations locales) et surtout sans recouvrement [39, 85] (voir la section 1.4.1). D'autres approches ont été proposées dans la communauté de l'extraction de motifs sous contraintes. Dans [99], les auteurs étendent la définition des ensembles fréquents à des ensembles tolérants au bruit. Ils proposent deux types de motifs qu'ils appellent "ETI forts" (error-tolerant frequent itemsets) et "ETI faibles". Le premier type de motifs est composé d'ensembles d'attributs e tels le nombre d'objets contenant au moins 1-ϵ valeurs "1" sur e est supérieur à K. ϵ correspond à un seuil d'erreur (un pourcentage) et k au seuil de fréquence. Pour ce type de motifs, l'erreur est bornée par les objets. Les ETI forts sont des ensembles d'attributs e tels que pour au moins k objets l il y a au moins $1 - \epsilon$ valeurs "1" dans le bi-ensemble (l, e). Les auteurs utilisent la propriété que si un ensemble d'attributs est un ETI fort alors il est aussi un ETI fiable. Ils proposent alors d'utiliser un algorithme par niveau de type a priori pour calculer les ETI faibles et ensuite, par post-traitement, de calculer les ETI forts. En revanche, comme la propriété de ETI faibles n'est pas anti-monotone, les nouveaux candidats sont générés à partir des candidats du niveau précédent mais en ajoutant n'importe quel attribut. Pour réduire la complexité de l'extraction, ils proposent un algorithme glouton calculant une solution incomplète. Dans [90], les auteurs recherchent à définir ces ensembles fréquents avec exceptions par l'intermédiaire d'une contrainte anti-monotone. Ils proposent alors d'extraire tous les ETI faibles dont tous leurs sous-ensembles sont aussi des ETI faibles. Il suffit alors d'exploiter cette contrainte anti-monotone dans un algorithme d'extraction d'ensembles fréquents. Dans [1], les auteurs recherchent une collection de k motifs qui approxime au mieux des collections complètes d'ensembles fréquents. La collection de k motifs extraite est satisfaisante si les sous-ensembles de ces k motifs est proche de la collection approximée. Les auteurs montrent que même en utilisant des algorithmes simples pour les calculer, les approximations restent très satisfaisantes.

L'extension de tels ensembles denses à des bi-ensembles est difficile. Dans [46], les auteurs calculent des motifs appelés "geometrical tiles" qui sont des bi-ensembles de lignes et de colonnes contigus au regard d'un ordre considéré (ayant une densité de valeurs 1 supérieure à un seuil fixé). Ils souhaitent extraire des motifs qui maximisent la vraisemblance. Or ce critère n'est pas monotone par rapport à l'ordre d'énumération traditionnel (inclusion ensembliste). Ce critère ne peut donc pas être exploité efficacement. Ainsi, pour extraire ces motifs, ils proposent d'utiliser un algorithme non

déterministe d'optimisation locale qui ne garantit pas la qualité globale des motifs extraits. L'algorithme débute avec un motif aléatoire puis essaye de l'étendre (en ajoutant des lignes et/ou des colonnes contigues) ou de le réduire. Lorsque le maximum de vraisemblance ne peut plus augmenté alors le motif est extrait. Ensuite, le processus est réitéré avec d'autres motifs aléatoires. Dans cette méthode, l'hypothèse forte qui est faite est qu'il existe un ordre sur les colonnes, hypothèse qui n'est pas satisfaisante dans notre cas.

Une autre approche importante consiste à étudier de façon systématique la notion de représentation condensée des collections de concepts formels ou de bi-ensembles denses, qu'il s'agisse de représentations exactes ou approximatives. L'objectif est alors de ne représenter, ou mieux de ne calculer, qu'un sous-ensemble des collections tout en pouvant retrouver, plus ou moins exactement mais à un faible coût, l'ensemble de la collection. Les δ-libres et leur pseudo-fermeture [78, 77] permettent de définir des motifs denses. Principalement, le nombre de valeurs "0" acceptées par colonne est borné par δ. Comme la contrainte de δ-liberté est anti-monotone, l'extraction de ces motifs est efficace en pratique même dans des grands jeux de données. Par contre, la manière dont sont définis ces motifs n'est pas complètement satisfaisante. Par exemple, le nombre de valeurs "0" par ligne ne peut être borné. Ce type de motif est typiquement un compromis entre l'efficacité de l'extraction et le type de contraintes que doit satisfaire le motif. L'approche des représentations condensées doit aussi intégrer des approches de "zoom" comme, par exemple, les travaux présentés dans [95] pour construire des treillis de Galois à différents niveaux d'abstraction. Cette méthode utilise une partition sur les objets qui permet de réduire le nombre de motifs extraits. Les auteurs utilisent une partition sur les lignes et ne conservent que les concepts qui sont en "accord" avec cette partition : une situation s appartient à l'extension d'un ensemble G si $\alpha\%$ des objets de la même classe que s satisfont G et que s satisfait aussi G.

3.1.3 Notre objectif

Nous venons de soulever le problème de la sensibilité des 1-rectangles au bruit. Avant d'aller plus loin, il faut s'intéresser à une notion importante : la maximalité des motifs. Dans la section 2, nous avons insisté sur la pertinence de la propriété de maximalité pour les 1-rectangles en privilégiant les concepts formels par rapport aux ensembles d'attributs (itemsets). En effet, elle permet d'abord d'améliorer dans la majorité des cas (données très corrélées) la faisabilité des extractions par rapport à celle sur les ensembles d'attributs. Dans ce cas, la maximalité des motifs est vue sous l'aspect de la représentation condensée des ensembles d'attributs. Ensuite, il nous est apparu qu'en pratique les biologistes, et sûrement de nombreux autres utilisateurs, s'intéressent aux motifs qui contiennent le plus d'information possible, i.e. le plus d'associations possible entre gènes et conditions. Ces considérations nous ont poussé

à travailler sur le problème de l'extraction de bi-ensembles maximaux capturant des associations fortes entre des lignes et des colonnes (bi-ensembles denses en valeurs vraies) mais admettant certaines exceptions (valeurs fausses). L'objectif est de réduire la sensibilité au bruit des concepts formels tout en conservant leur pertinence vis-à-vis des problèmes traités.

La notion de densité d'un bi-ensemble peut être envisagée sous deux angles selon que l'on mesure le nombre de 0 par ligne/colonne ou sur l'ensemble du bi-ensemble (densité forte versus faible) et selon que l'on considère ce nombre de manière absolue ou relativement à la taille du bi-ensemble (densité absolue versus relative). La densité faible peut engendrer la présence dans les bi-ensembles de lignes ou de colonnes de 0 c'est-à-dire des éléments qui ne sont jamais en relation avec les autres éléments du bi-ensemble. D'un autre côté, la densité relative est difficile à définir si l'on souhaite imposer la maximalité des motifs. Ainsi, nous avons décidé d'utiliser la densité forte absolue. Elle a de plus la propriété très importante d'être anti-monotone par rapport à $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$ (voir paragraphes suivants).

La contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}(X,Y)$ de la définition 3.2 impose que la densité forte absolue pour les lignes (resp. colonnes) du bi-ensemble (X,Y) soit inférieure à α (resp. α'). Les lignes (resp. colonnes) du bi-ensemble (X,Y) doivent contenir au plus α (resp. α') valeurs 0 sur les colonnes de Y (resp. sur les lignes de X). Ces bi-ensembles seront appelés bi-ensembles denses.

Définition 3.1 (Fonctions $\mathcal{Z}_{\mathbf{r}-l}$ et $\mathcal{Z}_{\mathbf{r}-c}$) *Nous notons $\mathcal{Z}_{\mathbf{r}-l}(x,Y)$ le nombre de 0 contenu dans la ligne x sur les colonnes de Y dans $\mathbf{r}$: $\mathcal{Z}_{\mathbf{r}-l}(x,Y) = \sharp\{y \in Y | (x,y) \notin \mathbf{r}\}$ avec $\sharp$ la taille de l'ensemble. De la même façon, $\mathcal{Z}_{\mathbf{r}-c}(y,X) = \sharp\{x \in X | (x,y) \notin \mathbf{r}\}$ est le nombre de 0 contenu dans la colonne y sur les lignes de X dans $\mathbf{r}$.*

Définition 3.2 (Contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$) *Un bi-ensemble (X,Y) satisfait $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ ssi*

$\forall x \in X,\ \mathcal{Z}_{\mathbf{r}-l}(x,Y) \leq \alpha,\ \ \forall y \in Y,\ \mathcal{Z}_{\mathbf{r}-c}(y,X) \leq \alpha'$

	a_1	a_2	a_3
o_1	1	1	1
o_2	1	1	0
o_3	1	0	1
o_4	1	0	0
o_5	0	1	0

TAB. 3.1 – Jeu de données $\mathbf{r}_4$

Propriété 3.1 *La contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ est anti-monotone par rapport à $\subseteq$.*

Preuve. Soit un bi-ensemble (X, Y) satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ alors pour tout bi-ensemble (X', Y') tel que $(X', Y') \subseteq (X, Y)$ on a :

$\forall x \in X',\ \mathcal{Z}_{\mathbf{r}-l}(x, Y') \leq \mathcal{Z}_{\mathbf{r}-l}(x, Y) \leq \alpha$

$\forall y \in Y',\ \mathcal{Z}_{\mathbf{r}-c}(y, X') \leq \mathcal{Z}_{\mathbf{r}-c}(y, X) \leq \alpha'$

Donc tout bi-ensemble (X', Y') satisfait $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$. $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ est anti-monotone par rapport à $\subseteq$.

Exemple 3.1 *Soit le jeu de données* $\mathbf{r}_4$ *(voir la table 3.1), le bi-ensemble* ($\{o_1o_2o_3\}$,-$\{a_1a_2a_3\}$) *satisfait la contrainte* $\mathcal{C}_{1\,1\,\mathbf{r}_4-d}$.

A partir de cette contrainte, on pourrait définir un nouveau type de concepts avec exception potentiellement plus tolérant au bruit : les bi-ensembles maximaux qui satisfont $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$. Mais cette proposition n'est pas du tout satisfaisante, les motifs ainsi extraits ne seraient pas pertinents. Par exemple avec $\alpha = \alpha' = 1$, les bi-ensembles $(\{o_1o_2o_3o_5\}, \{a_1a_2\})$ et $(\{o_1o_2o_4o_5\}, \{a_1a_2\})$ seraient extraits de $\mathbf{r}_4$. Or, les lignes o_3 et o_4 sont identiques sur $\{a_1a_2\}$ mais ne peuvent appartenir au bi-ensemble. Cet exemple illustre les deux problèmes importants de ce type de motifs. D'abord, le nombre de motifs extraits serait très important car on pourrait combiner de nombreuses façons les lignes et les colonnes identiques tout en conservant la maximalité et la contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$. Ensuite, l'interprétation de tels motifs n'est pas aisée : comment interpréter le rôle de ces lignes et de ces colonnes identiques dans les motifs. Nous proposons d'ajouter une contrainte supplémentaire. Nous avons envisagé deux stratégies possibles. Elles sont décrites dans les sections 3.2 et 3.3. La première est moins stringente mais plus difficile à exploiter alors que la seconde est plus stringente et offre une définition plus déclarative. Dans les deux cas, il faudra considérer des bi-ensembles grands devant α et α'.

Les sections 3.2 et 3.3 présentent chacune un nouveau type de motifs avec exceptions, un algorithme pour les calculer et des expérimentations mettant en avant leurs forces et leurs faiblesses. Les premiers motifs appelés CBS (pour Consistent Bi-Sets) sont obtenus par un post-traitement d'une collection de concepts formels. La seconde approche offre un cadre pour l'extraction sous-contraintes de bi-ensembles avec exceptions.

3.2 Les motifs CBS

3.2.1 Formalisation du problème

Dans [14], nous avons proposé une première stratégie simple pour traiter le cas des lignes et des colonnes identiques. Nous proposons soit d'ajouter toutes les lignes

(resp. les colonnes) identiques par rapport à l'ensemble des colonnes (resp. des lignes) au bi-ensemble si la contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-d}$ est satisfaite ou de n'en mettre aucune si ce n'est pas le cas.

Exemple. Dans $\mathbf{r}_4$, nous avons vu le problème soulevé par les deux bi-ensembles $(\{o_1o_2o_3o_5\}, \{a_1a_2\})$ et $(\{o_1o_2o_4o_5\}, \{a_1a_2\})$. En considérant d'abord le bi-ensemble $(\{o_1o_2o_5\}, \{a_1a_2\})$, on peut se rendre compte que les lignes o_3 ou o_4 peuvent être ajoutées à ce bi-ensemble mais pas les deux en même temps (échec de $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ avec $\alpha = \alpha' = 1$). On a vu que la stratégie visant à produire deux motifs l'un contenant o_3 et l'autre o_4 n'est pas satisfaisante. Nous proposons de n'ajouter aucune de ces lignes, le bi-ensemble $(\{o_1o_2o_5\}, \{a_1a_2\})$ formant ainsi un bi-ensemble solution (a_3 ne pouvant être ajouté). Effectivement, comme les lignes o_3 et o_4 sont identiques et qu'elles ne peuvent être ajoutées ensemble alors aucune n'est ajoutée dans le bi-ensemble.

La contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-cons}$ (définition 3.3) formalise cette stratégie.

Définition 3.3 (Contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-cons}$) *soit (X, Y) un bi-ensemble, (X, Y) satisfait $\mathcal{C}_{\alpha\alpha'\mathbf{r}-cons}$ ssi*

$\forall i \in \mathcal{O}/X,\ E = \{x \in \mathcal{O} \mid \phi(i) \cap Y = \phi(x) \cap Y\}$ *alors* $\neg\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}((X \cup E, Y))$

$\forall i \in \mathcal{A}/Y,\ E = \{y \in \mathcal{A} \mid \psi(i) \cap X = \psi(y) \cap X\}$ *alors* $\neg\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}((X, Y \cup E))$

Finalement, nous souhaitons extraire les bi-ensembles (X, Y) maximaux qui satisfont $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d} \wedge \mathcal{C}_{\alpha\alpha'\mathbf{r}-cons}$, i.e. , tels qu'il n'existe pas de bi-ensembles (X', Y') qui satisfont la contrainte précédente et que $(X, Y) \subseteq (X', Y')$.

3.2.2 Algorithme

Nous proposons d'utiliser une méthode basée sur un post-traitement de collections de concepts formels. L'idée est de fusionner les concepts formels et de ne conserver que les bi-ensembles maximaux qui satisfont $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$. Cette méthode est incomplète mais permet en pratique d'étendre les concepts formels avec des exceptions.

Définition 3.4 (Union de bi-ensembles) *Soit $B_1 = (X_1, Y_1)$ et $B_2 = (X_2, Y_2)$ deux bi-ensembles alors l'union de B_1 et de B_2 est $B_1 \sqcup B_2 = (X_1 \cup X_2, Y_1 \cup Y_2)$. Par définition, l'union de deux concepts formels différents n'est pas un concept formel puisqu'il contient nécessairement un 0.*

Nous proposons de calculer la collection $\mathcal{K}_{\alpha\alpha'}$ suivante :

Définition 3.5 *Soit* $U = \{\bigsqcup_{i \in \mathcal{K}'} i \mid \mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ *et* $\mathcal{K}' \subseteq \mathcal{K}\}$ *avec* $\mathcal{K}$ *la collection des concepts formels alors nous définissons la collection* $\mathcal{K}_{\alpha\alpha'}$ *telle que*

$$\mathcal{K}_{\alpha\alpha'} = \{s \in U \mid \nexists s\prime \in U \text{ tel que } s \subseteq s\prime\}$$

Les éléments de $\mathcal{K}_{\alpha\alpha'}$ *sont les ensembles maximaux de* U *et sont appelés des* CBS*.*

Pour extraire $\mathcal{K}_{\alpha\alpha'}$, on peut adapter un algorithme de recherche d'ensembles maximaux (voir e.g., [48]). Les attributs sont alors remplacés par les concepts formels et les ensembles d'attributs par les bi-ensembles résultants de la fusion des concepts associés aux ensembles d'attributs. De plus la contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ peut être exploitée comme contrainte anti-monotone en plus de la contrainte de fréquence (anti-monotone) habituellement utilisée.

Preuve. Soit I un ensemble d'attributs composé de n concepts $C_1 \dots C_n$. Alors tout ensemble d'attributs successeur S de I est tel que $S = I \cup E$ avec E un ensemble de concepts. Or comme $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ est anti-monotone suivant $\subseteq$ et que $I \subseteq S$ alors $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ peut être exploitée comme contrainte anti-monotone pour élaguer l'espace de recherche.

Cette méthode basée sur la fusion de concepts ne permet pas de calculer exactement la collection des bi-ensembles satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d} \wedge \mathcal{C}_{\alpha\alpha'\mathbf{r}-cons}$ (voir l'exemple 3.2).

Exemple 3.2 *Le contexte booléen* $\mathbf{r}_5$ *(voir Table 3.2) contient 6 concepts formels :*

$$(\{s_1 s_2 s_4\}, \{g_3\})$$
$$(\{s_1 s_3 s_5\}, \{g_2\})$$
$$(\{s_2 s_3 s_6\}, \{g_1\})$$
$$(\{s_1\}, \{g_2 g_3 g_6\})$$
$$(\{s_2\}, \{g_1 g_3 g_5\})$$
$$(\{s_3\}, \{g_1 g_2 g_4\})$$

Le bi-ensemble $(\{s_1 s_2 s_3\}, \{g_1 g_2 g_3\})$ *satisfait* $\mathcal{C}_{\alpha\alpha'\mathbf{r}_5-d} \wedge \mathcal{C}_{\alpha\alpha'\mathbf{r}_5-cons}$ *avec* $\alpha = \alpha' = 1$*. Or aucun des 6 concepts formels n'est un sous-ensemble de ce motif. Ainsi, il ne peut être obtenu par la fusion de concepts formels de* $\mathbf{r}_5$*.*

De plus, ces motifs ne sont pas munis de fonctions : à un ensemble de lignes (resp. colonnes) donné peut être associé plusieurs ensembles de colonnes (resp. lignes) différents.

Exemple. Dans $\mathbf{r}_4$, les bi-ensembles $(\{o_1 o_2 o_3 o_4\}, \{a_1 a_2 a_3\})$ et $(\{o_1 o_2 o_3 o_5\}, \{a_1 a_2 a_3\})$ sont des CBS avec $\alpha = \alpha' = 1$. A l'ensemble $\{a_1 a_2 a_3\}$ est associé deux ensembles différents de lignes $\{o_1 o_2 o_3 o_4\}$ et $\{o_1 o_2 o_3 o_5\}$.

	g_1	g_2	g_3	g_4	g_5	g_6
s_1	0	1	1	0	0	1
s_2	1	0	1	0	1	0
s_3	1	1	0	1	0	0
s_4	0	0	1	0	0	0
s_5	0	1	0	0	0	0
s_6	1	0	0	0	0	0

TAB. 3.2 – Jeu de données $\mathbf{r}_5$

3.2.3 Exemple d'exécution

Les concepts de $\mathbf{r}_4$ sont :

$$c_1 = (\{o_1\}, \{a_1a_2a_3\}) \qquad c_2 = (\{o_1o_2\}, \{a_1a_2\})$$
$$c_3 = (\{o_1o_3\}, \{a_1a_3\}) \qquad c_4 = (\{o_1o_2o_5\}, \{a_2\})$$
$$c_5 = (\{o_1o_2o_3o_4\}, \{a_1\})$$

A partir des concepts de $\mathbf{r}_4$, on peut rechercher les CBS de $\mathbf{r}_4$ avec $\alpha = \alpha' = 1$.

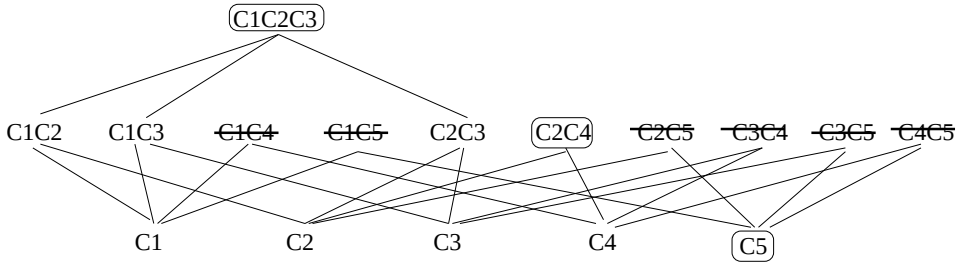


FIG. 3.2 – Extraction des CBS ($\alpha = \alpha' = 1$) de $\mathbf{r}_4$

La figure 3.2 montre comment la collection des CBS est extraite de $\mathbf{r}_4$. Les trois motifs entourés forment la collection finale : $\{\{c_1, c_2, c_3\}, \{c_2, c_4\}, \{c_5\}\}$. Les motifs barrés ne satisfont pas $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$. Finalement, ces trois motifs correspondent aux bi-ensembles suivants : $(\{o_1o_2o_3\}, \{a_1a_2a_3\})$, $(\{o_1o_2o_5\}, \{a_1a_2\})$ et $(\{o_1o_2o_3o_4\}, \{a_1\})$.

3.2.4 Validation expérimentale

Données synthétiques

Pour montrer la pertinence des CBS en présence de données bruitées, nous avons d'abord généré des jeux de données synthétiques. Notre objectif est de montrer que les

CBS permettent de redécouvrir des motifs (concepts) présents dans les données avant l'introduction de bruits. Ainsi, nous avons généré des jeux de données booléennes contenant 20 concepts formels sans recouvrement contenant chacun 5 objets et 5 attributs (matrice de taille 100*100). Ensuite nous avons introduit un bruit uniforme dans les données en utilisant deux niveaux de bruit : 5% et 10%. Chaque valeur a 5% (resp. 10%) de chance de changer de valeur passant de 0 en 1 ou de 1 en 0. Pour chaque niveau de bruit, dix jeux de données ont été générés. On s'est intéressé au nombre de CBS extraits en fonction de α et de α' et de la taille minimale des motifs sur les deux dimensions. La figure 3.3 en haut montre la moyenne et l'écart type du nombre de CBS (en ordonnée) en fonction de la taille minimale des motifs sur les objets et les attributs (en abscisse) avec 5% de bruit. La figure 3.3 en bas représente les mêmes informations mais avec 10% de bruit. Chaque courbe représente une valeur différente de α et α' entre 0 et 2. Par exemple, sur la figure 3.3 en bas, il y a 126 CBS en moyenne contenant au moins 3 objets et 3 attributs avec $\alpha = 2$ et $\alpha' = 1$.

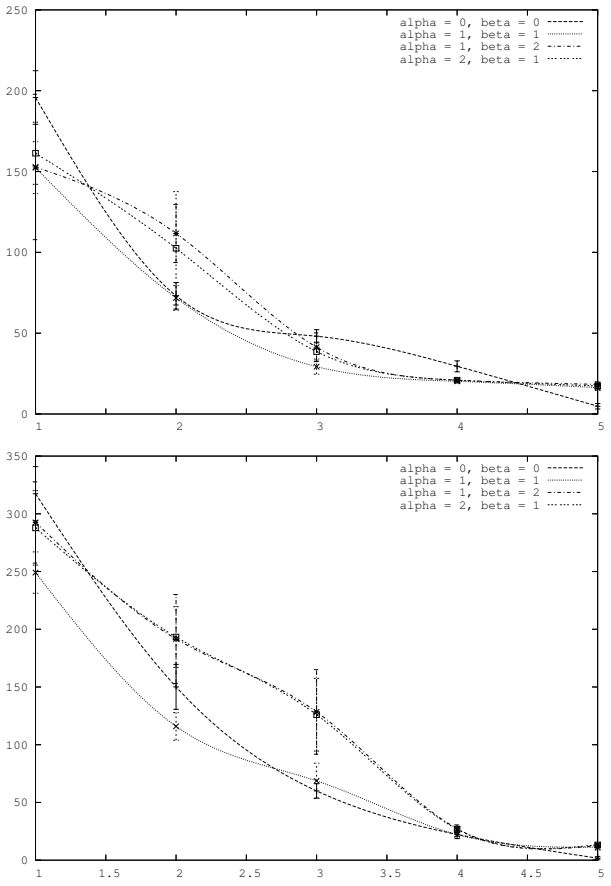


FIG. 3.3 – Nombre de CBS en fonction de leur taille sur les deux dimensions avec 5% de bruit (en haut) et 10% de bruit (en bas)

Sur le jeu de données avec 5% de bruit, il y a en moyenne 196 concepts (courbe avec $\alpha = \alpha' = 0$) parmi lesquels 48 ont au moins 3 objets et 3 attributs. Avec 10% de bruit, il y a 317 concepts en moyenne dont 60 qui ont au moins 3 objets et attributs, et 2 qui ont au moins 5 objets et attributs. Dans la collection de concepts, il est difficile de retrouver les 20 concepts formels originaux. Quand α et α' sont égaux à 5, la collection extraite est presque égale à la collection des 20 concepts initiaux. Par exemple, avec $\alpha = \alpha' = 1$, il y a 20.2 (resp. 22.1) CBS en moyenne ayant une taille supérieure à 4 dans le jeu de données avec 5% (resp. avec 10%) de bruit.

Extraction à partir d'une collection de concepts formels sur des données d'expression

Malheureusement, le nombre de CBS extraits peut augmenter considérablement avec α et α', même si cette stratégie est plus stringente que l'approche naïve (tous les bi-ensembles maximaux satisfaisants $\mathcal{C}_{\delta\delta'\mathbf{r}-d}$). Effectivement dans beaucoup de jeux de données réelles, il n'est pas possible d'extraire la collection complète des concepts ou du moins cette collection peut comporter des millions de motifs. Il n'est pas envisageable alors de post-traiter cette collection pour extraire les CBS. Dans ces cas là, des contraintes peuvent être poussées pour réduire la taille de la collection et rendre l'extraction faisable. Par exemple, avec D-Miner on peut s'intéresser aux motifs ayant un nombre minimal de lignes et de colonnes. Il faut alors s'intéresser à la possible extension de collections incomplètes de concepts.

Pour illustrer ce point, nous avons utilisé la matrice relative aux facteurs de transcription (voir la section 4.3). Elle contient 356 lignes et 155 colonnes. Cette matrice contient plus de 5 millions de concepts. Il n'est pas possible d'utiliser la méthode proposée sur la collection complète des concepts formels. Nous nous sommes alors intéressés aux motifs issus de la fusion des concepts contenant au moins 25 gènes et 10 facteurs de transcription. En utilisant D-Miner 1 699 concepts satisfaisant ces contraintes ont été extraits. A partir de ces concepts, nous avons extrait les CBS. La table 3.3 fournit le nombre de CBS extraits pour quatre valeurs de α et α' en fonction du nombre de concepts fusionnés (n). Par exemple, 1546 CBS sont extraits avec $\alpha = \alpha' = 1$, dont 54 sont issus de la fusion de 2 concepts formels et 2 de la fusion de 5 concepts formels.

Cette expérience montre que, même si l'on considère seulement de grands motifs avec des valeurs relativement petites de α et α', la méthode proposée fonctionne toujours et permet d'étendre les concepts. Par exemple, au moins 6 concepts sont fusionnés avec $\alpha = \alpha' = 1$ alors que 15 concepts sont fusionnés lorsque $\alpha = \alpha' = 4$. Dans ce jeu de données, nous obtenons de grands CBS contenant peu de valeurs 0. Typiquement, le CBS ($\alpha = \alpha' = 4$) issu de la fusion de 15 concepts est composé de 36 gènes et de 12 facteurs de transcription et ne contient que 3.7% de valeurs 0. La table 3.4 montre le nombre de 0 contenu dans chaque facteur de transcription pour

n	$\alpha = \alpha' = 1$	$\alpha = \alpha' = 2$	$\alpha = \alpha' = 3$	$\alpha = \alpha' = 4$
1	1450	1217	927	639
2	54	49	61	95
3	31	57	75	73
4	8	40	50	64
5	2	8	25	58
6	1	3	11	29
7	0	0	6	11
8	0	0	1	12
9	0	0	0	2
10	0	0	1	6
11	0	0	0	0
12	0	0	0	3
13	0	0	0	1
14	0	0	1	0
15	0	0	0	1
Total	1546	1374	1158	994

TAB. 3.3 – Nombre de CBS produits par la fusion de n concepts pour différentes valeurs de α et α'

ce motif.

3.2.5 Conclusion

Les méthodes basées sur l'extraction de concepts formels, ou plus généralement de motifs locaux ensemblistes sont très sensibles au bruit dans les données. Pour résoudre ce problème, nous avons essayé d'étendre la notion de concepts formels à des motifs tolérants au bruit. Dans ce chapitre, nous avons proposé une première approche visant à calculer des bi-ensembles maximaux ayant un nombre borné de valeurs 0 sur les deux dimensions appelés CBS. Une méthode proposée consiste en un post-traitement d'une collection de concepts formels. L'idée est de procéder à une fusion de certains concepts formels de telle sorte que le nombre de valeurs 0 par ligne et par colonne soit borné. Cette contrainte étant anti-monotone suivant $\subseteq$, elle peut être exploitée activement. Ce procédé peut être réalisé en adaptant un algorithme d'extraction d'ensembles maximaux. Dans [87], nous avons proposé une autre façon d'obtenir des motifs tolérants au bruit par fusion de concepts formels en utilisant cette fois-ci une optimisation locale. Cette méthode propose essentiellement une mesure de distance entre bi-ensembles qui peut être utilisée pour regrouper les concepts formels "proches" en adaptant des méthodes de clustering hiérarchique.

L'extraction des CBS devient très difficile dès lors que la matrice contient plusieurs milliers de concepts formels. On peut regretter que les CBS ne sont pas munis de fonctions, c'est-à-dire qu'à un même ensemble d'objets, différents ensembles d'attributs peuvent être associés. Cette propriété est pourtant très intéressante pour aider l'interprétation des motifs extraits.

Nombre de zéro
0
0
0
1
3
2
0
3
0
3
4
0

TAB. 3.4 – Nombre de valeurs 0 pour chaque facteur de transcription du CBS (36×12) issu de la fusion de 15 concepts avec $\alpha = \alpha' = 4$

Cette méthode s'apparente à de l'usage multiple de motifs [62]. En effet, nous utilisons les concepts formels pour générer de nouveaux motifs. Au lieu d'extraire directement ces motifs en utilisant un algorithme d'extraction de bi-ensembles sous contraintes, on utilise les propriétés de maximalité des concepts formels vis-à-vis de la contrainte de 1-rectangle pour induire automatiquement d'autres propriétés.

3.3 Les motifs DR-bi-sets

3.3.1 Introduction

Dans la section précédente, nous avons proposé un nouveau type de bi-ensemble dense qui s'est avéré particulièrement intéressant pour améliorer la qualité des motifs extraits dans des données bruitées. En revanche, ces motifs ne sont pas munis de fonctions et n'offrent pas un cadre générique à l'extraction de bi-ensembles sous contraintes. Notre conviction est que les bi-ensembles denses doivent satisfaire différentes notions associées aux concepts formels : les notions de justesse et de complétude des extractions, la dualité entre les objets et les attributs, sa définition très déclarative et finalement le rôle important que jouent les contraintes pour améliorer la pertinence des motifs extraits (et accessoirement la faisabilité des extractions). C'est autour de ces notions que nous avons défini un nouveau type de bi-ensembles denses appelés DR-bi-set [13]. De façon intuitive, les DR-bi-sets sont des bi-ensembles contenant un nombre borné de valeurs "0" par ligne et par colonne et tels que les lignes et les colonnes qui n'en font pas partie contiennent plus de "0" qu'à l'intérieur.

Cette définition est très intuitive, offrant une extension naturelle des concepts formels.

	g_1	g_2	g_3	g_4
s_1	1	1	1	1
s_2	1	1	0	0
s_3	0	1	0	0
s_4	0	0	1	0
s_5	0	0	0	0

FIG. 3.4 – Contexte booléen $\mathbf{r}_6$

Exemple. Le bi-ensemble $(\{s_1s_2s_3\}, \{g_1g_2\})$ est un DR-bi-set dans $\mathbf{r}_6$. En effet, il contient au maximum une valeur "0" et toutes les lignes et les colonnes à l'extérieur en contiennent au moins 2. Ce bi-ensemble tend à capturer beaucoup de valeurs "1" par rapport aux lignes et aux colonnes qui sont à l'extérieur.

Nous avons privilégié une approche "extraction de bi-ensembles sous-contraintes" permettant d'exploiter à la fois les contraintes définissant les DR-bi-sets mais aussi d'autres contraintes comme les contraintes monotones et anti-monotones sur $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$.

3.3.2 Formalisation du problème

Nous voulons extraire des bi-ensembles (X, Y) satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ (voir la section 3.1.3) et tels que les lignes de X (resp. les colonnes de Y) aient une densité en 1 plus importante sur les colonnes de Y (resp. sur les lignes de X) que sur les autres colonnes, i.e. $\mathcal{A} \setminus Y$ (resp. les autre lignes, i.e. $\mathcal{O} \setminus X$). Cette contrainte est formalisée par la contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ (voir définition 3.6).

Définition 3.6 (Contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$) *Soit un bi-ensemble (X, Y) et deux paramètres δ et δ' strictement positifs, (X, Y) satisfait la contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ ssi*

$$(\forall s \in \mathcal{O} \setminus X,\ \forall t \in X,\ \mathcal{Z}_l(s, Y) \geq \mathcal{Z}_l(t, Y) + \delta)$$
$$\wedge(\forall g \in \mathcal{A} \setminus Y,\ \forall h \in Y,\ \mathcal{Z}_c(g, X) \geq \mathcal{Z}_c(h, X) + \delta')$$

Les paramètres δ et δ' permettent de fixer la différence de densité entre les éléments à l'intérieur et à l'extérieur du bi-ensemble. Plus ils sont grands, plus cette différence doit être importante.

Exemple. Le bi-ensemble $(\{g_1g_2\}, \{s_1s_2s_3\})$ satisfait la contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ avec $\delta = \delta' = 1$. En effet, toutes les lignes (resp . colonnes) du bi-ensemble contiennent un seul 0 sur $\{g_1g_2\}$ (resp. $\{s_1s_2s_3\}$) et toutes les lignes (resp. les colonnes) à l'extérieur contiennent au moins un 0 en plus sur $\{g_1g_2\}$ (resp. sur $\{s_1s_2s_3\}$).

Les bi-ensembles qui satisfont les contraintes primitives $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ sont clairement une généralisation des concepts formels. En effet, les concepts formels peuvent aussi être définis de la façon suivante :

Définition 3.7 *Un bi-ensemble (X, Y) est un concept formel ssi*

(1) $\forall x \in X,\ \mathcal{Z}_l(x, Y) = 0,\ \ \forall y \in Y,\ \mathcal{Z}_c(y, X) = 0$

(2) $\forall x \in \mathcal{O} \setminus X,\ \mathcal{Z}_l(x, Y) \geq 1,\ \ \forall y \in \mathcal{A} \setminus Y,\ \mathcal{Z}_c(y, X) \geq 1$

Autrement dit, un concept formel est un bi-ensemble satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d} \wedge \mathcal{C}_{\delta\delta'\mathbf{r}-p}$ avec $\delta = \delta' = 1$ et $\alpha = \alpha' = 0$.

$\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ est une généralisation directe de l'équation 1 de la Définition 3.7. Alors que $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ généralise l'équation 2 de la définition 3.7 en imposant que tous les éléments à l'extérieur du bi-ensemble contiennent au moins δ valeurs 0 en plus que ceux qui sont à l'intérieur du bi-ensemble.

Les paramètres α et α' contrôlent la densité du bi-ensemble alors que les paramètres δ et δ' imposent une différence significative de densité avec les éléments extérieurs. La contrainte $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ (voir Définition 3.1) est anti-monotone par rapport à $\subseteq$ et peut donc être exploitée pour élaguer l'espace de recherche. $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ n'est ni monotone ni anti-monotone mais nous verrons dans la section 3.3.3 comment l'exploiter efficacement grâce à une représentation particulière des bi-ensembles.

La figure 3.5 montre la collection des bi-ensembles satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ avec $\alpha = 5$, $\alpha' = 4$, et $\delta = \delta' = 1$ dans $\mathbf{r}_6$ (voir la figure 3.4) et ordonnée par $\subseteq$. Chaque niveau indique le nombre maximal de zéros contenu dans le bi-ensemble par ligne et par colonne. Si l'on utilise les paramètres $\alpha = \alpha' = 1$ et $\delta = \delta' = 1$, alors une sous-collection contenant 5 bi-ensembles est obtenue, 4 étant des concepts formels.

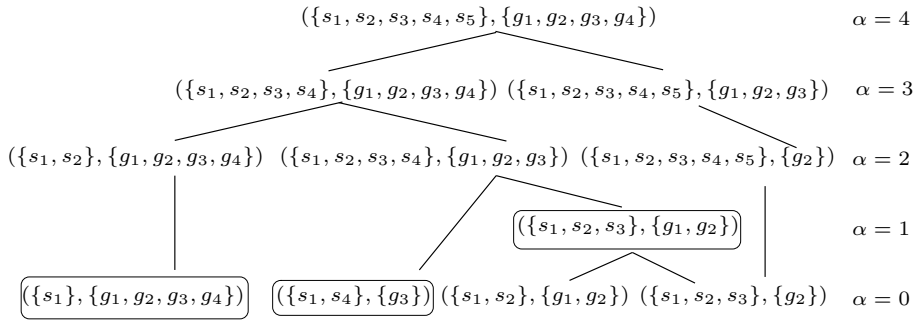


FIG. 3.5 – Les bi-ensembles satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}_6-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}_6-p}$ avec $\alpha = 5$, $\alpha' = 4$ et $\delta = \delta' = 1$

Ces deux contraintes n'imposent pas la maximalité des motifs. Par exemple, dans $\mathbf{r}_6$, si $(\{s_1, s_2, s_3\}, \{g_1, g_2\})$ est noté B alors nous avons $(\{s_1, s_2\}, \{g_1, g_2\}) \preceq B$ et

$(\{s_1, s_2, s_3\}, \{g_2\}) \preceq B$. La collection avec $\alpha = \alpha' = 1$ et $\delta = \delta' = 1$ contient donc des éléments qui ne sont pas maximaux. Or, nous souhaitons extraire des bi-ensembles maximaux (voir section 3.1.3). Finalement, ce sont les bi-ensembles maximaux satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ que nous appelons DR-bi-sets. La collection contenant tous les DR-bi-sets (avec α, α', δ et δ' fixés) est notée $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$. Par exemple, la collection $\mathcal{M}_{DR}^{1111}$ de $\mathbf{r}_6$ correspond aux trois bi-ensembles entourés de la figure 3.5.

La propriété suivante montre que la taille des DR-bi-sets augmente avec α et α'.

Propriété 3.2 *Soit $\alpha_1 \leq \alpha$, $\alpha'_1 \leq \alpha'$ alors $\forall X_1 \in \mathcal{M}_{DR}^{\alpha_1\alpha'_1\delta\delta'}$, $\exists X \in \mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ tel que $X_1 \preceq X$.*

Plus α et α' augmentent, plus la taille des bi-ensembles extraits augmente alors que les associations extraites avec des α et α' plus petits sont conservées. En pratique, une importante réduction de la taille des collections est constatée quand les paramètres sont convenablement choisis (voir section 3.3.3). Un effet de "zoom" est obtenu quand α et α' varient. La table 3.5 montre des collections $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ de $\mathbf{r}_6$.

	$\mathcal{M}_{DR}$	
α	$\delta = 1$	$\delta = 2$
0	$\{\{s_1\}, \{g_1, g_2, g_3, g_4\}\}$ $\{\{s_1, s_4\}, \{g_3\}\}$ $\{\{s_1, s_2\}, \{g_1, g_2\}\}$ $\{\{s_1, s_2, s_3\}, \{g_2\}\}$	$\{\{s_1\}, \{g_1, g_2, g_3, g_4\}\}$
1	$\{\{s_1\}, \{g_1, g_2, g_3, g_4\}\}$ $\{\{s_1, s_4\}, \{g_3\}\}$ $\{\{s_1, s_2, s_3\}, \{g_1, g_2\}\}$	$\{\{s_1\}, \{g_1, g_2, g_3, g_4\}\}$

TAB. 3.5 – Collections $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ de $\mathbf{r}_6$

Les DR-bi-sets sont munis de fonctions :

Propriété 3.3 (Fonctions pour $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$) *Les collections $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ sont munies de deux fonctions :*

$\forall\ (X, Y) \in \mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}\ \forall (X', Y') \in \mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ avec $(X, Y) \neq (X', Y')$, on a $X \neq X'$ et $Y \neq Y'$. A un ensemble de lignes (resp. de colonnes) correspond au maximum un ensemble de colonnes (resp. de lignes) dans $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$. La preuve de cette propriété est donnée en annexe D.

Propriété 3.4 (Monotonicité des functions (α fixé)) *soit $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ et $\mathcal{M}_{DR}^{\tau\tau'}$ le sous-ensemble de $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ tel que $(X, Y) \in \mathcal{M}_{DR}^{\tau\tau'}$ ssi (X, Y) contient au moins une*

ligne (resp. une colonne) avec τ (resp. τ') valeurs 0 sur Y (resp. X) et tel qu'aucune ligne (resp. colonne) n'en contient plus. Alors $\mathcal{M}_{DR}^{\tau\tau'}$ est munie de fonctions décroissantes. La preuve de cette propriété est donnée en annexe D.

3.3.3 Algorithme

Nous avons développé un algorithme appelé DR-Miner permettant de calculer les DR-bi-sets. C'est un algorithme en profondeur réalisant une énumération binaire à la fois sur les lignes et sur les colonnes. Pour pouvoir exploiter la contrainte $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ et les contraintes (anti)-monotones, nous utilisons une autre représentation des motifs candidats (nœuds de l'arbre d'énumération) inspirée de Dual-Miner [29]. Chaque candidat est représenté par trois bi-ensembles (Y, P, N) :

• $Y = (Y_S, Y_G)$ (pour Yes bi-set) contient les éléments appartenant au candidat et à tous ses fils (descendants).

• $P = (P_S, P_G)$ (pour Potential bi-set) contient les éléments qui n'ont pas encore été considérés, et dont on ne sait pas encore à quel ensemble ils appartiennent.

• $N = (N_S, N_G)$ (pour No bi-set) contient les éléments qui n'appartiennent pas au candidat et à ses fils.

Chaque élément de $\mathcal{O}$ et $\mathcal{A}$ appartient à un et un seul bi-ensemble parmi Y, P et N. La figure 3.6 illustre ce partitionnement des données pour un candidat donné (les lignes et les colonnes ayant été réordonnées).

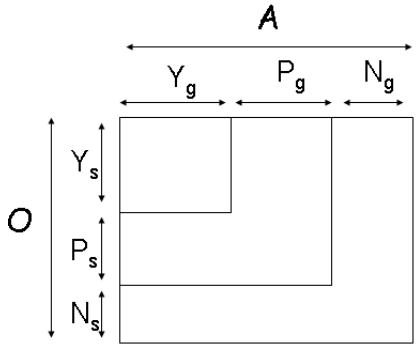


Fig. 3.6 – Partitionnement des données pour un candidat (Y, P, N)

Chaque candidat $(\{Y_S, Y_G\}, \{P_S, P_G\}, \{N_S, N_G\})$ représente le treillis d'un espace de recherche sur les bi-ensembles ayant comme plus petit élément (Y_S, Y_G) et comme plus grand élément $(Y_S \cup P_S, Y_G \cup P_G)$ avec $\subseteq$ comme relation de généralisation. Ce treillis est noté $[(Y_S, Y_G), (Y_S \cup P_S, Y_G \cup P_G)]$. Un bi-ensemble (X, Y) appartient à ce treillis si et seulement si $Y_S \subseteq X \subseteq Y_S \cup P_S$ et $Y_G \subseteq Y \subseteq Y_G \cup P_G$.

L'espace de recherche total est donc représenté par $((\emptyset,\emptyset),(\mathcal{O},\mathcal{A}),(\emptyset,\emptyset))$ qui est la racine de l'arbre d'énumération. L'algorithme DR-Miner (voir la table 3.6) est appelé la première fois sur ce candidat. Les feuilles de l'arbre sont des candidats tels que $P = (\emptyset,\emptyset)$.

Afin de réduire la taille de l'espace de recherche, il faut exploiter les contraintes $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$. Cette réduction est réalisée en réduisant la taille du treillis c'est-à-dire en diminuant le plus grand élément (voir la propriété 3.6) et en augmentant le plus petit élément (voir la propriété 3.6). Quand le treillis ne peut plus être réduit par les contraintes, il est découpé en deux nouveaux sous-treillis : le premier en déplaçant un élément e de P dans Y (le fils gauche) et le second en déplaçant e de P dans N (le fils droit). La figure 3.7 montre le processus d'énumération.

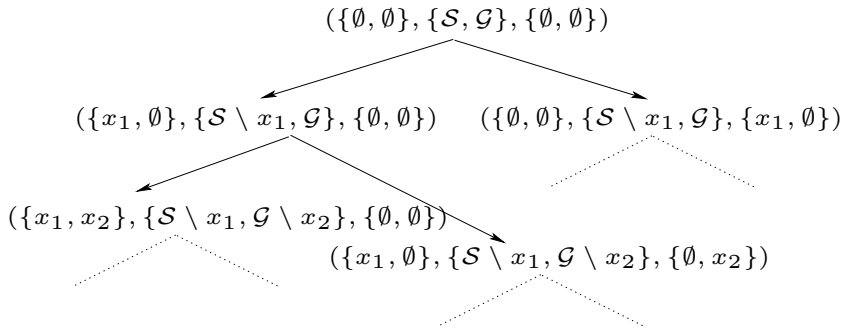


Fig. 3.7 – Processus d'énumération

Ensuite, pour chaque fils, la consistance est vérifiée. Par exemple, si l'élément minimal ne satisfait pas une contrainte anti-monotone (voir les propriétés 3.5-1 et 3.7) alors aucun des éléments du treillis ne la satisfont, le candidat est alors élagué. Si la consistance est vérifiée, le processus récursif continue.

Finalement, les bi-ensembles (Y_S, Y_G) des feuilles de l'arbre d'énumération sont les bi-ensembles satisfaisant $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$. Les DR-bi-sets sont les éléments maximaux de cette collection. Pour extraire seulement les maximaux, nous avons utilisé la stratégie de Dual-Miner pour pousser la maximalité.

Exploitation des contraintes

Les contraintes $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$ permettent de vérifier la consistence des candidats, i.e., en stoppant éventuellement l'énumération. Ces contraintes peuvent également être propagées permettant de réduire l'espace de recherche en déplaçant des éléments de P dans Y ou dans N.

Propriété 3.5 (Vérification de la consistance) *Soit un candidat* $T = (Y, P, N)$ *:*

1. *si* $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}(Y_S, Y_G)$ *n'est pas vérifiée alors aucun des fils de* T *ne la satisfait non plus (voir la figure 3.8(1)). Le candidat est élagué.*
2. *si* $\exists s \in N_S\ \exists t \in Y_S$ *tel que* $\mathcal{Z}_l(s, Y_G \cup P_G) < \mathcal{Z}_l(t, Y_G) + \delta$*, alors aucun des fils de* T *ne la satisfait non plus (voir la figure3.8(2)). Le candidat est élagué.*

Propriété 3.6 (Propagation de contraintes) *Soit un candidat* $T = (Y, P, N)$ *et* $s \in P_S$ *:*

1. *si* $\exists t \in Y_S$*,* $\mathcal{Z}_l(s, Y_G \cup P_G) < \mathcal{Z}_l(t, Y_G) + \delta$ *alors* s *appartient à tous les fils de* T *qui satisfont* $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$*.* s *est déplacé dans* Y_S *(voir la figure 3.8(3)).*
2. *si* $\mathcal{Z}_l(s, Y_G) > \alpha$ *alors* s *n'appartient à aucun bi-ensemble satisfaisant* $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ *qui est un fils de* T*.* s *est déplacé dans* N_S *(voir la figure 3.8(4)).*

Les propriétés 3.5 et 3.6 sont définies de la même façon sur les colonnes. La propriété 3.7 explique comment DR-Miner exploite les contraintes monotones et anti-monotones sur $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$.

Propriété 3.7 (Elagage de contraintes (anti)-monotones) *Soit* $\mathcal{L}$ *un treillis tel que* $\mathcal{L} = [(Y_S, Y_G), (Y_S \cup P_S, Y_G \cup P_G)]$*,* $\mathcal{C}_m$ *une contrainte monotone et* $\mathcal{C}_{am}$ *une contrainte anti-monotone par rapport à* $\subseteq$*, alors*

1. *si* $\neg\mathcal{C}_m(Y_S \cup P_S, Y_G \cup P_G)$ *alors aucun bi-ensemble de* $\mathcal{L}$ *ne satisfait* $\mathcal{C}_m$*.*
2. *si* $\neg\mathcal{C}_{am}(Y_S, Y_G)$ *alors aucun bi-ensemble de* $\mathcal{L}$ *ne satisfait* $\mathcal{C}_{am}$*.*

La contrainte qui impose une taille maximale sur les deux dimensions est un exemple de contraintes monotones ($\mathcal{C}_m$*) alors que la contrainte qui impose l'absence de certains éléments dans le bi-ensemble est une contrainte anti-monotone (*$\mathcal{C}_{am}$*).*

Ainsi, les contraintes monotones sont vérifiées sur $Y \cup P$ alors que les contraintes anti-monotones sont vérifiées sur Y.

Les preuves de consistance et de complétude de DR-Miner sont données en annexe D.

Une heuristique importante est utilisée pour accélérer l'énumération : l'élément e utilisée pour l'énumération est $\text{argmax}_t \mathcal{Z}_l(t, Y_S \cup P_S)$ (l'élément qui contient le plus de 0 sur $Y \cup P$). Cette heuristique permet de réduire plus rapidement la taille de P tout en conservant la complétude de l'algorithme.

Extension de bi-ensembles

En pratique, le calcul des DR-bi-sets peut être difficile. Nous formalisons ici une utilisation intéressante de DR-Miner . Nous nous plaçons dans une situation pragmatique où un expert connaît (connaissance personnelle) ou possède (par exemple le

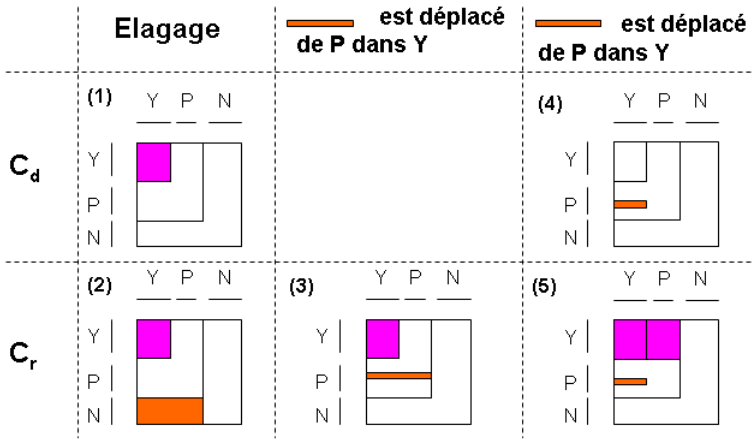


FIG. 3.8 – Vérification des contraintes $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ et $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$

Un contexte booléen **r**, une conjonction de contraintes monotones et anti-monotones $\mathcal{C}$ sur $2^{\mathcal{O}} \times 2^{\mathcal{A}}$, les paramètres α, α', δ *et* δ' et un candidat (Y, P, N).
DR-Miner(Y, P, N)
Si $P \neq (\emptyset, \emptyset)$ alors

1. Soit x un élément (ligne ou colonne) de P
2. $T_1 \leftarrow$ *propagating_constraint*$(Y \cup x, P \setminus x, N)$ (voir la propriété 3.6)
3. si T_1 satisfait les propriétés 3.5 et 3.7 alors **DR-Miner**(T_1)
4. $T_2 \leftarrow$ *propagating_constraint*$(Y, P \setminus x, N \cup x)$ (voir la propriété 3.6)
5. si T_2 satisfait les propriétés 3.5 et 3.7 alors **DR-Miner**(T_2)

sinon Enregistrement de Y

TAB. 3.6 – Le pseudo-code de DR-MINER

résultat d'une extraction de concepts) certains bi-ensembles intéressants. DR-MINER peut alors être utilisé pour étendre ces bi-ensembles permettant d'offrir à l'expert de nouvelles associations potentielles.

Définition 3.8 *L'extension d'un bi-ensemble (X, Y) peut être vue comme l'extraction des DR-bi-sets (X', Y') satisfaisant la contrainte $\mathcal{C}_{XY-sur}(X', Y')$. Or comme $\mathcal{C}_{XY-sur}$ est monotone (voir la définition 2.5),* DR-MINER *peut l'exploiter efficacement.*

3.3.4 Validation expérimentale

Evaluation de la robustesse au bruit sur données synthétiques

Pour montrer la pertinence de $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ dans les données bruitées, nous avons tout d'abord généré des jeux de données synthétiques. Notre but est de montrer que l'extraction des $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ permet de retrouver les concepts introduits dans le jeu de données avant qu'il ne soit bruité. Ainsi, les jeux de données construits sont composés de 4 concepts disjoints comportant chacun 10 éléments sur chaque dimension (matrice booléenne de taille $40 * 40$). Ensuite, un bruit aléatoire uniforme a été introduit dans les données. Nous avons généré 20 jeux de données pour chaque niveau de bruit : 5%, 10%, 15% et 20%. Chaque valeur a 5% (resp. 10%, 15% et 20%) de chance de changer de valeur. Le tableau 3.7 indique le nombre moyen (Moy.) suivi de l'écart-type (E.T.) du nombre de motifs extraits pour chaque niveau de bruit pour $\alpha = \alpha'$ variant de 0 à 3, $\delta = \delta' = 3$ et contenant au moins 4 éléments sur chaque dimension. Dans le tableau 3.7, nous donnons également le nombre moyen de concepts contenant au moins 4 éléments sur chaque dimension pour chaque niveau de bruit.

α		0		1		2		3	
	♯ concepts	Moy.	E.T.	Moy.	E.T.	Moy.	E.T.	Moy.	E.T.
5%	51.63	0	0	0.55	0.51	3	0	4	0
10%	141.53	0	0	0	0	1.6	0.6	2.8	0.41
15%	248.63	0	0	0	0	0	0	0.85	0.49
20%	309.05	0	0	0	0	0	0	1.1	0.45

TAB. 3.7 – Moyenne et écart-type du nombre de motifs extraits (sur 20 essais) en fonction de $\alpha = \alpha'$ et du pourcentage de bruit dans les données ($\delta = \delta' = 3$ et $\mathcal{C}_{44-mis}$)

Lorsqu'il y a 5% de bruit, on retrouve systématiquement les 4 concepts originaux avec $\alpha = \alpha' = 3$. Pour un pourcentage de bruit plus élevé (10% et 15%), seulement certains des concepts originaux sont retrouvés. Lorsque le bruit est trop important (20%), le nombre de motifs extraits est assez variable (l'écart-type vaut 3). Ce qui est très important, c'est que les collections des DR-bi-sets extraites sont des sous-ensembles de la collection des concepts. Ainsi, même si l'on ne parvient pas quand le bruit est trop important, à extraire tous les concepts originaux, les DR-bi-sets permettent de supprimer considérablement (dans cette expérience complètement) le nombre de motifs extraits dû au bruit.

Il faut noter que tous les motifs extraits correspondent aux concepts originaux.

Pour les CBS, nous avons montré qu'en augmentant la taille minimale des motifs dans le même type de jeu de données, les collections extraites se rapprochent de la

collection des concepts formels initiaux. En revanche, les collections extraites sont beaucoup plus grandes que les collections initiales, la démultiplication des motifs due au bruit n'est que partiellement réduite par les CBS et la contrainte de taille.

L'influence des paramètres α et α'

Pour voir l'influence des paramètres α et α' sur $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$, nous avons réalisé plusieurs extractions sur le jeu de données CAMDA [28]. Ce jeu de données montre l'évolution des niveaux d'expression de 3719 gènes (colonnes) de Plasmodium falciparum (responsable de la malaria) durant son invasion des globules rouges. La série temporelle comporte 46 mesures du niveau d'expression des gènes.

Nous avons fixé $\delta = \delta' = 1$ et nous avons fait varier $\alpha = \alpha'$ de 0 à 4. De plus, les motifs doivent satisfaire la contrainte $\mathcal{C}_{\sigma_1\sigma_2-mis}$ avec $\sigma_2 = 3$ et σ_1 qui varie de 19 à 24. Comme la contrainte de fréquence habituellement utilisée lors de l'extraction des ensembles fréquents, la contrainte $\mathcal{C}_{\sigma_1\sigma_2-mis}$ permet de rendre les extractions faisables.

$\alpha = \alpha'$	0	1	2	3	4
$\sigma_1 = 24$	0	4	4	5	5
$\sigma_1 = 23$	9	10	8	9	12
$\sigma_1 = 22$	35	23	22	24	251
$\sigma_1 = 21$	97	68	66	69	-
$\sigma_1 = 20$	241	202	197	213	-
$\sigma_1 = 19$	578	511	513	608	-
TEMPS (S), $\sigma_1 = 23$	0	2	19	171	1185

TAB. 3.8 – Nombre de DR-bi-sets satisfaisant $\mathcal{C}_{\sigma_1 3-mis}$ avec $\delta = \delta' = 1$

Le nombre de motifs extraits pour $\alpha = \alpha'$ de 0 à 2 diminue globalement. Certains motifs sont enrichis et deviennent des sur-ensembles de motifs pour $\alpha = \alpha'$ plus petits. Ensuite, pour $\alpha = \alpha' > 2$, le nombre de motifs extraits tend à augmenter de nouveau. Ceci peut s'expliquer par deux phénomènes :

- Tout d'abord, la taille de certains motifs, initialement non comptabilisés car étant trop petits, augmentent de telle sorte qu'ils satisfont la contrainte de taille.
- Lorsque $\alpha \geq 3$, le nombre d'erreurs acceptées par ligne est supérieur ou égal au nombre de colonnes minimum du motif, ce qui conduit à accepter des motifs pouvant avoir très peu de 1 par ligne. Cela induit une augmentation du nombre de motifs. En pratique, il faut imposer une contrainte de taille minimale sur les deux dimensions nettement supérieure à α et α'.

Lorsque α augmente, l'extraction des DR-bi-sets devient de plus en plus difficile. Nous n'avons pas réussi à extraire les motifs pour $\alpha = \alpha' = 4$ et $\sigma_1 \leq 21$. La figure

3.7 montre le temps d'execution de DR-MINER pour $\sigma_1 = 23$. On peut voir que le temps d'extraction augmente beaucoup avec α et α'.

L'influence des paramètres δ et δ'

Pour montrer l'influence des paramètres δ et δ' sur $\mathcal{M}_{\alpha\alpha'\delta\delta'}$, nous avons réalisé des extractions sur un jeu de données UCI (Internet Advertisements) de dimension 3279×1555. Il ne s'agit pas d'une matrice d'expression mais nous avons cherché un contexte booléen peu dense pour mieux illustrer les variations du nombre de motifs lorsque δ et δ' augmentent.

Pour ces extractions, α et α' sont fixés à 1, δ et δ' varient de 1 à 10 et les motifs doivent satisfaire la contrainte $\mathcal{C}_{\sigma_1\sigma_2-mis}$ avec $\sigma_2 = 0$ et $\sigma_1 \in \{31, 78, 155, 330\}$.

δ	1	2	3	4	5	6	7
$\sigma_1 = 78$	128	17	3	1	1	1	0
$\sigma_1 = 155$	42	6	0	0	0	0	0
$\sigma_1 = 330$	16	5	0	0	0	0	0
TEMPS (s) $\sigma_1 = 78$	841	262	308	326	311	288	272

TAB. 3.9 – Nombre de DR-bi-sets satisfaisant $\mathcal{C}_{\sigma_1 0-mis}$ avec $\alpha = \alpha' = 1$ et $\delta = \delta'$

Les extractions du tableau 3.9 montrent une diminution importante du nombre de motifs extraits au fur et à mesure de l'augmentation de δ et δ'. On peut aussi voir que pour $\sigma_1 = 78$, le temps d'extraction diminue sensiblement entre $\delta = 1$ et les autres paramétrages.

Extraction complète vs. extension de "connaissances"

Il est intéressant de regarder les différences entre des extractions complètes et celles issues de l'extension de concepts formels. D'un point de vue théorique, nous savons que certains DR-bi-sets ne peuvent être obtenus par l'extension de concepts formels.

Exemple 3.3 *Le contexte booléen* **r** *(voir Table 3.10) contient 6 concepts formels :*

$$(\{s_1, s_2, s_4\}, \{g_3\})$$
$$(\{s_1, s_3, s_5\}, \{g_2\})$$
$$(\{s_2, s_3, s_6\}, \{g_1\})$$

$$(\{s_1\}, \{g_2, g_3, g_6\})$$
$$(\{s_2\}, \{g_1, g_3, g_5\})$$
$$(\{s_3\}, \{g_1, g_2, g_4\})$$

Le bi-ensemble $(\{s_1, s_2, s_3\}, \{g_1, g_2, g_3\})$ *satisfait* $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d} \wedge \mathcal{C}_{\delta\delta'\mathbf{r}-p}$ *avec* $\alpha = \alpha' = \delta = \delta' = 1$. *Or aucun des 6 concepts formels n'est un sous-ensemble de ce motif. Ainsi, il ne peut être obtenu par l'extension de concepts formels de* $\mathbf{r}$.

	g_1	g_2	g_3	g_4	g_5	g_6
s_1	0	1	1	0	0	1
s_2	1	0	1	0	1	0
s_3	1	1	0	1	0	0
s_4	0	0	1	0	0	0
s_5	0	1	0	0	0	0
s_6	1	0	0	0	0	0

TAB. 3.10 – Un contexte booléen $\mathbf{r}$

En conséquence, si l'on calcule l'extension des concepts formels pour trouver les DR-bi-sets, alors certains ne seront pas extraits. En calculant les deux collections, on peut mesurer cette différence. Nous avons utilisé deux jeux de données UCI : `Lenses` (matrice 24×12) et `Zoo` (matrice 101×28) [20]. Nous utilisons ces petits jeux de données pour pouvoir réaliser des extractions avec des valeurs de α suffisamment grandes. La table 3.11 montre le nombre de motifs extraits pour la collection complète et la collection issue de l'extension de tous les concepts formels. Pour `Lenses`, les deux

	`Lenses`		`Zoo`	
α	Complète	Extension	Complète	Extension
0	128	128	377	377
1	119	119	342	342
2	98	98	333	327
3	95	95	333	327
4	76	76	416	324
5	73	73	546	-

TAB. 3.11 – Nombre de DR-bi-sets extraits sur `Lenses` et `Zoo` pour les deux stratégies en fonction de α et α'

stratégies calculent la même collection alors que certaines différences apparaissent pour `Zoo`. L'extraction devient infaisable sur `Zoo` avec $\alpha = 5$.

Extension de bi-ensembles : une approche quantitative

Pour étudier l'augmentation de la taille des motifs par rapport à α, nous allons utiliser le jeu de données UCI `Mushroom` (matrice 8 124×128). Nous nous sommes intéressés aux motifs correspondant aux champignons toxiques ayant un seul anneau et une grille étroite. En plus, les motifs extraits doivent contenir au moins 4 champignons et 22 caractéristiques. En utilisant les contraintes précédentes avec D-Miner , 68 concepts formels sont extraits. Ensuite, nous avons calculé les extensions de ces concepts avec $\delta = \delta' = 1$ et $\alpha = \alpha'$ variant de 1 à 4. Pour chaque valeur de α, 68 DR-bi-sets ont été extraits. La taille de la collection extraite est identique à celle des concepts, mais la taille des bi-ensembles obtenus a augmenté. La figure 3.12 montre le pourcentage moyen de champignons et de caractéristiques ajoutés en fonction de α.

$\alpha = \alpha'$	`Champignons`	Caractéristiques
1	0	0
2	0	0
3	15%	0
4	15%	16%

Tab. 3.12 – Pourcentage moyen de champignons et de caractéristiques ajoutés en fonction de α lors de l'extension de 68 concepts formels

Extension de bi-ensembles : une approche qualitative

Nous considérons qu'il est important d'apporter une validation qualitative de la plus-value de nos nouveaux motifs. Nous allons pour cela utiliser de nouveau le jeu de données `camda` [28] (voir section 3.3.4). Parmi les 3 719 gènes, 483 ont une fonction biologique connue. Durant son infection, il est connu que le développement du Plasmodium Falciparum peut être décomposé en 3 phases principales appelées : "ring", "trophozoite" et "shizont". Après avoir encodé les propriétés d'expression de gènes [6], la matrice contient 3 800 concepts. Parmi ces concepts, nous avons sélectionné un concept contenant 8 conditions relatives à la phase "ring" et 4 gènes dont 3 sont connus pour avoir une fonction cytoplasmique. Il est aussi connu que des gènes ayant cette fonction tendent à être sur-exprimés pendant la phase "Ring". Nous avons décidé de regarder l'extension de ce concept particulier. La figure 3.9 montre en bas une description du concept utilisé (8 conditions, 4 gènes et 0% de valeurs 0) et, au dessus, les descriptions de ces extensions. Chaque triplet représente le nombre de conditions, le nombre de gènes et le nombre de 0 contenu dans le bi-ensemble par rapport à sa taille (densité faible relative de 0). Par exemple, le bi-ensemble entouré en gras dans la figure 3.9 contient 9 gènes (5 gènes ajoutés), 11 conditions (3 conditions ajoutées) et sa densité faible relative de 0 est de seulement 7 %. (i.e., 7%

de ces 99 valeurs sont des 0). Les trois conditions qui ont été ajoutées font partie de la phase "ring" et parmi les 5 gènes ajoutés, 4 ont une fonction cytoplasmique. De plus, pour les 6 nouveaux motifs obtenus, 5 des 7 nouveaux gènes ajoutés ont une fonction cytoplasmique et les 8 conditions expérimentales ajoutées appartiennent à la phase "ring". Les extensions du concept considéré apportent donc des informations supplémentaires et montrent la pertinence d'une telle approche.

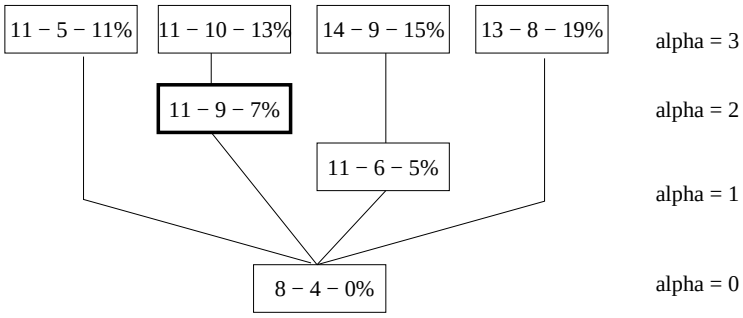


FIG. 3.9 – Extension de concepts formels : chaque triplet représente le nombre de conditions puis de gènes contenus dans le DR-bi-set et enfin sa densité faible relative de 0

Prise en compte de l'impact de la discrétisation

Quand les données initiales sont numériques, par exemple des données d'expression de gènes, il faut encoder les propriétés booléennes. Pour ce faire, différentes techniques ont été proposées et pour chacune d'entre elle, il faut fixer certains paramètres. Nous allons montrer que DR-MINER permet d'améliorer la robustesse des motifs par rapport à la discrétisation. En d'autres termes, l'utilisation des DR-bi-sets permet de réduire l'impact de la technique de discrétisation choisie.

Nous avons étudié deux collections de bi-ensembles extraites de `camda` [28]. Nous avons utilisé la technique de discrétisation MAX - X* MAX pour encoder les propriétés de sur-expression [80] avec X fixé à 70% et à 80%. Ainsi, deux jeux de données ont été obtenus notés $\mathbf{r}_{70}$ et $\mathbf{r}_{80}$. Sur ces deux jeux de données, nous avons regardé les DR-bi-sets contenant au moins 19 conditions et 4 gènes. La figure 3.10 donne le nombre de DR-bi-sets extraits pour différentes valeurs de $\alpha = \alpha'$ avec $\delta = \delta' = 1$. La figure 3.11 donne le pourcentage d'éléments ajoutés.

Le nombre de motifs extraits diminue et la taille des motifs extraits augmente avec $\alpha = \alpha'$ pour les deux discrétisations.

Nous avons calculé une distance, la distance de Manhattan, entre les collections de concepts de $\mathbf{r}_{70}$ et de $\mathbf{r}_{80}$ et entre les collections de DR-bi-sets avec $\alpha = \alpha' = 3$

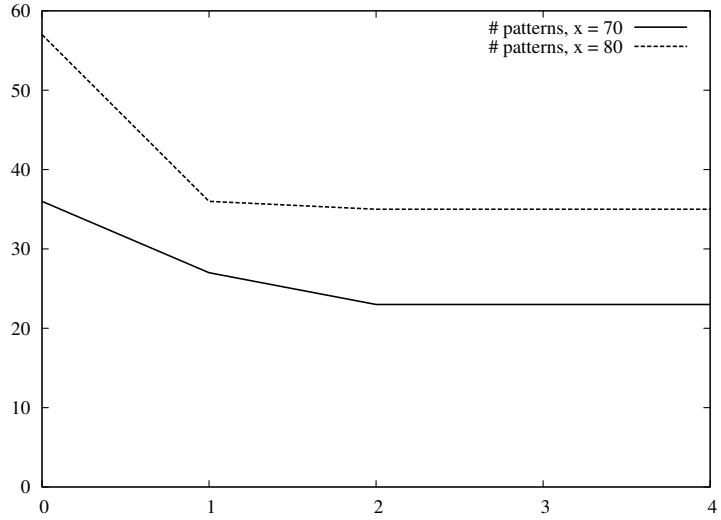


FIG. 3.10 – Nombre de DR-bi-sets extraits pour différentes valeurs de $\alpha = \alpha'$ avec $\delta = \delta' = 1$

de $\mathbf{r}_{70}$ et de $\mathbf{r}_{80}$. Comme par construction $\mathbf{r}_{70} \subseteq \mathbf{r}_{80}$, pour chaque α, nous avons associé à chaque DR-bi-set de $\mathbf{r}_{80}$ le DR-bi-set de $\mathbf{r}_{70}$ le plus proche (par rapport à la distance de Manhattan). On considère que la distance totale est égale à la somme des distances individuelles. On obtient alors finalement que les deux collections de DR-bi-sets (avec $\alpha = \alpha' = 3$) sont plus proches que les deux collections de concepts formels (3.11 contre 3.35). En conséquence, l'impact de la discrétisation peut être réduit (lissé) en utilisant les DR-bi-sets à la place des concepts formels.

3.3.5 Conclusion

Les données réelles sont très souvent bruitées ce qui pose des problèmes pour l'extraction de concepts formels ou plus généralement de motifs locaux ensemblistes. Des méthodes ont été proposées pour pallier ce problème. Elles sont basées soient sur des techniques d'optimisations locales mais sans recouvrement soit sur la recherche d'ensembles d'attributs pour lesquels le support est plus tolérant au bruit. Or, nous souhaitons travailler sur des bi-ensembles satisfaisant certaines propriétés : la complétude des extractions, la dualité entre les objets et les attributs, leur définition très déclarative et finalement le rôle important que jouent les contraintes pour améliorer la pertinence des motifs extraits (et accessoirement la faisabilité des extractions). L'extension des concepts formels à des bi-ensembles denses tolérants au bruit est une approche intéressante pour y parvenir.

Nous avons proposé un nouveau type de motifs appelé DR-bi-sets et un algorithme

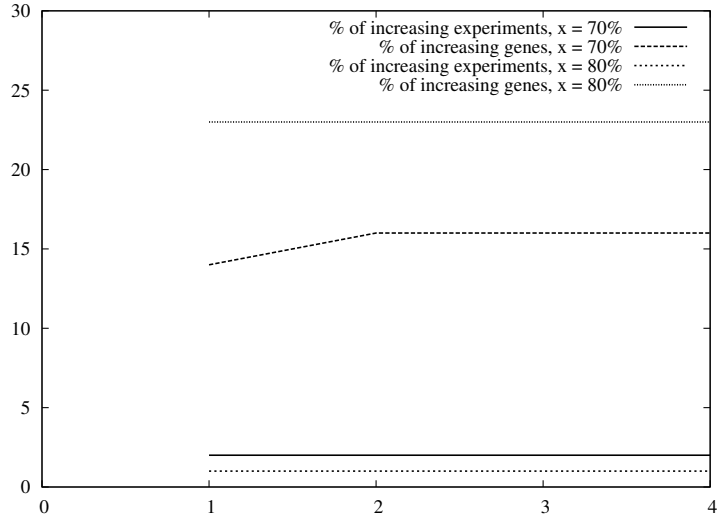


FIG. 3.11 – Pourcentage d'augmentation des motifs pour différentes valeurs de $\alpha = \alpha'$ avec $\delta = \delta' = 1$

pour les calculer. Nous avons opté pour une approche extraction de bi-ensembles sous contraintes pour laquelle les propriétés des motifs sont définies de façon déclarative et exploitées comme des contraintes. L'algorithme proposé permet en plus d'exploiter activement les contraintes monotones et anti-monotones sur $(2^{\mathcal{O}} \times 2^{\mathcal{A}}, \subseteq)$, augmentant considérablement le type de contraintes qui peut être utilisées durant l'extraction. Cet algorithme offre ainsi un cadre très intéressant pour l'extraction de bi-ensembles sous contraintes. L'extraction des DR-bi-sets est difficile en pratique. Pour autant, nous avons proposé une méthode très simple pour exploiter l'algorithme. En effet, il peut être intéressant d'étendre des associations déjà découvertes en ajoutant des lignes et des colonnes contenant certaines exceptions par rapport au motif initial. Cette direction de recherche nous parait très prometteuse dans l'optique d'une assistance à la découverte de connaissances dans des données réelles, que ce soit dans le cadre de la biologie moléculaire ou plus généralement pour le traitement de données transactionnelles bruitées, denses et/ou très corrélées.

Une autre voie intéressante, pour prendre en compte les exceptions, est de regarder non pas comment les spécifier déclarativement à l'intérieur des motifs mais d'étudier les mécanismes permettant de borner, pour un motif donné, le nombre de fois où les contraintes ont été violées [17]. Effectivement, une valeur 0 dans un motif viole la contrainte de 1-rectangle. Réussir à borner le nombre de fois où cette contrainte est violée permet de borner le nombre de valeurs 0 contenu dans un motif. Cette approche peut s'appliquer à d'autres types d'exceptions.

Chapitre 4

Application à l'insulino-résistance

4.1 Introduction

La recherche de l'équipe "Régulation nutritionnelle de l'expression de gènes" de l'UMR INSERM/INRA 1235 s'articule autour de deux axes qui visent (a) à définir les origines moléculaires de l'insulino-résistance musculaire au cours du diabète de type 2 et (b) à caractériser les mécanismes de l'adaptation du muscle et du tissu adipeux aux modifications nutritionnelles et leur implication dans le développement de l'obésité et de ses complications. Cette équipe travaille en particulier sur la compréhension des mécanismes de régulation des gènes en réponse à l'insuline. Pour avancer sur cette problématique, l'équipe dispose de données de puces à ADN qui mesurent la variation d'expression de gènes dans le muscle squelettique humain avant et après injection d'insuline chez des personnes saines. Ces expériences ont été réalisées afin de comprendre la régulation transcriptionnelle de l'insuline chez des personnes saines, pour ensuite découvrir et mieux appréhender les altérations de la régulation chez les personnes insulino-résistantes.

Plus précisément, nous travaillons à partir de données de puces à ADN qui concernent 5 sujets sains. Ces expériences ont été rendues publiques [89]. Elles concernent l'analyse du transcriptome de biopsies musculaires humaines, avant et après 3 heures de clamp hyperinsulinémique-euglycémique. Après ce clamp, les ARNm des cellules musculaires des différents individus ont été "reverse transcrits" en présence de dCTP-Cy5 (la population d'ARNm est marquée avec un fluorochrome rouge) et dCTP-Cy3 (la population d'ARNm est marquée avec un fluorochrome vert) pour former des cDNA marqués. Ces deux populations de cDNA ont été mélangées et hybridées sur des puces à cDNA de Stanford en utilisant le protocole du laboratoire

de P.Brown[1]. Ces lames contiennent 42 557 spots représentant 29 308 UniGene clusters. Après hybridation, les lames ont été scannées avec le scanner GenePix 4000A microarray scanner (Axon Instruments, Union City, Ca). Les images provenant des deux couleurs de fluorescence ont été analysées en utilisant le logiciel Genepix pro 4.0. La fluorescence de chaque spot ainsi que le bruit de fond associé à chaque spot ont été quantifiés. Ces données sont disponibles dans la Stanford Microarray Database[2]. Il faut noter que c'est le bruit de fond local qui est calculé par Genepix. Le bruit de fond local correspond au bruit autour de chaque spot par opposition au bruit de fond global qui est calculé sur toute la lame. Nous avons utilisé la médiane du signal et du bruit de fond car elle permet de lisser les écarts dus à une trop grande hétérogénéité de l'hybridation sur les spots.

À partir de ces données, nous avons donc essayé d'appliquer nos techniques d'extraction de connaissances. Nous sommes dans une situation où l'on cherche à découvrir de nouvelles connaissances biologiques à partir de données brutes de puces à ADN. On se place donc au niveau d'un processus d'extraction de connaissances complet pour lequel il faut pré-traiter les données (normalisation/standardisation et sélection), préparer un contexte d'extraction, extraire les motifs, analyser les motifs extraits et ensuite, pour obtenir le statut de connaissances, vérifier expérimentalement la validité biologique des hypothèses qui ont émergé "in silico".

La figure 4.1 montre le processus d'extraction que nous avons réalisé. D'abord, nous avons effectué des étapes de pré-traitement sur nos données (voir la section 4.2.1) et obtenu une matrice d'expression de gènes de taille $5 * 22069$. A partir de cette matrice, nous avons essayé d'extraire des modules de transcription à l'aide de concepts formels et d'un clustering hiérarchique (voir la section 4.2.2). Les extractions montrent que ces techniques permettent effectivement d'extraire des modules de transcription mais qu'il est très difficile sur un tel jeu de données d'appréhender les mécanismes de régulation des gènes. Nous avons donc ajouté de la connaissance supplémentaire à nos données et en particulier des informations sur des sites de fixation putatifs de facteurs de transcription (voir la section 4.3). Cet enrichissement nous a permis, après une validation expérimentale, de trouver de nouveaux gènes cibles de SREBP1 qui est un facteur de transcription connu comme étant impliqué dans la réponse à l'insuline.

[1] cmgm.stanford.edu/pbrown/protocols/index.html
[2] genome-www5.stanford.edu/MicroArray/SMD/

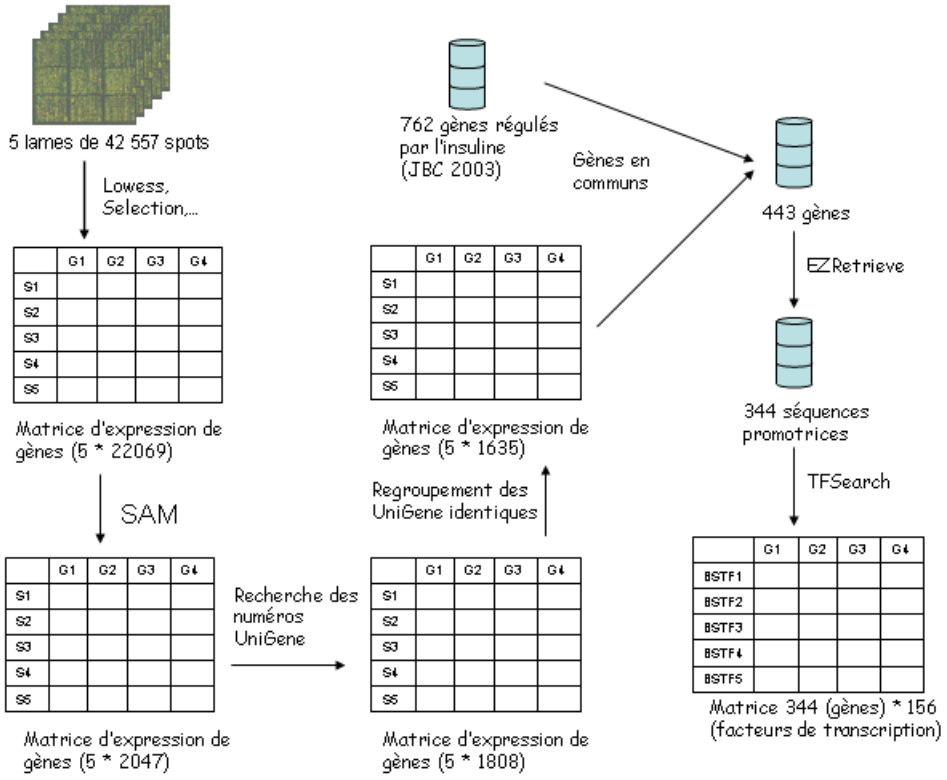


FIG. 4.1 – Notre scénario d'extraction de connaissances

4.2 Premières analyses de nos données

4.2.1 Pré-traitement

Les données brutes obtenues après analyse avec le logiciel Genepix ne peuvent pas être analysées directement (biologiquement ou par des outils informatiques). Il faut d'abord réaliser un pré-traitement de ces données consistant en une normalisation/standardisation des données et en une sélection des gènes à conserver pour l'analyse.

Il faut d'abord supprimer les spots qui ont été détectés comme incorrects par Genepix (spots "flagués") et ceux qui ont un signal rouge ou vert inférieur à 2.5 fois celui du bruit de fond. En dessous de ce seuil, il est difficile de savoir si le signal est véritablement différent du bruit de fond (protocole du laboratoire de P.Brown).

Ensuite, il faut normaliser/standardiser les données. En effet, les puces à ADN génèrent certains biais expérimentaux comme par exemple une intensité de fluorescence différente entre les marqueurs Cy3 et Cy5 à concentration égale et une fluores-

cence non linéaire de Cy3 et Cy5 en fonction de leur concentration.

Les pré-traitements que nous avons effectués sur les puces à ADN utilisent le fait que parmi les milliers de gènes présents sur la puce, peu d'entres eux ont un taux d'expression qui varie significativement entre les deux conditions expérimentales (marquage vert et rouge). En effet, même une substance comme l'insuline ne touche qu'un sous-ensemble réduit des gènes de l'organisme. Les grandes variations observées sont essentiellement dues aux biais de la technologie des puces à ADN et non à une variation réelle du taux d'expression des gènes. Les fluorescences rouges et vertes des spots doivent donc être considérées comme suivant des distributions normales identiques sur les différentes lames. Pour comparer les deux fluorescences, le log_2 des ratios des intensités est à privilégier. En effet, la distribution des ratios n'est pas forcément normale, ce qui pose des problèmes quant à l'application de certaines méthodes statistiques qui ont la normalité comme critère d'application. Il faut mieux privilégier les *log* ratios dont la distribution se rapproche d'une loi normale. Ensuite, les *log* base 2 permettent de faciliter la visualisation des variations d'expression ("fold change"). Une visualisation couramment utilisée pour représenter les fluorescences sur une lame est de dessiner le $log_2(cy3)$ en fonction du $log_2(cy5)$ pour chaque spot (similaire à la représentation MA Plot). La figure 4.2 montre un exemple de tels graphiques sur une de nos lames. En revanche, il faut noter que toute étape de normalisation/standardisation transforme les données et a tendance à écraser les différences et donc à supprimer de l'information. Il faut ainsi essayer de réaliser le moins de traitement possible sur les données tout en essayant de supprimer le plus possible les biais introduits par la technologie des puces à ADN.

Pour normaliser/standardiser nos lames, nous avons appliqué un "Lowess" (Lowess avec itérations et poids) qui est particulièrement bien adapté quand les biais dépendent de l'intensité du signal. L'idée est de rapprocher le nuage de points définis par les couples $(log_2(cy3), log_2(cy5))$ de la droite $y = x$. En effet, ce nuage de points a souvent une forme de "goutte" (expression consacrée) pour les données brutes ce qui montre bien une fluorescence non-linéaire des deux marqueurs. Le lowess qui est une régression linéaire locale permet d'aplatir cette forme de "goutte" et de rendre le nuage de point plus "plat" et plus proche de la droite $y = x$. La figure 4.2 montre à gauche le nuage de points d'une de nos lames (données brutes), au centre une régression linéaire standard sur ces données et à droite le nuage de points après un lowess sur ces même données.

Un deuxième point important est à soulever : faut-il supprimer le bruit de fond local du signal avant la normalisation ? Sur ce point, il n'y a pas véritablement de consensus. Mais le conseil qui est généralement donné est de ne retirer le bruit de fond que s'il n'est pas homogène sur toute la lame. En visualisant les fluorescences de nos données avec ou sans bruit de fond local, il nous est apparu très clairement qu'il était indispensable, sur nos lames, de soustraire aux signaux le bruit de fond local.

Une fois le bruit de fond soustrait et le lowess réalisé, nous avons décidé de ne conserver que les spots qui ont été retenus sur toutes les lames lors de la première étape. Nous avons fait ce choix car nous disposons de peu de lames. Nous avons ainsi privilégié les approches les plus stringentes afin de réduire au maximum les faux positifs même si beaucoup d'informations peuvent être perdues. Le coût et le temps important nécessaire pour analyser ensuite chaque résultat incitent à faire un tel choix.

Ces étapes de normalisation réalisées, nous disposons d'une matrice d'expression de gènes de taille $5 * 22069$.

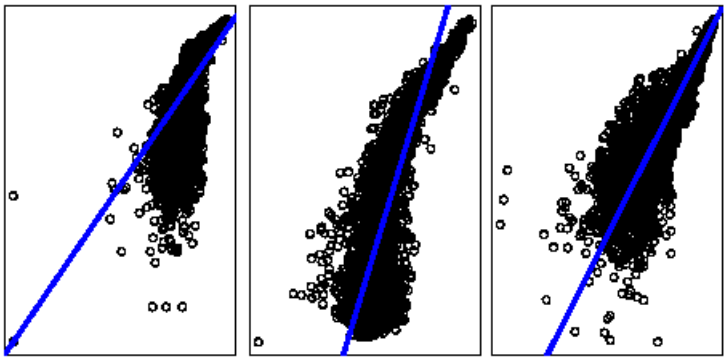


FIG. 4.2 – Représentation ($log_2(cy5)$ en abscisse et $log_2(cy3)$ en ordonnée) des données d'une puce à ADN (gauche), les mêmes données après une régression linéaire (milieu) et avec une régression linéaire locale (droite)

Nous avons développé un programme sous R (http ://www.r-project.org/) permettant de réaliser automatiquement ces étapes de pré-traitement. Ce programme est maintenant un outil de routine utilisé pour le traitement des puces à ADN dans l'unité INRA/INSERM 1235.

4.2.2 Analyse biologique

Les données de puces à ADN dont nous disposons ont déjà été analysées dans [89] d'une façon descriptive (avec un autre pré-traitement). Sur ces données, il a été mis en évidence un groupe de 762 gènes répondant à l'insuline dont 478 ayant un niveau d'expression qui augmente et 284 qui diminue. Ces gènes codent pour des protéines impliquées dans la régulation transcriptionnelle (29% des gènes), le métabolisme (14%), la signalisation intracellulaire (12%), le cytosqueleton et le trafic vésiculaire (9%). Cette étude a permis de mieux comprendre les mécanismes globaux de la réponse transcriptionnelle de l'insuline. A partir de ces informations, nous avons essayé de comprendre plus précisément certains de ces mécanismes. L'objectif est de

découvrir et de mieux appréhender les mécanismes de régulation de la réponse à l'insuline.

Nous avons donc voulu extraire les modules de transcription contenus dans notre matrice d'expression de gènes. Ils permettent, en capturant des groupes de gènes ayant des profils d'expression identiques, d'offrir des hypothèses de travail aux biologistes. En effet, ces associations entre gènes peuvent par exemple indiquer qu'ils partagent une même fonction biologique ou qu'ils interviennent dans un même réseau de régulation. Pour extraire ces modules de transcription, nous avons appliqué un clustering hiérarchique classique et nos techniques d'extraction de concepts formels. Il faut noter qu'une analyse "à la main" de ces données n'est évidemment pas envisageable (matrice $5 * 22069$).

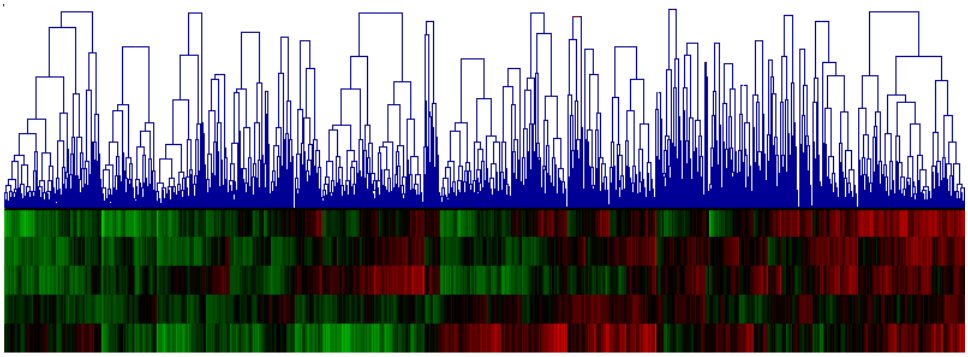


FIG. 4.3 – Clustering hiérarchique sur les données d'expression

La figure 4.3 montre un exemple de clustering hiérarchique classique réalisé sur nos données (Coefficient de corrélation de Pearson), les colonnes représentant les gènes. Ensuite, nous avons utilisé les concepts formels pour extraire les modules de transcription (voir la section 1.5.1). Nous avons décidé d'encoder la propriété "variation d'expression importante". Ainsi, nous avons mis un "1" dans la matrice booléenne d'expression de gènes si sa variation d'expression (en $log_2(cy3, cy5)$) est supérieure en valeur absolue à 0.8. En utilisant D-MINER , nous avons finalement obtenu 32 concepts formels (2^5).

Dans ces deux expériences, les modules de transcription extraits sont composés de centaines de gènes ayant des profils d'expression identiques. Il est donc apparu très clairement qu'avec de tels motifs, il n'est pas possible d'appréhender des mécanismes de régulation précis. Ils apportent au mieux une vision globale des données et ne permettraient qu'une analyse descriptive des mécanismes biologiques sous-jacents. Ce ne sont pas les méthodes que l'on critique mais le fait d'utiliser seulement ce type de données pour obtenir des informations précises sur les mécanismes biologiques. En effet, les deux méthodes utilisées ont déjà montré leur pertinence sur des données

d'expression de gènes dans [28, 79]. En revanche dans ces études, les données traitées sont plus complexes et considèrent différentes conditions expérimentales. En effet, pour interpréter des données d'expression de gènes, il faut pouvoir comparer des données issues de conditions expérimentales différentes. Or, les seules comparaisons dont nous disposons sont celles issues des deux conditions présentes sur chaque lame (marquage rouge et vert), c'est-à-dire les conditions avant et après injection d'insuline. Les différentes lames ne sont que des réplicats qui permettent de mesurer la variabilité intra-individus.

Pour aller plus loin dans l'analyse, il faut alors utiliser d'autres informations ou d'autres données [36]. Nous avons décidé d'enrichir nos données en ajoutant des informations sur les sites de fixation de facteurs de transcription. Ces informations vont nous permettre de mieux appréhender les mécanismes de régulation des gènes régulés par l'insuline. Le chapitre suivant décrit ce travail.

4.3 Découvertes de nouveaux gènes cibles de SREBP1

4.3.1 Introduction

Nous avons vu que les données dont nous disposons ne contiennent pas suffisamment d'informations pour pouvoir étudier précisément les mécanismes de régulation des gènes régulés par l'insuline. Les modules de transcription dans ce type de données ne permettent pas de les appréhender. De façon plus générale, même si les modules de transcription sont très pertinents dans certains jeux de données, leur interprétation est longue et compliquée. En effet, les biologistes doivent utiliser d'autres sources d'informations/connaissances (ontologie des gènes, bibliographie, réseaux de régulations, librairies SAGE...) liées aux gènes et/ou aux conditions expérimentales pour véritablement extraire de nouvelles connaissances biologiques.

Une idée intéressante pour essayer de pallier ce problème est de prendre en compte directement cette connaissance supplémentaire à l'intérieur de notre méthode d'extraction de connaissances. Comme les biologistes de l'UMR 1235 s'intéressent aux mécanismes de régulation liés à l'insuline, nous avons enrichi nos données avec des informations sur les facteurs de transcription, éléments clés de la régulation génique. Les facteurs de transcription se fixent sur des sites de fixation particuliers en amont des gènes (région promotrice) en stimulant ou en inhibant le complexe d'initiation de la transcription. En analysant toutes ces associations entre gènes et facteurs de transcription, il devient alors possible d'appréhender directement les mécanismes de la régulation transcriptionnelle. En réalité, la régulation des gènes se produit très souvent par l'intermédiaire de plusieurs facteurs de transcription appelés alors cofacteurs. Ainsi, s'intéresser aux associations "un gène un site de fixation" n'est pas du tout satisfaisant pour comprendre toute la complexité des mécanismes de régulation.

Il faut alors être capable de découvrir des associations plus pertinentes associant des ensembles de gènes et des ensembles de facteurs de transcription. Nous appellerons ces associations des "modules de régulation". Les concepts formels sont alors de très bons motifs pour capturer les modules de régulation.

Nous proposons de travailler sur de nouveaux contextes d'extraction contenant en colonnes les gènes, en lignes les sites de fixation potentiels des facteurs de transcription et tel que un "1" dans la matrice entre une ligne l et une colonne c représente le fait que le gène correspondant à c contient dans sa région promotrice le site de fixation correspondant à l.

	G_1	G_2	G_3
$TFBS_1$	0	1	1
$TFBS_2$	0	1	1
$TFBS_3$	0	0	1

TAB. 4.1 – Contexte d'extraction avec des gènes en colonne (G_i) et des sites de fixation de facteurs de transcription en ligne ($TFBS_j$)

Exemple. Dans la table 4.1, les gènes G_1 et G_2 ont sur leur séquence promotrice les sites de fixation $TFBS_1$ et $TFBS_2$. $TFBS_1$ et $TFBS_2$ peuvent être suspectés d'être des co-facteurs qui régulent l'expression des gènes G_1 et G_2. L'association ($\{TFBS_1$,$TFBS_2\}$,$\{G_1$,$G_2\}$) correspond exactement à un concept formel dans la table 4.1.

4.3.2 Préparation des données

Pour travailler sur les mécanismes de régulation des gènes régulés par l'insuline, il faut d'abord identifier les spots qui ont une variation d'expression significative entre avant et après le clamp hyperinsulinémique-euglycémique. Les 5 lames forment des réplicats qui permettent de mesurer la variabilité inter-individus. Une méthode utilisée pour quantifier "la variation d'expression significative" est de calculer le ratio de la moyenne des $log_2(cy3/cy5)$ par leur écart-type. Une petite valeur (e.g., 0.01) est généralement ajoutée aux écarts-types afin de ne pas donner trop d'importance aux écarts-types proches de 0. Effectivement, les spots qui sont considérés comme intéressants sont ceux qui ont une variation d'expression moyenne importante mais avec un faible écart-type. Plus le ratio est important et plus le spot est considéré comme ayant une variation significative. En revanche, ce score ne fournit pas d'information sur l'erreur commise en sélectionnant un spot avec un ratio donné. Différentes méthodes existent pour mesurer (approximativement) l'erreur commise. Certaines basées sur Bonferroni permettent de calculer le FWER (Family Wise Error Rate) c'est-à-dire la probabilité que les spots extraits ne contiennent aucun faux positif.

Néanmoins, il vaut mieux privilégier les méthodes qui calculent le FDR (False Discovery Rate) qui est une quantification de l'erreur nettement moins restrictive. Le FDR correspond au pourcentage du nombre de spots qui sont identifiés, par erreur, comme significatifs. Pour estimer ce FDR, nous avons utilisé un logiciel appelé SAM[3][94] (Significance Analysis of Microarrays). Il permet de calculer, pour un FDR donné, le ratio minimal qu'il faut utiliser pour sélectionner les spots. Plus le FDR souhaité est petit et plus le groupe de gènes extrait est petit et moins il contient de faux positifs. Pour notre part, nous avons utilisé un FDR de 1.44 correspondant à des seuils de 0,867 pour les ratios positifs et de -0,964 pour les ratios négatifs. Nous avons pris ces seuils car ils correspondent à une plage de valeur où le FDR est relativement stable.

Finalement, nous avons obtenu 2047 spots dont 1320 ayant une augmentation du niveau d'expression et 727 une diminution après perfusion d'insuline.

Jusqu'à présent, nous avons réalisé les pré-traitements sur les spots en les considérant comme les entités à analyser (identifiées par leur position sur la lame). Or les spots ne correspondent pas exactement à des gènes. Chaque spot correspond à une sonde (séquence d'ADN d'une longueur comprise entre 300 et 1Kb choisie dans la banque de clones du consortium IMAGE). Une sonde correspond à un gène, mais à un gène peut correspondre plusieurs sondes. Ainsi pour pouvoir travailler sur les entités biologiques que sont les gènes, il faut d'abord les identifier à partir de leur numéro Unigene. Ensuite, il faut pour chaque numéro UniGene rassembler les sondes qui leur correspondent et au vu de leurs variations d'expression, décider ou non de conserver le gène.

A partir de nos 2047 spots, nous avons pu identifier 1808 gènes différents en utilisant SOURCE[4]. Ensuite, nous avons regroupé les spots correspondant au même numéro UniGene. Pour chacun de ces groupes, nous avons recalculé le ratio : moyenne des $log_2(cy3/cy5)$ divisée par leur écart-type. Puis, nous n'avons conservé que les gènes qui ont des ratios supérieurs au seuil 0,867 pour les ratio positifs et inférieurs au seuil -0,964 pour les ratios négatifs (les mêmes que ceux qui ont été utilisés sur la matrice totale). Finalement, nous avons obtenu 1635 gènes qui ont une variation d'expression significative entre avant et après injection d'insuline dans le muscle squelettique humain.

Afin de faciliter l'analyse et l'interprétation des résultats, nous avons travaillé sur les gènes en commun entre nos 1635 gènes et ceux cités dans [89] (section 4.2.2). En effet, les biologistes de l'unité INSERM/INRA 1235 connaissent déjà bien ces gènes. De plus, comme les deux analyses ont été réalisées à partir des mêmes données mais avec des pré-traitements différents (régression linéaire simple et Lowess), les gènes communs aux deux analyses doivent contenir moins de faux positifs (en proportion). Nous avons obtenu finalement 443 gènes.

[3]http ://www-stat.stanford.edu/ tibs/SAM/

[4]http ://genome-www5.stanford.edu/cgi-bin/source/sourceBatchSearch

	Hs.101174	Hs.10283	Hs.105656
M00075	0	0	1
M00076	1	1	1
M00271	0	1	1

TAB. 4.2 – Matrice booléenne avec en ligne des sites de fixation, en colonne des gènes et indiquant si un site est présent en amont d'un gène.

Ensuite, nous avons utilisé un logiciel appelé "TFSEARCH" [102][5] qui permet d'identifier les sites de fixation putatifs des facteurs de transcription. Ce logiciel fournit, à partir de séquences d'ADN données par l'utilisateur (au format FASTA), les sous-séquences similaires à des sites de fixation connus contenus dans la base de données TRANSFAC [40]. Il utilise des matrices de poids (TRANSFAC) et un seuil de similarité fixé par l'utilisateur. Nous avons donc recherché dans les séquences d'ADN les régions promotrices de nos 443 gènes. Nous n'avons conservé que les 500 premières paires de bases des régions promotrices à partir du "+1". Nous n'avons pu extraire les séquences promotrices que de 344 gènes parmi les 443 gènes. En effet, certains gènes sont encore mal annotés ou mal séquencés.

Finalement, en utilisant le logiciel TFSEARCH avec un seuil de similarité de 0.8, nous avons obtenu une matrice de taille $344 * 156$ contenant les informations relatives aux sites de fixation pour nos 344 gènes. La table 4.2 montre la forme de la matrice ainsi obtenue. Les lignes correspondent à des sites de fixation, les colonnes à des gènes et un "1" dans la matrice indique qu'un site est potentiellement présent en amont d'un gène ("0" sinon).

Sur cette matrice booléenne, nous avons appliqué un clustering hiérarchique (voir la figure 4.4). Aucun groupe de gènes (en colonne) ou de sites de fixation (en ligne) n'apparaît très clairement. Il n'y a pas dans ces données de motifs globaux, c'est-à-dire de grands ensembles de gènes qui contiennent presque tous les mêmes sites de fixation. Il est alors intéressant de regarder ce que peuvent apporter les motifs locaux, en particulier les concepts formels, dans ces données. Les concepts formels permettent de capturer les associations qui nous intéressent. Par exemple dans la table 4.2, le concept formel $(\{M00076, M00271\}, \{Hs.10283, Hs.105656\})$ indique que les gènes Hs.10283 et Hs.105656 ont les sites de fixation M00076 et M00271 sur leur région promotrice.

[5] http ://siriusb.umdnj.edu :18080/EZRetrieve/index.jsp

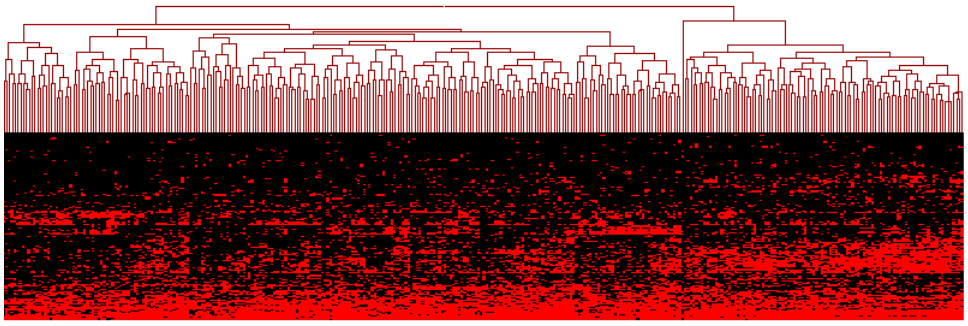


FIG. 4.4 – Clustering hiérarchique sur les données relatives aux sites de fixation de facteurs de transcription

4.3.3 Extraction de concepts formels

Cette nouvelle matrice offre de nouvelles perspectives pour essayer de comprendre le mécanisme de la régulation transcriptionnelle en réponse à l'insuline. En particulier, les concepts formels permettent d'extraire les associations maximales entre les gènes et les sites de fixation de facteurs de transcription qu'ils contiennent. Nous allons donc extraire les concepts formels contenus dans cette matrice. Or, elle contient plus de 5 millions de concepts formels, il est alors inenvisageable de les analyser tous directement.

Nous avons alors centré notre étude sur le facteur de transcription SREBP1 (Sterol-responsive-element binding protein 1) qui est connu pour être impliqué dans la réponse transcriptionnelle de l'insuline [72]. SREBP1 contrôle le flux du métabolisme lipidique dans le foie, le tissu adipeux et le muscle chez les mammifères. Parmi les 344 gènes régulés par l'insuline, 1/3 ont effectivement des sites de fixation potentiels pour SREBP1 (motif SRE) dans leur séquence promotrice ce qui confirme les études précédentes décrivant ce facteur de transcription comme jouant un rôle majeur dans l'action de l'insuline [42].

Nous avons donc voulu extraire tous les modules de régulation qui contiennent le facteur de transcription SREBP1. La base de données de facteurs de transcription TRANSFAC [40] décrit deux motifs de fixation pour SREBP1 M00220 (5'-NATCACGGTGAY-3') et M00221 (5'-KATCACCCCAC-3'). Le site M00220 est très similaire au site de fixation de USF appelé M00217 (5'-NCACGTGN-3') correspondant à une E-box. In vitro, les protéines SREBP forment des dimères qui reconnaissent à la fois le site de la E-box et celui de SRE. Mais in vivo, le site SRE est le seul qui est fonctionnel [60]. Ainsi, nous avons utilisé le site M00221 comme site de fixation de SREBP1. Finalement, nous avons extrait tous les modules de régulation

contenant M00221. En utilisant D-Miner nous avons obtenu 3 690 439 modules de régulation (concepts formels). Cette fois encore, il n'est pas envisageable d'analyser la collection complète.

Parmi tous les sites de fixation de SREBP1, très peu d'entre eux ont effectivement été validés biologiquement. De plus, il est connu que SREBP1 a une faible affinité pour son site de reconnaissance SRE et a besoin d'autres facteurs de transcription pour agir efficacement. En particulier, la présence des facteurs de transcription SP1 (Stimulatory protein) et NF-Y (nuclear factor-Y) améliore l'affinité de SREBP1 pour SRE. Le facteur NF-Y par son action conjointe avec SREBP1 est connu comme agissant sur le métabolisme du cholestérol. De plus, les sites de fixation de SP1 et NF-Y sont souvent proches du site de fixation de SREBP1 comme par exemple pour le gène HK2 (hexokinase 2). Ces informations ont guidé les biologistes de l'UMR 1235 à suspecter que les trois facteurs de transcriptions SREBP1, SP1 et NF-Y ont une action coordonnée pour la réponse à l'insuline. Ainsi, parmi les 344 gènes régulés par l'insuline, nous souhaitions savoir quels étaient les gènes qui pourraient potentiellement être régulés simultanément par les trois facteurs de transcription SREBP1, SP1 et NF-Y.

Ainsi en utilisant D-Miner nous avons extrait les modules de régulation contenant ces trois facteurs. Finalement, 1477 motifs ont été extraits. Nous nous sommes intéressés au module de régulation contenant le plus de gènes. Ce module est composé de 6 sites de fixation de facteurs de transcription : GATA-1 (M00075), GATA-2 (M00076), AML-1a/Runx1 (M00271) et évidemment SREBP1/NF-Y/SP1. Les facteurs de transcription AML-1a/Runx1, GATA-1 et GATA-2 ne sont pas exprimés dans le muscle. Ainsi, ils ne peuvent pas agir avec SREBP1 dans le muscle, mais leur présence dans le même module que SREBP1 indique qu'ils pourraient agir comme tel dans d'autres tissus. Le module est composé de 13 gènes : SPOP, SF1 (transport et processing de l'ARN) MORF4L2 (régulation de la transcription) MAPRE1, SDC1 (cytosquelette), VPS29, ARF4 (trafic vésiculaire et réseau trans-golgien), ABCA7 (transporteurs), PGRMC2 (récepteur membranaire), FEM1B (induction d'apoptosis), HK2 (glycolyse), HIG1 et CRYBA4 (fonctions inconnues). Il est intéressant de noter la présence de HK2 dans ce module. Son expression est induite par l'insuline et il a été montré récemment que SREBP1 régule sa transcription (travail de thèse de Yvan Gosmain dans l'UMR 1235). Ainsi, les 12 autres gènes sont fortement suspectés d'être de nouvelles cibles de SREBP1 dans le muscle humain. Cette hypothèse a été vérifiée en utilisant la technique de chromatin immunoprecipitation (ChIP [71]) par Emmanuelle Meugnier (étudiante en thèse dans l'unité UMR 1235). Cette technique permet de montrer si effectivement un facteur de transcription s'accroche in vivo sur la région promotrice d'un gène.

4.3.4 Validation expérimentale par ChIP

Deux protéines différentes produites par SREBP1 ont été décrites : SREBP-1a et SREBP-1c. Ces facteurs n'ont pas la même efficacité transcriptionnelle. SREBP-1a est un activateur plus puissant de la transcription que SREBP-1c. Comme les deux isoformes de SREBP1 possèdent le même site de fixation, notre analyse ne permet pas de discriminer le rôle de ces deux facteurs. De plus, il n'y a pas d'anti-corps commercialement disponibles pour SREBP1a et SREBP1c. La validation biologique a donc été réalisée avec un anti-corps anti-SREBP1 total. Les résultats montrent que SREBP1 se fixe sur la région promotrice de ARF4, SPOP, FEM1b, VSP29, HIG1, PGRMC2, SDC1 et SF1 mais pas sur celle de CRYBA4 et ABCA7. Les expériences n'ont pu être réalisées pour les gènes MORF4L2 et MAPRE1. Il y a deux explications possibles pour l'absence de fixation sur CRYBA4 et ABCA7. D'abord, la protéine encodée par ABCA7 est membre de la super-famille des transporteurs de ABC (ATP-binding cassette) et est très poche de ABCA1. Il a été démontré que SREBP2 peut inhiber l'activité de ABAC1 dans les cellules vasculaires endothéliales. Le fait que SREBP2 plutôt que SREBP1 régule ABCA7 n'est pas exclu. Ensuite, il a été démontré que la protéine SREBP1 forme un complexe avec une très forte affinité quand les autres facteurs de transcription sont présents et leur espacement très faible. Par exemple, la régulation transcriptionnelle du gène de la farnesyl diphosphate synthase est dépendante d'une interaction synergique entre SREBP-1 et NF-Y [17]. Les sites de fixation de SREBP1 sur CRYBA4 et ABCA7 sont très éloignés de ceux de NF-Y et de SP1 (plus 50bp). De plus, les deux régions promotrices de CRYBA4 et ABCA7 sont relativement pauvres en sites de fixation de facteurs de transcription qui pourraient interagir avec SREBP1, contrairement aux autres promoteurs. La figure 4.5 montre la position des sites de fixation des facteurs de transcription USF (M00217), SREBP (M00220 et M00221), NF-Y (M00209) et SP1 (M00008) pour les 13 gènes.

La séquence promotrice du gène UBE2V2 a été utilisée comme contrôle négatif pour cette expérience car il possède des sites de fixation pour SREBP1 et SP1 mais pas pour NF-Y. L'expérience montre que SREBP1 ne se fixe pas sur la région promotrice de UBE2V2 ce qui semble indiquer que c'est véritablement la présence des deux autres facteurs de transcription qui permettrait à SREBP1 de s'accrocher.

4.3.5 Autres modules

Parmi les 1477 modules de régulation extraits, 327 modules contiennent des sites de fixation de type CREB (M00039). CREB a été initialement identifié comme un élément de régulation en réponse à l'augmentation du niveau de cAMP. Ensuite, c'est une famille entière de facteurs de transcription (CREB/ATF) qui a été identifiée comme activant la transcription en agissant directement sur les composants

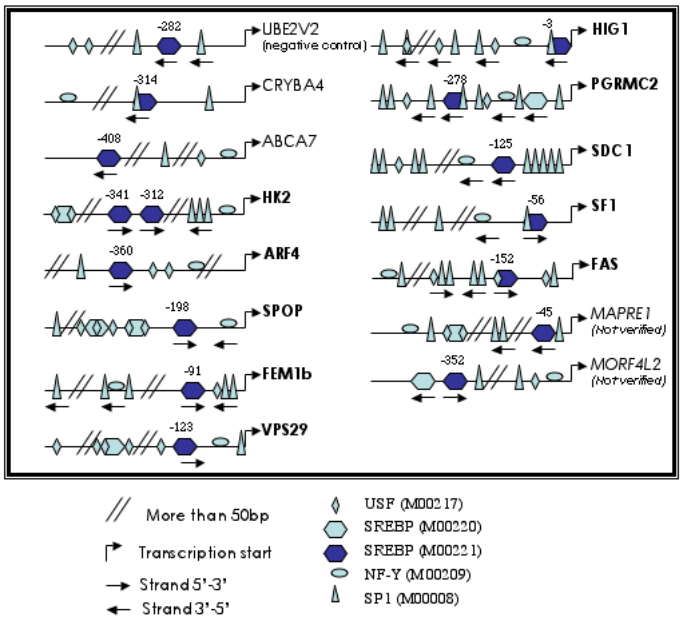


FIG. 4.5 – Positions des sites de fixation pour les gènes étudiés

chromatines. Il a été montré que CREB, un membre de cette famille, et NF-Y sont tous les deux recrutés sur le promoteur de la HMG-CoA reductase par SREBP1. Les 327 modules contiennent 9 des 13 gènes identifiés précédemment (ABCA-7, HK2, MAPRE1, MORF4L2, ARF4, SF1, VPS29, CRYBA4 et HIG1). Par exemple, un module contient les 9 gènes et 10 sites de fixation (GATA-1, GATA-2, AML1a, SREBP1, SP1, NF-Y, CREB (M00039), v-Myb (M00227), Cdx-A (M00100), Nkx-2 (M00240)). Les trois facteurs de transcription v-Myb, Cdx-A et Nkx-2 ne sont pas présents dans le muscle mais pourraient agir avec SREBP1 dans d'autres tissus.

Parmi les 1477 modules de régulation extraits, 57 sont composés de gènes qui ont la E-box dans leur région promotrice. Par exemple, un module est composé des gènes SPOP, HK2, MAPRE1, MORF4L2, VSP29, PGRMC2 et des sites GATA-1, GATA-2, AML-1a, Cdx-A, Nkx-2, SP1, NF-Y, SREBP1, CRE-BP (M00041) et USF (M00217). Il est intéressant de noter de nouveau la présence d'un gène (CRE-BP) associé à la réponse à l'augmentation du niveau de cAMP, ce qui renforce l'hypothèse que cette famille de facteurs de transcription agit avec SREBP1. Il a été montré que les USFs (upstream regulatory factors) sont aussi des co-facteurs de SREBP1. En effet, le site de fixation USF M00217 (5'-NCACGTGN-3') est similaire à celui de la E-box et habituellement la chevauche.

4.4 Conclusion et discussion

Nous avons travaillé sur les mécanismes de régulation des gènes en réponse à l'insuline chez les êtres humains. Pour cela, il a fallu réaliser l'ensemble du processus d'extraction de connaissances : pré-traitement des données (normalisation/standardisation et sélection), préparation du contexte d'extraction, extraction de motifs, analyse des motifs extraits et ensuite pour obtenir le statut de connaissances vérifier expérimentalement les nouvelles informations extraites. Les données de puces à ADN dont nous disposons permettent de découvrir les gènes qui sont régulés par l'insuline. En revanche, elles ne contiennent pas suffisamment d'information pour pouvoir décrypter les mécanismes de régulation sous-jacents. Nous avons alors proposé d'utiliser des informations sur les sites de fixation des facteurs de transcription pour les gènes régulés par l'insuline. Dans ce nouveau contexte d'extraction, il est possible en utilisant les concepts formels, d'extraire les ensembles de gènes et les ensembles de sites de fixation tels que tous ces gènes ont simultanément tous ces sites dans leur région promotrice. Ces informations sont alors très précieuses pour comprendre les mécanismes de régulation des gènes. Nous nous sommes intéressé plus particulièrement à une combinaison particulière de trois facteurs de transcription : SREBP1, NF-Y et SP1 qui sont connus pour être impliqués dans la réponse à l'insuline. En utilisant D-Miner , nous avons extrait les concepts formels contenant ces trois sites de fixation et nous avons analysé un motif particulier contenant 13 gènes et 6 sites. Une validation biologique par ChIP à montré que 8 de ces gènes ont effectivement un site de fixation actif pour SREBP1. Ces gènes sont fortement suspectés d'être impliqués dans la réponse à l'insuline. Il est bien sûr envisagé par la suite d'utiliser DR-Miner pour extraire des motifs plus tolérants au bruit afin d'améliorer la pertinence des motifs extraits. Il est, par exemple, possible d'essayer d'étendre le concept formel composé des 13 gènes et des 6 sites en admettant certaines exceptions. Au delà de cette analyse, la méthode que nous proposons permet d'appréhender plus facilement les mécanismes de régulation des gènes et peut être utilisée sur d'autres problématiques que celle sur l'insulino-résistance.

Conclusion et perspectives

Dans ce travail, nous nous sommes intéressé à l'extraction de connaissances dans des données transcriptomiques (données d'expression de gènes) avec une application à l'étude de l'insulino-résistance.

L'état de l'art au début de notre travail pouvait se résumer en (a) de multiples techniques d'analyse statistique et relativement peu instrumentées pour étudier de petits ensembles de gènes et/ou de situations expérimentales, (b) de nombreuses propositions basées sur les algorithmes de clustering et donc l'extraction de motifs globaux par des techniques heuristiques, et enfin (c) quelques travaux très préliminaires sur les possibilités des motifs locaux comme les règles d'association.

Nous avons décidé d'étudier de façon approfondie des processus d'extraction de connaissances à partir de matrices booléennes (e.g., codant des propriétés d'expression de gènes ou des associations entre gènes et facteurs de transcription). Ces découvertes de connaissances s'appuient alors sur divers types de motifs locaux comme les concepts formels ou de nouveaux types de motifs plus robustes au bruit. Tout en étant guidé par notre application à l'étude de l'insulino-résistance, nous avons développé de nouvelles techniques d'extraction de motifs. Nous avons tenu à développer des algorithmes justes et complets, une hypothèse de travail qui gagne du terrain chez les spécialistes de la découverte de motifs locaux.

Les principaux résultats obtenus concernent donc trois axes de recherche.

– Extraction de motifs sous contraintes et contribution au cadre des bases de données inductives dans le cadre de contextes réalistes.
  Nous avons d'abord travaillé sur l'extraction de concepts formels sous contraintes. Notre première contribution, en collaboration avec le GREYC (Université de Caen), concernait la faisabilité des extractions dans le cas de données contenant beaucoup de colonnes mais peu de lignes. Notre seconde contribution a concerné un nouvel algorithme d'extraction de concepts formels qui exploite efficacement de nouveaux types de contraintes. L'idée était d'améliorer la pertinence a priori des motifs extraits puisque l'utilisateur peut spécifier une conjonction de contraintes sur les concepts formels désirés et avoir la garantie que, lorsque l'extraction est faisable, tous les concepts formels répondant à sa contrainte ont

été calculés. Nous avons aussi travaillé sur le problème des données bruitées. Les concepts formels capturent des associations très fortes entre des ensembles d'attributs et des ensembles d'objets. Dans des données réelles, cette association est trop forte et induit à la fois une diminution de la pertinence des concepts formels extraits et une augmentation de leur nombre. Nous avons donc proposé deux nouveaux types de motifs permettant de capturer des associations fortes mais avec une certaine tolérance aux exceptions.

– Mise en oeuvre de processus d'extraction de connaissances à partir de données transcriptomiques.
  Les scénarios d'extraction de connaissances depuis des données transcriptomiques (typiquement des données d'expression de gènes produites par des techniques à haut débit comme les puces ADN ou la technique SAGE) et basées sur des contextes d'extraction booléens sont aujourd'hui très crédibles car ils ont déjà été utilisés avec succès sur plusieurs problématiques biologiques précises. Les problèmes posés par l'encodage de propriétés booléennes sont mieux maîtrisés, les mérites (et les défauts) des motifs locaux (vs. motifs globaux) sont mieux compris, de multiples extracteurs de motifs ont été implémentés et sont aujourd'hui intégrés dans une plate-forme logicielle (Bio++). Celle-ci a déjà été utilisée ponctuellement pour l'enseignement et devrait bientôt être rendue publique pour la communauté des bioinformaticiens.
– Découvertes de connaissances sur l'insulino-résistance
  Nous avons travaillé sur des données de puces à ADN produites par l'unité INSERM/INRA 1235 pour mieux comprendre les mécanismes de régulation liés à la réponse à l'insuline chez l'être humain. Concrètement, cela veut dire que nous avons participé au travail de pré-traitement de données puces à ADN réelles et à la mise en oeuvre de plusieurs techniques d'analyse basées sur les motifs. Ce travail a permis de découvrir de nouveaux gènes cibles de SREBP1 qui est un facteur de transcription connu pour être impliqué dans la réponse à l'insuline. Le résultat a été validé biologiquement par INSERM/INRA 1235.

Nous reprenons maintenant ces différents axes en dégageant les perspectives de notre travail.

Extraction sous contraintes

Alors que les très nombreux travaux sur l'extraction de motifs sous contraintes portaient sur des motifs mono-dimensionnels (i.e., essentiellement les ensembles d'attributs et les motifs séquentiels), nous avons étudié plusieurs approches pour l'extraction de bi-ensembles sous contraintes, i.e., des motifs bi-dimensionnels comme les concepts formels ou les DR-bi-sets. Ces travaux ont permis d'engager des recherches prometteuses

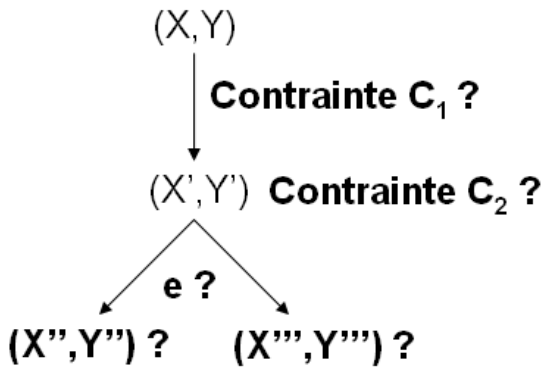
D'abord, il est très intéressant de travailler sur les deux dimensions simultanément.

Dans les problématiques réelles, les experts utilisent les éléments contenus sur les deux dimensions pour interpréter les motifs. Par exemple, un motif qui contient des gènes qui sont simultanément sur-exprimés dans de nombreuses conditions expérimentales relatives à des patients diabétiques et qui ne le sont pas dans les conditions relatives à des personnes saines, prend tout son sens du moment que l'on considère les deux dimensions. Nous avons aussi montré qu'il était important de ne privilégier aucune dimension pour l'énumération et la relation de spécialisation utilisée dans les algorithmes d'extraction. Ce choix incite à énumérer indifféremment des objets et/ou des attributs et à utiliser les même contraintes sur les deux dimensions. Ce point de vue nous place alors naturellement dans le cadre générique de l'extraction de bi-ensembles sous contraintes dans lequel les ensembles d'attributs (et l'ensemble des objets qui les portent), les concepts formels ou encore les DR-bi-sets ne sont que des instances particulières.

Ensuite, il faut s'intéresser au rapport entre le coût de la vérification d'une contrainte et son impact dans la réduction de l'espace de recherche. Les contraintes qui nécessitent beaucoup de calculs pour être vérifiées et qui ne sont pas très stringentes ne devraient être utilisées que partiellement au cours de l'extraction. Par exemple, dans D-Miner , la contrainte qui impose que les bi-ensembles extraits soient des 1-rectangles est activement exploitée. En revanche celle qui impose la maximalité n'est utilisée que pour vérifier la consistance de l'espace de recherche. Ce choix a été fait car cette dernière est coûteuse à vérifier et ne permet pas de réduire significativement l'espace de recherche.

Ces différents points permettent de mettre en avant quatre degrés de liberté des algorithmes complets d'extraction de bi-ensembles sous contraintes (voir la figure suivante) :

- l'élément à énumérer (e).
- la relation de spécialisation (les fils (X'', Y'') et (X''', Y''')).
- la contrainte à utiliser pour réduire l'espace de recherche (C_1).
- la contrainte à utiliser pour vérifier la consistance de l'espace de recherche (C_2).



La question est maintenant : "quel choix faire pour ces quatre degrés de liberté en fonction du jeu de données à traiter (densité, dimensions, ...), des motifs souhaités et des contraintes utilisées ?". Il faut noter qu'une telle question peut aussi se poser pour chaque candidat durant l'extraction, induisant alors une approche adaptative pour décider de la gestion des contraintes. C'est un problème très compliqué et, à notre connaissance, l'on ne dispose pas encore de techniques pour caractériser les données et l'impact des contraintes (connaissance a priori de la sélectivité). Il est donc très difficile de mettre au point des méthodes qui réaliseraient automatiquement ces choix. En revanche, une approche expérimentale pourrait, dans un premier temps, nous faire avancer considérablement sur cette voie. Par exemple, l'algorithme DR-Miner offre un cadre suffisamment générique pour pouvoir réaliser une telle tâche.

Données transcriptomiques

Dans la section 4, nous avons présenté le travail que nous avons réalisé autour de l'insulino-résistance chez l'être humain. Pour pouvoir extraire des informations pertinentes, nous avons utilisé de la connaissance supplémentaire liée aux sites de fixation des facteurs de transcription. Mais le cadre de l'extraction de bi-ensembles sous contraintes dans des données issues du transcriptome permet de répondre à beaucoup d'autres questions ou objectifs d'analyse. En effet, beaucoup d'informations sont disponibles comme les ontologies, la bibliographie, des réseaux de régulations partiels, des librairies SAGE, etc. Dans la mesure où elles sont exprimables sous la forme d'une relation booléenne, ces informations permettent de produire de nouveaux contextes d'extraction sur lesquels on peut poser des requêtes très différentes. Par exemple, il peut être intéressant de travailler sur une matrice contenant des informations sur les variations d'expression de gènes chez des personnes saines et des patients diabétiques en réponse à l'insuline, des informations cliniques sur ces personnes, les sites de fixation de facteurs de transcription et des ontologies GO. La table 4.3 est un exemple de matrice contenant ce type d'information. Les G_l représentent des gènes, les S_i des conditions expérimentales, les $TFBS_j$ des sites de fixation, les $TermeGo_k$ des termes GO et "Age" et "Diabétique" des informations cliniques sur les personnes concernées par les conditions expérimentales. Il faut noter que la matrice ainsi générée est en fait issue de la fusion de différentes matrices dont les lignes, les colonnes et la relation liant ces éléments sont différentes :

- $(\{S_1S_2S_3...S_i\}, \{G_1G_2G_3...G_l\})$ est une matrice d'expression de gènes.
- $(\{TBFS_1TBFS_2...TBFS_j\}, \{G_1G_2G_3...G_l\})$ contient des informations sur les sites de fixation.
- $(\{TermeGo_1TermeGO_2...TermeGO_k\}, \{G_1G_2G_3...G_l\})$ contient des termes GO des gènes.
- $(\{S_1S_2S_3...S_i\}, \{Age, Diabetique\})$ contient des informations cliniques sur les personnes associées aux conditions.

	G_1	G_2	G_3	G_l	Age	Diabétique
S_1	1	1	0	...	1	1
S_2	1	0	0	...	0	1
S_3	1	1	1	...	0	0
S_i	...	...	...	...	...	...
$TFBS_1$	0	0	1	...	$\underline{1}$	$\underline{1}$
$TFBS_2$	0	0	1	...	$\underline{1}$	$\underline{1}$
$TFBS_j$	...	...	...	...	$\underline{1}$	$\underline{1}$
$TermeGo_1$	0	1	1	...	$\underline{1}$	$\underline{1}$
$TermeGo_2$	1	1	1	...	$\underline{1}$	$\underline{1}$
$TermeGo_k$	...	...	...	...	$\underline{1}$	$\underline{1}$

TAB. 4.3 – Exemple d'une matrice booléenne contenant de nombreuses informations biologiques.

Notons que dans cette matrice, certaines associations entre lignes et colonnes n'ont pas de sens (valeurs sous-lignées) comme, par exemple, l'association entre un terme GO et l'âge de la personne associée à une condition. Pour pouvoir extraire des bi-ensembles dans cette matrice, il faut alors pour ces associations mettre des valeurs "1".

Dans cette matrice, on peut par exemple s'intéresser à tous les ensembles de gènes de taille supérieure à 4, associés au terme GO $TermeGo_k$, ayant au moins trois sites de fixation de facteurs de transcription en commun et variant significativement chez au moins 3 sujets diabétiques. Cette requête peut s'exprimer comme la recherche de tous les concepts formels (X, Y) satisfaisant :

$\mathcal{C}_{0\,4-mis}(X, Y \cap \{G_1G_2G_3...G_l\}) \wedge$
$\mathcal{C}_{3\,0-mis}(X \cap \{TBFS_1TBFS_2...TBFS_j\}, Y) \wedge$
$\mathcal{C}_{3\,0-mis}(X \cap \{S_1S_2S_3...S_i\}, Y) \wedge$
$\mathcal{C}_{\{TermeGo_k\}\{Diabetique\}-sur}(X, Y)$

Les algorithmes D-MINER et DR-MINER peuvent exploiter ces contraintes. On peut aussi produire une matrice booléenne contenant toutes les données d'expression de gènes de SMD (Stanford Microarray Database), les sites de fixation et les annotations GO du génome entier, etc. Cette matrice offre alors un contexte d'extraction générique qui peut être interrogé par différents biologistes en fonction de leurs problématiques précises. Le cadre de l'extraction complète de bi-ensembles sous contraintes devient alors un mécanisme puissant pour que les utilisateurs formulent des requêtes inductives caractérisant les propriétés recherchées sans avoir à se préoccuper des mécanismes d'évaluation de telles requêtes.

Le développement d'un extracteur générique de bi-ensembles sous contraintes et la fabrication d'un véritable entrepôt de données contenant de nombreuses informations sur le transcriptome permettrait d'offrir aux biologistes un système d'interrogation souple pour assister la découverte de connaissances sur le transcriptome.

Annexe A

Validation expérimentale de la transposition

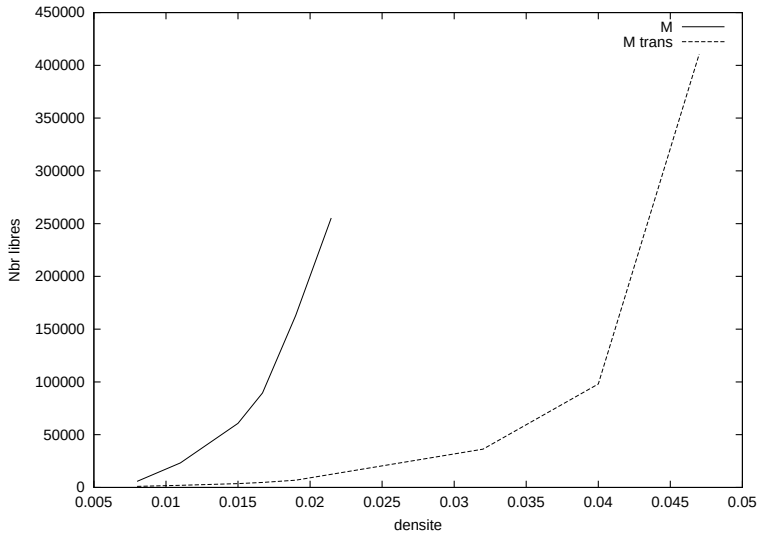


FIG. A.1 – Nombre d'ensembles libres sur les données "drosophile"

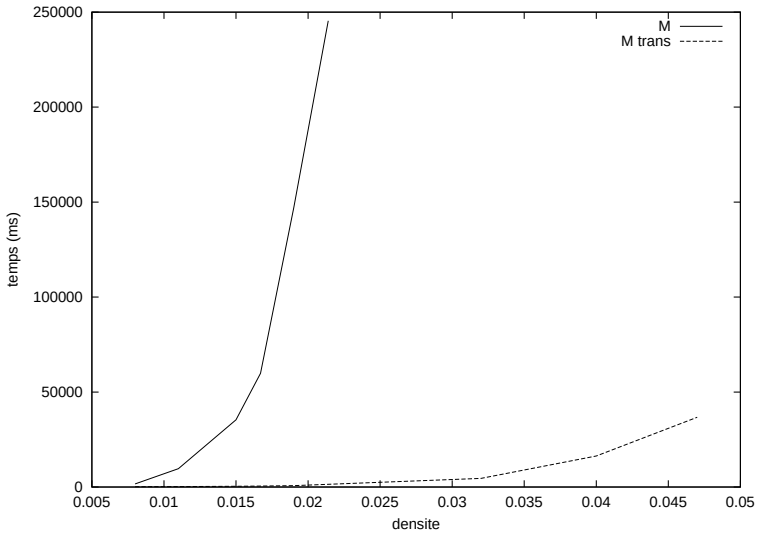


FIG. A.2 – Temps d'extraction en fonction de la densité sur les données "drosophile"

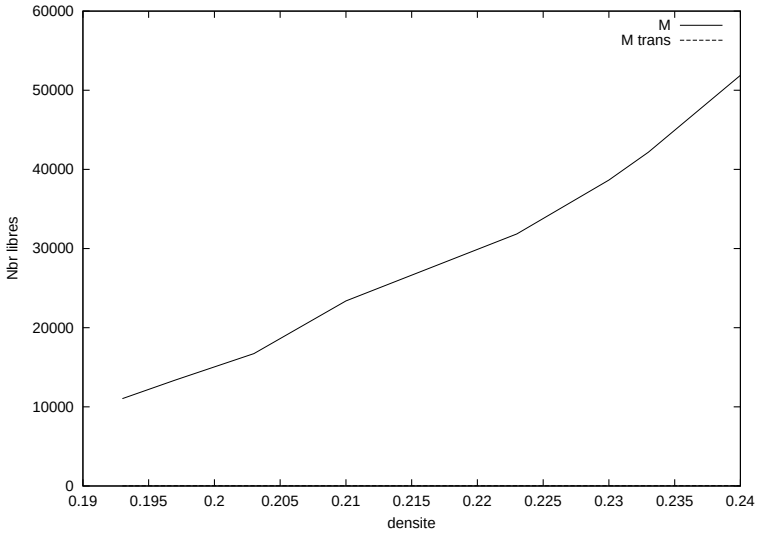


FIG. A.3 – Nombre d'ensembles libres sur les données "humaines"

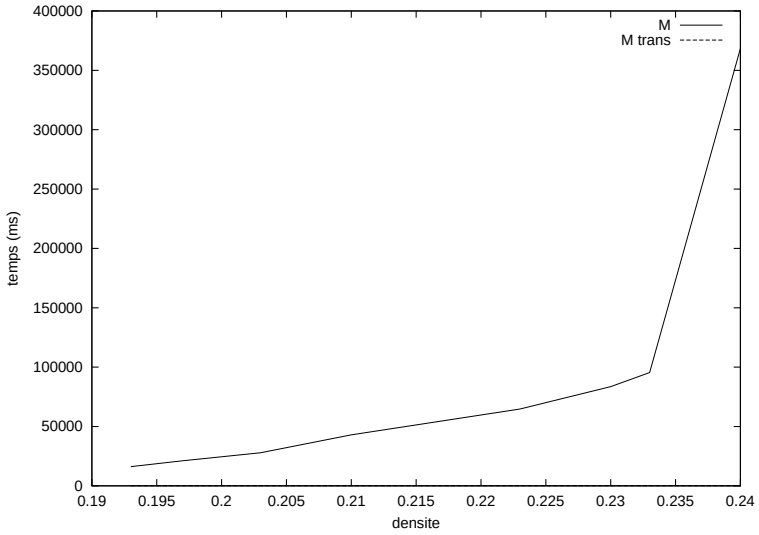


FIG. A.4 – Temps d'extraction sur les données humaines

Annexe B

Preuves de D-Miner

Nous allons d'abord montrer que toutes les feuilles d'un arbre d'exécution de D-Miner correspondent à des 1-rectangles (voir la section B). Ensuite que toutes les feuilles sont différentes (voir la section B). Puis que l'ensemble des feuilles noté L est égal à l'ensemble des concepts formels noté C.

Toutes les feuilles sont des 1-rectangles

Soit (L_X, L_Y) une feuille de l'arbre d'énumération. Nous allons montrer que $\forall i \in L_X$, $\forall j \in L_Y$, $(i,j) \in \mathbf{r}$. Nous allons le montrer par l'absurde et supposer qu'il existe $i \in L_X$ et $j \in L_Y$ tel que $(i,j) \notin \mathbf{r}$. Ainsi, à cause de la définition de H, il existe $(A,B) \in H$ tel que $i \in A$ et $j \in B$. Or chaque élément de H est utilisé sur le chemin de la racine à la feuille. Soit (A_X, A_Y) l'ancêtre de (L_X, L_Y) découpé par (A,B). La relation d'ordre implique que $L_X \subseteq A_X$ et que $L_Y \subseteq A_Y$ et donc que $i \in A_X$ and $j \in A_Y$. Les fils de (A_X, A_Y), s'ils existent, sont $(A_X \setminus A, A_Y)$ et $(A_X, A_Y \setminus B)$. Ainsi, on a $L_X \subseteq A_X \setminus A \subseteq A_X \setminus \{i\}$ et $L_Y \subseteq A_Y \setminus B \subseteq A_Y \setminus \{j\}$. Ce qui est en contradiction avec les hypothèses, on a ainsi :

$$\forall i \in L_X,\ \forall j \in L_Y,\ (i,j) \in \mathbf{r}$$

Toutes les feuilles sont différentes

Soit (L_{X_1}, L_{Y_1}) et (L_{X_2}, L_{Y_2}) deux feuilles de l'arbre d'exécution de D-Miner, (A_X, A_Y) le plus petit ancêtre commun de ces feuilles et (A,B) l'élément de H utilisé pour le découper. On suppose, que (L_{X_1}, L_{Y_1}) est le fils gauche de l'ancêtre commun et que (L_{X_2}, L_{Y_2}) est son fils droit. A cause de l'algorithme, tous les descendants du fils gauche de (A_X, A_Y) ne contiennent pas l'élément A et tous les descendants du fils droit de (A_X, A_Y) contiennent A. Finalement, les deux feuilles sont différentes.

$C \subseteq L$**(Complétude)**

Nous allons démontrer que pour chaque concept (C_X, C_Y), il existe un chemin de la racine $(\mathcal{O}, \mathcal{P})$ au concept :

$$(\mathcal{O}, \mathcal{P}),\ (X_1, Y_1),\ \cdots,\ (X_{n-1}, Y_{n-1}),\ (C_X, C_Y)$$

– D'abord, on montre récursivement que le chemin est tel que :

$$(X_0, Y_0),\ (X_1, Y_1),\ \cdots,\ (X_{n-1}, Y_{n-1}),\ (L_X, L_Y)$$
$$\text{avec } (L_X, L_Y) \supseteq (C_X, C_Y) \text{ et } (\mathcal{O}, \mathcal{P}) = (X_0, Y_0)$$

Nous notons $H_{L_k}(X_k, Y_k)$ l'ensemble des éléments de H qui gênèrent des fils gauches pour le chemin de (X_0, Y_0) à (X_k, Y_k). L'hypothèse de récurrence est la suivante :
(1) $(C_X, C_Y) \subseteq (X_k, Y_k)$
(2) $\forall (A, B) \in H_{L_k},\ C_Y \cap B \neq \emptyset$
Preuve :
Base Par définition des concepts formels, chaque concept (C_X, C_Y) est inclus dans le nœud racine, c'est à dire que $C_X \subseteq X_0$ et $C_Y \subseteq Y_0$. L'hypothèse (1) est donc vérifié au rang 0. De plus, $H_{L_0} = \emptyset$ et donc l'hypothèse (2) est aussi vérifiée au rang 0.
Récursion Soient les hypothèses (1) et (2) vérifiées au rang $k-1$, nous allons montrer qu'elles sont toujours vraies au rang k avec (X_{k-1}, Y_{k-1}) un nœud qui est découpé avec $(A, B) \in H$.
– Si $X_{k-1} \cap A = \emptyset$ ou $Y_{k-1} \cap B = \emptyset$ alors $(X_k, Y_k) = (X_{k-1}, Y_{k-1})$ et $H_{L_k} = H_{L_{k-1}}$. Cela signifie que (X_k, Y_k) vérifie les hypothèses (1) et (2).
– Sinon, à partir de la définition des concepts formels, seulement une des propriétés suivantes est vraie :
– Si $C_X \cap A \neq \emptyset$ et $C_Y \cap B = \emptyset$ alors le fils droit $(X_{k-1}, Y_{k-1} \setminus B)$, un sous-ensemble de (X_{k-1}, Y_{k-1}), contient le concept (C_X, C_Y) car $C_Y \subseteq Y_{k-1} \setminus B$. Ce fils est généré par l'algorithme car l'hypothèse (2) est vérifiée au rang $k-1$. $H_{L_k} = H_{L_{k-1}}$ ((A, B) n'appartient pas à H_{L_k} car (X_k, Y_k) est le fils droit de (X_{k-1}, Y_{k-1})), et donc l'hypothèse (1) est vérifiée.
– Si $C_X \cap A = \emptyset$ et $C_Y \cap B \neq \emptyset$ alors $(X_{k-1} \setminus A, Y_{k-1})$, le fils gauche, contient (C_X, C_Y). Le fils gauche est toujours généré par l'algorithme. L'hypothèse (1) est donc vérifiée au rang k. De plus, comme $C_Y \cap B \neq \emptyset$ et $H_{L_k} = H_{L_{k-1}} \cup (A, B)$ alors l'hypothèse (2) est aussi vérifiée.
– Nous avons prouvé que $(C_X, C_Y) \subseteq (L_X, L_Y)$. Nous allons maintenant montrer que $(L_X, L_Y) = (C_X, C_Y)$. Dans la section B, nous avons montré que chaque feuille est un 1-rectangle. Les feuilles sont donc à la fois des des sur-ensembles de (C_X, C_Y) et un 1-rectangle, donc $(C_X, C_Y) = (L_X, L_Y)$.

$L \subseteq C$

Nous avons montré que dans la section B chaque feuille est un 1-rectangle et ensuite que chaque concept est associé à une feuille. On doit prouver que chaque feuille est un concept.

Une feuille (L_X, L_Y) est un concept ssi

$$\begin{cases} (L_X, L_Y) \text{ est un 1-rectangle} \\ \nexists x \in \mathcal{O} \setminus L_X \; such \; that \; \forall y \in L_Y, \; (x,y) \in \mathbf{r} \;\; (1) \\ \nexists y \in \mathcal{P} \setminus L_Y \; such \; that \; \forall x \in L_X, \; (x,y) \in \mathbf{r} \;\; (2) \end{cases}$$

Nous allons montrer par l'absurde les propriétés (1) et (2).

- On suppose qu'il existe $x \in \mathcal{O} \setminus L_X$ tel que $\forall y \in L_Y$, $(x,y) \in \mathbf{r}$. Soit (x, B) un élément de H. Ce 0-rectangle (bi-set ne contenant que des valeurs 0) existe car :
  - Soit x est en relation avec tous les éléments de $\mathcal{P}$ et dans ce cas on a une contradiction car si $x \notin H$ alors tous les n œuds de l'arbre contiennent x alors que l'on a supposé que $x \notin L_X$.
  - Soit il existe $B \subseteq \mathcal{P}$ tel que $\forall y \in B$, $(x,y) \notin \mathbf{r}$. (x, B). B est unique par construction de H.

  Soit (A_X, A_Y) un ancêtre de (L_X, L_Y) avant son découpage par (x, B). (L_X, L_Y) doit être un descendant de $(A_X \setminus x, A_Y)$ qui est le fils gauche de (A_X, A_Y). Cela est dû au fait que $x \notin L_X$ (voir la figure B.1).
  Par construction de l'arbre, le fils gauche de (A_X, A_Y) contient B mais, par hypothèse ($\forall y \in L_Y$, $(x,y) \in \mathbf{r}$), L_Y ne contient pas B. Pour qu'un descendant de $(A_X \setminus x, A_Y)$ ne contienne aucun élément de B, il faut qu'un de ces descendants soit un fils droit d'un autre. Soit (A', B') le 0-rectangle qui a mené au fils droit ne contenant pas les éléments de B. Pour que le découpage se produise, il est nécessaire que $B \cap B' \neq \emptyset$. En conséquence, chaque descendant contient au moins un élément de B, ce qui est en contradiction avec le fait que $L_Y \cap B = \emptyset$.

- On suppose qu'il existe $y \in \mathcal{P} \setminus L_Y$ tel que $\forall z \in L_X$, $(z,y) \in \mathbf{r}$. Soit (A_{X_1}, A_{Y_1}) et (A_{X_2}, A_{Y_2}) les deux ancêtres de (L_X, L_Y) tel que (A_{X_1}, A_{Y_1}) est le père de (A_{X_2}, A_{Y_2}), $y \in A_{Y_1}$ et $y \notin A_{Y_2}$. Nous notons (x, B) le 0-rectangle qui a été utilisé pour découper (A_{X_1}, A_{Y_1}) (voir la figure B.2). Ces deux ancêtres existent et (A_{X_2}, A_{Y_2}) est le fils droit de (A_{X_1}, A_{Y_1}). Car $y \notin L_Y$,

  $$(L_X, L_Y) \text{ est un descendant de} (A_{X_2}, A_{Y_2}) = (A_{X_1}, A_{Y_1} \setminus B)$$

  Par construction de H, x appartient à un seul élément de H. Comme (A_{X_1}, A_{Y_1}) a été découpé par (x, B), $x \in A_{X_1}$. De plus, (A_{X_2}, Y_{Y_2}) et tous ces descendants contiennent aussi x. Malheureusement, nous avons montré que (L_X, L_Y) est un descendant de (A_{X_2}, A_{Y_2}). En conséquence,

  $$x \in L_X$$

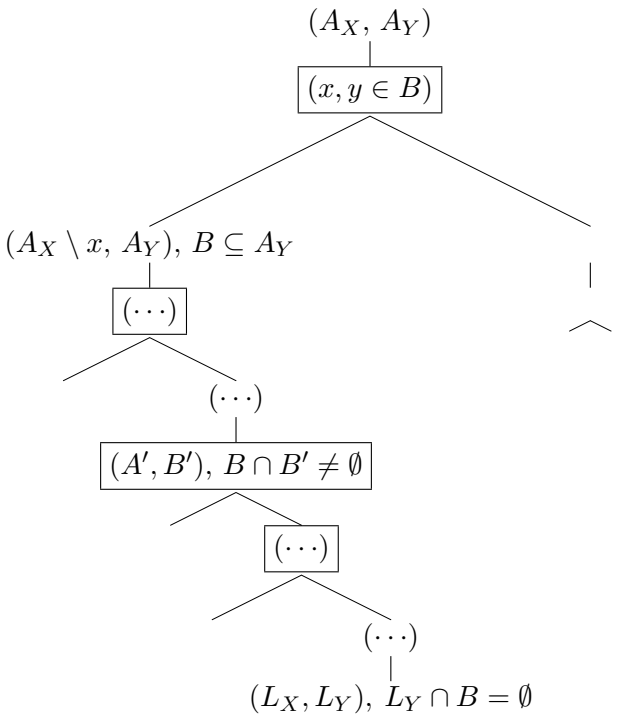


FIG. B.1 – Première illustration : $L_Y \subseteq A_Y \wedge B \subseteq A_Y \wedge B \cap L_Y \neq \emptyset \Rightarrow \neg\,(B \cap L_Y = \emptyset)$

L'hypothèse $\forall z \in L_X,\ (z, y) \in \mathbf{r}$ est contredite par le fait que $(x, y) \notin \mathbf{r}$ (x et y appartiennent au même 0-rectangle).

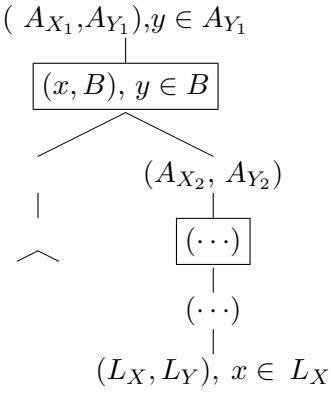


FIG. B.2 – Seconde illustration : $x \in L_X \wedge (x, y) \notin \mathbf{r} \Rightarrow \neg\,(\forall x \in L_X, (x, y) \in \mathbf{r})$

Annexe C

Validation expérimentale de D-Miner

Cette annexe contient des validations expérimentales complémentaires de D-Miner sur différents jeux de données artificielles. Nous avons généré 60 jeux de données denses ayant 300, 700 et 900 colonnes, 100 et 300 lignes et une densité en 1 de 15% et de 35%. Pour chaque combinaison de paramètres, 5 jeux de données ont été générés. La table C.1 (resp. la table C.2) donne la moyenne (la colonne M) et l'écart type (la colonne ET) du temps d'exécution de chaque algorithme en secondes pour les jeux de données ayant une densité de 15% (resp. 35%). Le paramètre σ correspond au seuil de fréquence (taille minimale sur les objets). Pour chaque jeu de données et chaque valeur de σ, nous avons aussi indiqué le nombre de "concepts fréquents" (NFC).

σ	♯ colonnes	300				700				900			
	♯ lignes	100		300		100		300		100		300	
		M	ET	M	ET	M	ET	M	ET	M	ET	M	ET
0,1	CLOSET	0,02	0	0,24	0,1	0,04	0,01	0,5	0,03	0,04	0,01	0,7	0,02
	AC-MINER	0,02	0	0,44	0,03	0,03	0,01	0,56	0,03	0,03	0	0,6	0,04
	D-MINER	0,01	0,01	0,11	0,01	0,01	0,01	0,12	0,01	0,02	0,01	0,15	0,01
	CHARM	0,01	0,01	0,03	0,01	0	0,1	0,06	0,01	0,01	0	0,06	0,01
	NFC	205	21	2631	229	199	18	2320	78	197	17	2254	53
0,01	CLOSET	1,36	0,14	451,2	64,77	1,18	0,14	475,76	32,85	1,16	0,14	384,75	25,63
	AC-MINER	1,32	0,12	–	–	1,82	0,19	–	–	1,96	0,18	–	–
	D-MINER	0,89	0,05	112,6	11,6	0,73	0,06	149,7	9,8	0,86	0,08	154,5	9,7
	CHARM	0,6	0,06	–	–	0,35	0,06	–	–	0,31	0,03	–	–
	NFC	$4\ 10^4$	$4\ 10^3$	$3\ 10^6$	$3\ 10^5$	$4\ 10^4$	$5\ 10^3$	$5\ 10^6$	$2\ 10^5$	$4\ 10^4$	$4\ 10^3$	$4\ 10^6$	$2\ 10^5$
0,001	CLOSET	3,69	0,31	–	–	20,71	2,07	–	–	34,51	3,74	–	–
	AC-MINER	4,79	0,43	–	–	19,86	1,98	–	–	30,51	2,94	–	–
	D-MINER	1,35	0,06	128,31	12,42	6,19	0,4	–	–	11,89	0,81	–	–
	CHARM	1,84	0,2	–	–	11,75	1,44	–	–	19,92	2,75	–	–
	NFC	$5\ 10^4$	$4\ 10^3$	$3\ 10^6$	$3\ 10^5$	$2\ 10^5$	$2\ 10^4$	–	–	$4\ 10^5$	$3\ 10^4$	–	–

TAB. C.1 – Temps d'exécution de D-MINER sur des jeux de données artificielles ayant une densité de 1 de 15% (– si le temps d'extraction est supérieur à 10 minutes)

σ	♯ colonnes	300				700				900			
	♯ lignes	100		300		100		300		100		300	
		M	ET	M	ET	M	ET	M	ET	M	ET	M	ET
0,1	CLOSET	14,57	4.74	–	–	14,02	3,45	–	–	14,08	3,06	–	–
	AC-MINER	16,69	4,28	–	–	22,01	4,26	–	–	24,34	4,51	–	–
	D-MINER	2,53	0,57	–	–	2,36	0,41	–	–	2,79	0,45	–	–
	CHARM	5,86	3,65	–	–	3,53	0,74	–	–	3,34	0,78	–	–
	NFC	$3\ 10^5$	$7\ 10^4$	–	–	$3\ 10^5$	$5\ 10^4$	–	–	$2\ 10^5$	$5\ 10^4$	–	–
0,01	CLOSET	–	–	–	–	–	–	–	–	–	–	–	–
	AC-MINER	–	–	–	–	–	–	–	–	–	–	–	–
	D-MINER	544,41	35,29	–	–	–	–	–	–	–	–	–	–
	CHARM	–	–	–	–	–	–	–	–	–	–	–	–
	NFC	$4\ 10^7$	$3\ 10^6$	–	–	–	–	–	–	–	–	–	–
0,001	CLOSET	–	–	–	–	–	–	–	–	–	–	–	–
	AC-MINER	–	–	–	–	–	–	–	–	–	–	–	–
	D-MINER	550,6	34,98	–	–	–	–	–	–	–	–	–	–
	CHARM	–	–	–	–	–	–	–	–	–	–	–	–
	NFC	$4\ 10^7$	$3\ 10^6$	–	–	–	–	–	–	–	–	–	–

TAB. C.2 – Temps d'exécution de D-MINER sur des jeux de données artificielles ayant une densité de 1 de 30% (– si le temps d'extraction est supérieur à 20 minutes)

Annexe D

Preuves de DR-Miner

Preuves des propriétés des DR-bi-sets

Preuve. [Existence de fonctions sur $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$] On suppose qu'il existe deux DR-bi-sets $B_1 = (S, G_1)$ et $B_2 = (S, G_2)$ tel que $G_1 \neq G_2$. On suppose que $G_2 \setminus G_1 \neq \emptyset$. Si ce n'est pas le cas alors on a forcément $G_1 \setminus G_2 \neq \emptyset$, et la démonstration serait la même en inversant les ensembles G_1 et G_2.

Deux cas sont possibles :
- $G_1 \setminus G_2 = \emptyset$. Dans ce cas, on a $B_2 \subseteq B_1$, ce qui est impossible car les DR-bi-sets sont des bi-ensembles maximaux.
- $G_1 \setminus G_2 \neq \emptyset$. soit $x \in G_1 \setminus G_2$ et $y \in G_2 \setminus G_1$ (x et y existent), alors d'après la propriété 3.6, on a :
  $\mathcal{Z}_l(x, S) < \mathcal{Z}_l(y, S) + \delta$ et $\mathcal{Z}_l(y, S) < \mathcal{Z}_l(x, S) + \delta$. Or comme δ est positif, c'est impossible.

Par l'absurde, on vient de montrer qu'il n'existe pas deux DR-bi-sets (S, G_1) et (S, G_2) tel que $G_1 \neq G_2$ dans un jeu de données. Ainsi, les bi-ensembles (X, Y) de $\mathcal{M}_{DR}^{\alpha\alpha'\delta\delta'}$ sont munis de fonctions notées f_1 et f_2 telles que $f_1(X) = Y$ et $f_2(Y) = X$.

Preuve. [Monotonicité des fonctions f_1 et f_2 avec τ fixé)] Soit deux DR-bi-sets $B_1 = (S_1, G_1)$ et $B_2 = (S_2, G_2)$ tels que $G_1 \subseteq G_2$ et tels qu'ils contiennent au moins une ligne (resp. une colonne) avec τ (resp. τ') valeurs 0 et aucune ligne (resp. colonne) avec plus de τ valeurs 0. On a alors $\forall x \in S_2\, \mathcal{Z}_l(x, G_1) < \tau + \delta$. Ainsi, toutes les lignes de B_2 sont aussi dans B_1 et donc on a $S_2 \subseteq S_1$. Le raisonnement est le même pour les colonnes.

Preuve de la consistance et de la complétude de DR-Miner

Consistence de D-Miner : toutes les feuilles satisfont $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d} \wedge \mathcal{C}_{\delta\delta'\mathbf{r}-p}$

Soit $L = ((X,Y),(\emptyset,\emptyset),(\mathcal{S}\setminus X, \mathcal{G}\setminus Y))$ une feuille. L satisfait $\mathcal{C}_{\alpha\alpha'\mathbf{r}-d}$ sinon ce nœud aurait été élagué (voir la propriété 3.5-1). On considère bien évidemment que $(\mathcal{O},\mathcal{A}) \neq (\emptyset,\emptyset)$.

Un nœud $((Y_S,Y_G),(P_S,P_G),(N_S,N_G)$ n'est pas élagué par la propriété 3.5-2, si

$$\neg(\exists s \in N_s,\ \exists t \in Y_s \ s.t.\ \mathcal{Z}_l(s, Y_G \cup P_G) < \mathcal{Z}_l(t,Y_G) + \delta)$$
$$\Leftrightarrow \forall s \in N_S,\ \forall t \in Y_S \ s.t.\ \mathcal{Z}_l(s, Y_G \cup P_G) \geq \mathcal{Z}_l(t,Y_G) + \delta$$

Or comme L n'a pas été élagué alors il vérifie la propriété suivante :

$$\forall s \in \mathcal{S}\setminus X,\ \forall t \in X \ s.t.\ \mathcal{Z}_l(s,Y) \geq \mathcal{Z}_l(t,Y) + \delta$$

et donc satisfait $\mathcal{C}_{\delta\delta'\mathbf{r}-p}$.

Complétude de DR-Miner

Soit $T_1 = (Y_1,P_1,N_1)$ *et* $T_2 = (Y_2,P_2,N_2)$ alors $T_1 \preceq T_2$ ssi $Y_2 \subseteq Y_1$, $P_1 \subseteq P_2$ et $N_2 \subseteq N_1$.

Nous allons montrer que pour chaque bi-ensemble (X,Y), il existe un chemin dans l'arbre d'exécution de la racine à une feuille correspondant à (X,Y), représentée par le triplet $L = ((X,Y),(\emptyset,\emptyset),(\mathcal{S}\setminus X, \mathcal{G}\setminus Y))$.

Nous allons d'abord montrer la propriété suivante :

Propriété D.1 *Si F est un nœud tel que $L \preceq F$ alors un et un seul fils de F est un sur-ensemble de L par rapport à $\preceq$. Cette propriété est conservée après la propagation de contraintes (voir la propriété 3.6).*

Soit $F = ((F_{YS},F_{YG}),(F_{PS},F_{PG}),(F_{NS},F_{NG}))$ un nœud tel que $L \preceq F$. On suppose que l'énumération est faite sur les lignes (la démonstration est similaire si elle est réalisée sur les colonnes) et que les deux fils de F sont obtenus en utilisant la ligne $s \in F_{PS}$ pour énumérer. Les deux fils ainsi obtenus sont $C_1 = ((F_{YS}\cup s, F_{YG}),(F_{PS}\setminus s, F_{PG}),(F_{NS},F_{NG}))$ et $C_2 = ((F_{YS},F_{YG}s),(F_{PS}\setminus s, F_{PG}),(F_{NS}\cup s, F_{NG})$. Si $s \in X$ alors $L \preceq C_1$ and $L \not\preceq C_2$, sinon $L \preceq C_2$ and $L \not\preceq C_1$. Nous allons montrer maintenant que la propagation de contraintes conserve cette propriété. Or comme elle réduit la taille des treillis (produit des triplets plus petits par rapport à $\preceq$) alors on sait qu'au

plus un des fils de F est un sur-ensemble de L. Il reste à montrer qu'il est effectivement un sur-ensemble de L.

Nous allons montrer que la propagation de contraintes (Property 3.6) appliquée sur un candidat $E = ((E_{YS}, E_{YG}), (E_{PS}, E_{PG}), (E_{NS}, E_{NG}))$ tel que $L \preceq E$ preserve cette ordre. Plus précisément, aucun élément de X n'est déplacé de E_{PS} à E_{YS} (Propriété 3.6-1) et que aucun élément de $\mathcal{O} \setminus X$ n'est déplacé de E_{PS} à E_{NS} (Propriété 3.6-2).

Propriété 3.6-1 : $\forall p \in E_{PS}$ tel que $p \in \mathcal{O} \setminus X$ et $\forall t \in E_{YS}$, on a $\mathcal{Z}_l(p, Y) \geq \mathcal{Z}_l(t, Y) + \delta$. Or on a $E_{YG} \subseteq Y \subseteq E_{YG} \cup E_{PG}$ et donc $\mathcal{Z}_l(p, E_{YG} \cup E_{PG}) \geq \mathcal{Z}_l(p, Y) \geq \mathcal{Z}_l(t, E_{YG}) + \delta$. Finalement, $\mathcal{Z}_l(p, E_{YG} \cup E_{PG}) < \mathcal{Z}_l(t, E_{YG}) + \delta$ est faux, et donc la propriété 3.6-1 n'est pas appliquée et p n'est pas déplacé dans E_{YS}.

Propriété 3.6-2 : $\forall p \in E_{PS}$ tel que $p \in X$, on a $\mathcal{Z}_l(p, Y) \leq \alpha$. Or on a $E_{YG} \subseteq Y$ et donc $\mathcal{Z}_l(p, E_{YG}) \leq \mathcal{Z}_l(p, Y) \leq \alpha$. Finalement, p n'est pas déplacé dans E_{NS}.

Nous venons de montrer que la propagation de contraintes préserve l'ordre $L \preceq E$.

Comme la racine $((\emptyset, \emptyset), (\mathcal{S}, \mathcal{G}), (\emptyset, \emptyset))$ est un sur-ensemble de L, sachant que la profondeur de l'arbre est bornée ($\leq |\mathcal{O}| + |\mathcal{A}|$) et que $L \preceq E$ est préservé par la propagation de contraintes, alors tous les bi-ensembles satisfaisant $\mathcal{C}_{\alpha\alpha' \mathbf{r} - d} \wedge \mathcal{C}_{\delta\delta' \mathbf{r} - p}$ sont extraits par DR-Miner .

Annexe E

Validation biologique

Results from the ChIP assays performed to verify the DNA binding of SREBP1 on the promoter sequences of the 13 genes identified by D-Miner.

The Negative Control primers flank a region of DNA that should not be bound by SREBP1 (exon 2 of HK2 with no SRE binding site). These primers are used on immunoprecipitated DNA to confirm that antibody-enrichment of a target DNA is due to specific immunoprecipitation of the protein target, rather than a non-specific precipitation of total DNA. Note: the Negative Control primers give an amplification product because chromatin immunoprecipitation is an enrichment of DNA bound by a particular protein, not a complete purification of the DNA of interest. If enough PCR cycles are used, it is always possible to get a PCR product for a given target locus. No SREBP1 interaction were detected with promotors of CRYBA4, UBE2V2 and ABCA7. MORF4L2 and MAPRG1 were not analysed as it was not possible to obtain a clean single band after PCR amplifications from Input.
AB = Chromatin immunoprecipitation with antibody against SREBP1. Mock = Chromatin immunoprecipitation without antibody (control). Input = Sonicated chromatin (control).
The SREBP1 (M00221) putative motifs are given for each gene (in color).

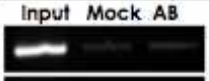
Gene name	SRE sequences	Gel ChIP	SREBP1 binding	Response to Insulin (fold changes) (11)
		Input Mock AB		
HK2 (exon2)	Negative control		0	–
UBE2V2*	ATCACCCGAG		0	3.08
CRYBA4	ATCACCACCC		0	-2.30
ABCA7	ATCACCCCAC		0	-1.98
ARF4	TATCACCCCG		+	3.11
SPOP	ATCGCACCAC		+	2.00
FEM1b*	GACACCCCAC		+	2.60
VPS29	TACCACCCCG		+	3.65
HK2*	CTCCCCCCAC		+	3.02
HIG1	CTTCTCCCAC		+	3.85
PGRMC2*	CTCGCCCCAC		+	2.07
SDC1	AACGCCCCAC		+	-1.43
SF1	TTCGCCCCAC		+	1.33
FAS*	ATCACCCCAC TRANSFAC concensus	Latasa MJ et al. 2003 [14]	+	

FIG. E.1 – Validation biologique par ChIP

Bibliographie

[1] F. N. Afrati, A. Gionis, and H. Mannila. Approximating a collection of frequent sets. In SIGKDD, pages 12–19, Seattle, WA, USA, Aug. 2004. ACM Press.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In SIGMOD, pages 207–216, Washington, D.C., USA, June 1993. ACM Press.

[3] M.N. Arbeitman, E.E. Furlong, F. Imam, E. Johnson, B.H. Null, B.S. Baker, M.A. Krasnow, M.P. Scott, R.W. Davis, and K.P. White. Gene expression during the life cycle of drosophila melanogaster. Science, 297(5590) :2270–2275, sept. 2002.

[4] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approch to bregman co-clustering and matrix approximation. In SIGKDD, pages 509–514, Seattle, WA, USA, Aug. 2004. ACM Press.

[5] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. SIGKDD Explorations, 2(2) :66–75, December 2000.

[6] C. Becquet, S. Blachon, B. Jeudy, J-F. Boulicaut, and O. Gandrillon. Strong association rule mining for large gene expression data analysis : a case study on human SAGE data. Genome Biology, 3(12), Nov. 2002. See http ://genomebiology.com/2002/3/12/research/0067.

[7] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review, 67, March 2003.

[8] A. Berry and A. Sigayret. Representing a concept lattice by a graph. In Workshop on Discrete Mathematics and Data Mining (DM&DM), volume 144, pages 27–42. Elsevier Science, November 2004.

[9] J. Besson and J-F. Boulicaut C. Robardet. Mining alpha-beta concepts as relevant bi-sets from transactional data. In KDID co-located with ECML/PKDD, pages 13–24, Pisa, Italie, September 2004.

[10] J Besson, R. G. Pensa, S. Blachon, C. Robardet, and J-F. Boulicaut. A simple tool to support gene expression analysis. In Délivrable D14C - Projet enropéen CinQ (IST-2000-26469), avril 2004.

[11] J. Besson, F. Rioult, B. Crémilleux, S. Rome, and J.-F. Boulicaut. Solutions pour le calcul d'ensembles fréquents dans des données biopuces. In Informatique pour l'analyse du transcriptome, pages 231–254. Hermes Science, 2004. Chapitre 8 dans ce volume.

[12] J. Besson, C. Robardet, and J-F. Boulicaut. Constraint-based mining of formal concepts in transactional data. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PaKDD), volume 3056 of LNCS, pages 615–624, Sydney, Australia, May 2004.

[13] J. Besson, C. Robardet, and J-F. Boulicaut. Approximation de collections de concepts formels par des bi-ensembles denses et pertinents. In Conférence d'Apprentissage (CAp), pages 313–328, Juin 2005.

[14] J. Besson, C. Robardet, and J-F. Boulicaut. Mining formal concepts with a bounded number of exceptions from transactional data, volume 3377 of Lecture Notes in Computer Science, pages 33–45. Springer-Verlag, 2005. This is a minor revision of our KDID'04 paper.

[15] J. Besson, C. Robardet, J-F. Boulicaut, and S. Rome. Constraint-based bi-set mining for biologically relevant pattern discovery in microarray data. Intelligent Data Analysis journal, 9(1) :59–82, 2004.

[16] J. Besson, C. Robardet, E. Meugnier, S. Rome, and J-F. Boulicaut. Extraction of relevant transcription modules using pattern discovery. In Integrative Post-Genomics IPG, anciennement JPGD, Lyon, France, Octobre 2004. Poster and oral communication.

[17] M. Bistarelli and F. Bonchi. Interestingness is not a dichotomy : introducing softness in constrained pattern mining. In Principles and Practice of Knowledge Discovery in Databases (PKDD), Porto, Portugal, Sept. 2005. To appear.

[18] S. Blachon, R. Pensa, J. Besson, C. Robardet, JF. Boulicaut, and O. Gandrillon. Using formal concepts to support knowledge discovery from sage data. Rapport de recherche, Villeurbanne : LIRIS CNRS FRE 2672, July 2005. 32 pages, Submitted to BMC Bioinformatics.

[19] S. Blachon, C. Robardet, J.-F. Boulicaut, and O. Gandrillon. Extraction de connaissances dans les données d'expression SAGE humaines. In Informatique pour l'analyse du transcriptome, pages 207–230. Hermes Science, 2004. Chapitre 7 dans ce volume.

[20] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. university of california, irvine, dept. of information and computer sciences, 1998. http ://www.ics.uci.edu/∼mlearn/MLRepository.html.

[21] J.P. Bordat. Calcul pratique du treillis de galois d'une correspondance. Mathématiques et sciences humaines, 96 :31–47, 1986.

[22] J-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In Pacific-Asia Conference on Knowledge Discovery

and Data Mining (PaKDD), volume 1805 of LNCS, pages 62–73, Kyoto, Japan, 2000. Springer-Verlag.

[23] J-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by mean of free-sets. In Principles and Practice of Knowledge Discovery in Databases (PKDD), volume 1910 of LNCS, pages 75–85, Lyon, France, Sept. 2000. Springer-verlag.

[24] J-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets : a condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery journal, 7(1) :5–22, January 2003.

[25] J-F. Boulicaut, M. Klemettinen, and H. Mannila. Querying inductive databases : a case study on the mine rule operator. In Principles of Data Mining and Knowledge Discovery, volume 1510 of LNCS, pages 194–202, Nantes, France, Sept. 1998. Springer-Verlag.

[26] J-F. Boulicaut, M. Klemettinen, and H. Mannila. Modeling kdd processes within the inductive database framework. In Data Warehousing and Knowledge Discovery (DaWak), volume 1676 of LNCS, pages 293–302, August 1999.

[27] J-F. Boulicaut, F. Rioult, J. Besson, B. Crémilleux, and S. Rome. Faisabilite des extractions d'ensembles frequents dans des donnees biopuces : elements de solution. In Actes de la journée Informatique pour l'analyse du transcriptome JPGD, Lyon, France, Mai 2003. 13 pages.

[28] Z. Bozdech, M. Llinás, B. L. Pulliam, E. Wong, J. Zhu, and J. DeRisi. The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum. PLoS Biol, 1(e5), 2003.

[29] C. Bucila, J. E. Gehrke, D. Kifer, and W. White. Dualminer : A dual-pruning algorithm for itemsets with constraints. Data Mining and Knowledge Discovery Journal, 7(4) :241–272, Oct. 2003.

[30] S. Busygin, G. Jacobsen, and E. Kramer. Double conjugated clustering applied to leukemia microarray data. In SIAM ICDM Workshop on clustering high dimensional data, San Diego, CA ,USA, August 2002. SIAM.

[31] A. Bykowski. Condensed representations of frequent sets : application to descriptive pattern discovery. PhD thesis, Institut National des Sciences Appliquées de Lyon, oct 2002.

[32] A. Bykowski and C. Rigotti. DBC : a condensed representation of frequent patterns for efficient mining. Information Systems Journal, 28(8) :949–977, 2003.

[33] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarray classification califano. In Computational Molecular Biology, pages 75–85, San Diego, CA, USA, August 2000. AAAI.

[34] Y. Cheng and G. M. Church. Biclustering of expression data. In ISMB, pages 93–103. AAAI Press, August 2000.

[35] M. Courtine. Changement de représentation pour la classification conceptuelle non supervisée de données complexes. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2002.

[36] M. Courtine, J.-D. Zucker, and K. Clément. Classification et caractérisation automatique des fonctions de gènes co-exprimés. In Informatique pour l'analyse du transcriptome, pages 255–272. Hermes Science, 2004. Chapitre 9 dans ce volume.

[37] L. de Raedt. Query evaluation and optimization for inductive database using version spaces (extended abstract). In DTDM co-located with EDBT, pages 19 – 28, Praha, Czech Republica, March 2002.

[38] L. de Raedt, M. J., S. D. Lee, and H. Mannila. A theory of inductive query answering. In ICDM, pages 123–130, Maebashi City, Japan, December 2002. IEEE Computer Society.

[39] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic co-clustering. In ACM SIGKDD, pages 89–98, Washington, DC, USA, August 2003. ACM.

[40] E. Wingender E, X. Chen, E. Fricke E, R. Geffers, R. Hehl, and etal. The transfac system on gene expression regulation. Nucleic Acids Res., 29(1) :281–283, January 2001.

[41] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. PNAS, 95(25) :14863–14868, Dec. 1998.

[42] M. Foretz, C. Guichard, P. Ferre, and F. Foufelle. Sterol regulatory element binding protein-1c is a major mediator of insulin action on the hepatic expression of glucokinase and lipogenesis-related genes. PNAS, 96(22) :12737–12742, Oct. 1999.

[43] O. Gandrillon and R. Houlgatte. Le transcriptome : le nouveau monde ? In Informatique pour l'analyse du transcriptome, pages 21–44. Hermes Science, 2004. Chapitre 1 dans ce volume.

[44] B. Ganter. Two basic algorithms in concept analysis. Technical report, Germany Darmstadt : Technisch Hochschule Darmstadt, Preprint 831, 1984.

[45] G. Getz and E. Domany. coupled two-way clustering analysis of gene expression microarray data. PNAS, 97(22) :12079–12084, October 2000.

[46] A. Gionis, H. Mannila, and J. K. Seppänen. Geometric and combinatorial tiles in 0-1 data. In Principles and Practice of Knowledge Discovery (PKDD), volume 3202 of LNCS, pages 173–184, Pisa, Italy, Sept. 2004. Springer.

[47] R. Godin, R. Missaoui, and H. Alaoui. Learning algorithms using galois lattice structure. In International conference on tools for Artificial Intelligence, pages 22–29, California, USA, November 1991. IEEE computer society.

[48] B. Goethals and M. J. Zaki, editors. Frequent Itemset Mining Implementations, volume 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, Decem. 2003. CEUR-WS.org.

[49] L. A. Goodman and W. H. Kruskal. Measures of association for cross classification. Journal of the American Statistical Association, 49 :732–764, 1954.

[50] Jiawei Han, Yongjian Fu, Wei Wang, Krzysztof Koperski, and Osmar Zaiane. DMQL : a data mining query language for relational databases. In Data Mining and Knowledge Discovery workshop, Mondreal, Canada, June 1996.

[51] J. Hartigan. Direct clustering of data matrix. American Statistical Association, 67(337) :123–129, March 1972.

[52] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. Nature Genetics, 31 :370–377, August 2002.

[53] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of the ACM, 39(11) :58–64, November 1996.

[54] B. Jeudy. Extraction de motifs sous contraintes : application à l'évaluation de requêtes inductives. PhD thesis, Institut National des Sciences Appliquées de Lyon, December 2002.

[55] D.S. Johnson, M. Yannakakis, and C.H. Papadimitriou. On generating all maximal independent sets. Inf. Process. Lett., 27 :119–123, 1988.

[56] Y. Kluger, R. Basri, JT Chang, and M. Gerstein. Spectral biclustering of microarray data : coclustering genes and conditions. Genome Research, 13 :703–716, 2003.

[57] T. Kohonen. Self-organizing maps. In Springer Series in Information Science, volume 30. Springer-Verlag, Berlin, Germany, 1995.

[58] S.O. Kuznetsov. Interpretation on graphs on complexity characteristics of a search for specific patterns. Automatic Documentation and Mathematical Linguistics, 24(1) :37–45, 1989.

[59] S.O. Kuznetsov and S. Obiedkov. Comparing performance of algorithms for generating concept lattices. In Experimental and Theoretical Artificial Intelligence, volume 14, pages 189–216. Taylor and Francis, April 2001.

[60] M.J. Latasa, M.J. Griffin, Y.S. Moon, C. Kang, and H.S. Sul. Occupancy and function of the -150 sterol regulatory element and -65 e-box in nutritional regulation of the fatty acid synthase gene in living animals. Mol Cell Biol, 23 :5896–907, 2003.

[61] L. Lazzeroni and A. Owen. Plaid models for gene expression data. Technical report, Stanford University, 2000.

[62] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In SIGKDD, pages 189–194, Portland, Oregon, USA, August 1996. ACM press.

[63] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. In Data Mining and Knowledge Discovery journal, volume 1(3), pages 241–258. Kluwer Academic Publishers, 1997.

[64] M.-L. Martin-Magniette and S. Robin. Techniques statistiques pour l'analyse du transcriptome. In Informatique pour l'analyse du transcriptome, pages 67–100. Hermes Science, 2004. Chapitre 3 dans ce volume.

[65] C. Masson. Contribution au cadre des bases de données inductives : formalisation et évaluation de scénarios d'extraction de connaissances. PhD thesis, Institut National des Sciences Appliquées de Lyon, 2005.

[66] R. Meo, P.L. Lanzi, and M. Klemettinen. Database support for data mining applications : Discovering knowledge with inductive queries. In Database Support for Data Mining Applications, volume 2682 of LNCS. Springer, 2004.

[67] R. Meo, G. Psaila, and S. Ceri. An extension to SQL for mining association rules. In Data Mining and Knowledge Discovery, volume 2(2), pages 195–224. Kluwer Academics Publisher, 1998.

[68] E. Meugnier, J. Besson, J-F. Boulicaut, E. Lefai, H. Vidal, and S. Rome. Resolving transcription network from microarray data with constraint-based formal concept mining revealed new target genes of srebp1. In PLOS Computational Biology, 2005. papier soumis.

[69] T. M. Mitchell. Generalization as search. Artificial Intelligence, 18 :203–226, 1982.

[70] C. Niehrs and N. Pollet. Synexpression groups in eukaryotes. Nature, 402 :483–487, 1999.

[71] V. Orlando, H. Strutt, and R. Paro. Analysis of chromatin structure by in vivo formaldehyde cross-linking. Methods, 11 :205–214, 1997.

[72] T.F. Osborne. Sterol regulatory element-binding proteins (srebps) : key regulators of nutritional homeostasis and insulin action. J Biol Chem, 275 :32379–32382, 2000.

[73] F. Pan, G. Cong, A. K.H. Tung, J. Yang, and M. J. Zaki. CARPENTER : Finding closed patterns in long biological datasets. In SIGKDD. ACM Press, August 2003.

[74] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. Information Systems, 24(1) :25–46, January 1999.

[75] J. Pei, J. Han, and R. Mao. CLOSET an efficient algorithm for mining frequent closed itemsets. In Data Mining and Knowledge Discovery, 2000. http ://www-sal.cs.uiuc.edu/ hanj/pubs/kdd.htm.

[76] R. Pensa, J. Besson, C. Robardet, and J-F. Boulicaut. Contribution to set pattern mining from large gene expression datasets. constraint-based mining and inductive databases. LNCS, 20, 2005. To appear-20 pages.

[77] R. Pensa and J-F. Boulicaut. From local pattern discovery to relevant bi-cluster characterization. In Symposium on Intelligent Data Analysis IDA, volume 3646 of LNCS, pages 293–304, Kos Island, Greece, Sept. 2005. Springer-Verlag.

[78] R. Pensa and J-F. Boulicaut. Towards fault-tolerant formal concept analysis. In Congress of the Italian Association for Artificial Intelligence AI*IA, volume 3673 of LNAI, pages 21–23, Milano, Italy, Sept. 2005. Springer-Verlag.

[79] R. G. Pensa, J. Besson, and J-F. Boulicaut. A methodology for biologically relevant pattern discovery from gene expression data. In International Converence on Discovery Science (DS), volume 3245, pages 230–241, Padova, Italy, October 2004. Springer-Verlag.

[80] R. G. Pensa, C. Leschi, J. Besson, and J-F. Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data. In Workshop on Data Mining in Bioinformatics co-located with ACM SIGKDD, pages 24–30, Seattle, USA, August 2004. ACM Press.

[81] J. L. Pfaltz and C. M. Taylor. Closed set mining of biological data. In BIOKDD co-located with SIGKDD, Edmonton, Alberta, Canada, July 2002.

[82] N. Reymond, H. Charles, S. Rome, and J. Marti. Les données d'expressions. In Informatique pour l'analyse du transcriptome, pages 45–66. Hermes Science, 2004. Chapitre 2 dans ce volume.

[83] F. Rioult, J-F. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In Data Mining and Knowledge Discovery, volume 16, pages 73–79, San Diego, USA, June 2003. ACM Press.

[84] F. Rioult and B. Crémilleux. Condensed representations in presence of missing values. In Intelligent Data Analysis (IDA), volume 2810 of LNCS, pages 578–588, Berlin, Germany, August 2003. Springer.

[85] C. Robardet. Contribution à la classification non supervisée : proposition d'une méthode de bi-partitionnement. PhD thesis, University Claude Bernard - Lyon 1, Villeurbanne, Juillet 2002.

[86] C. Robardet and F. Feschet. Comparison of three objective functions for conceptual clustering. In Principles and Practice of Knowledge Discovery (PKDD), volume 2168 of LNCS, pages 399–410, Freiburg, Germany, Sept. 2001. Springer-Verlag.

[87] C. Robardet, R. G. Pensa, J. Besson, and J-F. Boulicaut. Using classification and visualization on pattern databases for gene expression data analysis. In Workshop on Pattern Representation and Management PaRMa co-located with EDBT, volume 16, pages 107–118, Heraclion - Crete, Greece, March 2004. CEUR Workshop Proceedings.

[88] C. Robardet and C. Rigotti. Etude de méthodes de recherche locale pour la construction de bi partitions. RSTI-RIA-ECA, 16 :705–728, 2002.

[89] S. Rome, K. Clément, R. Rabasa-Lhoret, E. Loizon, C. Poitou, G. S. Barsh, J-P. Riou, M. Laville, and H. Vidal. Microarray profiling of human skeletal muscle reveals that insulin regulates 800 genes during an hyperinsulinemic clamp. Journal of Biological Chemistry, May 2003. 278(20) :18063-8.

[90] J. K. Seppänen and H. Mannila. Dense itemsets. In SIGKDD, pages 683–688. ACM Press, Seattle, Washington, USA, August 2004.

[91] A. Soulet and B. Cremilleux. An efficient framework for mining flexible constraints. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PaKDD), volume 3518 of LNCS, pages 661–671, Hanoi, Vietnam, 2005.

[92] G. Stumme, R. Taouil, Y. Bastide, N. Pasqier, and L. Lakhal. Computing iceberg concept lattices with titanic. Data and Knowledge Engineering, 42 :189–222, 2002.

[93] A. Tanay, R. Sharan, and R. Shalir. discovery statistically significant biclusters in gene expression data. Bioinformatics, 18 :136–144, 2002.

[94] V. Goss Tusher, R. Tibshirani, and G. chu. Significance analysis of microarrays applied to the ionizing radiation response. PNAS, 98 :5116–5121, 2001.

[95] V. Ventos, H. Soldano, and T. Lamadon. Alpha galois lattices. In ICDM, pages 555–558, Brighton, UK, November 2004. IEEE Computer Society.

[96] H. Wang, W. Wang, J. Yang, and P. S. YU. Clustering by pattern similarity in large data sets. In SIGMOD, pages 394–405, Madison, Wisconsin, USA, june 2002. ACM Press.

[97] J. Wang, J. Han, and J. Pei. CLOSET+ : searching for the best strategies for mining frequent closed itemsets. In SIGKDD, Washington, DC, USA, August 2003. ACM Press.

[98] R. Wille. Restructuring lattice theory : an approach based on hierarchies of concepts. In I. Rival, editor, Ordered sets, pages 445–470. Reidel, Dordrecht, 1982.

[99] C. Yang, U. Fayyad, and P. S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In SIGKDD, pages 194–203, San Francisco, California, USA, August 2001. ACM Press.

[100] J. Yang, W. Wang, H. Wang, and P. Yu. Delta-cluster : capturing subspace correlation in a large data set. In Data Engineering, pages 517–528, San Jose, CA, february 2002. IEEE.

[101] M. J. Zaki and C-J. Hsiao. CHARM : An efficient algorithm for closed itemset mining. In SIAM DM, Arlington, VA, USA, April 2002. SIAM.

[102] H. Zhang, Y. Ramanathan, P. Soteropoulos, M.L. Recce, and P.P. Tolias. Ez-retrieve : a web-server for batch retrieval of coordinate-specified human dna sequences and underscoring putative transcription factor-binding sites. Nucleic Acids Research, 30(21) :121–127, 2002.