UNIVERSITÉ DE STRASBOURG

UNIVERSITÉ DE STRASBOURG

msii ÉCOLE DOCTORALE

École Doctorale Mathématiques,
Sciences de l'Information et de l'Ingénieur

ICube

THÈSE

soutenue le 26 novembre 2014 pour obtenir le grade de

Docteur de l'université de Strasbourg
Discipline : Informatique

par

Mickaël FABRÈGUE

Extraction d'informations synthétiques à partir de données séquentielles

Application à l'évaluation de la qualité des rivières

Jury composé de :

Rapporteurs :	
Bruno CRÉMILLEUX	Professeur, Université de Caen
Amedeo NAPOLI	Directeur de recherche, CNRS, Nancy
Examinateurs :	
Danielle LEVET	Directrice bureau d'études environnement
Philippe USSEGLIO-POLATERA	Professeur, Université de Lorraine
Directrices de thèse :	
Florence LE BER	Ingénieure en chef des Ponts, des Eaux et des Forêts (ICPEF), HDR, Université de Strasbourg/ENGEES
Maguelonne TEISSEIRE	Directrice de recherche, IRSTEA, Montpellier

---

Membres invités :

Agnès BRAUD	Maître de conférence, Université de Strasbourg
Sandra BRINGAY	Maître de conférence, Université Montpellier III

TETIS

iCUBE

We are drowning in information
but starved for knowledge.

John Naisbitt

Remerciements

Aboutissement de trois années de travail, ce manuscrit est ce qu'il est grâce à de nombreuses personnes. Leur contribution est à la fois professionnelle, académique mais également humaine. Je tiens donc sincèrement a remercier toutes ces personnes qui ont participé, chacune à leur manière, à cette thèse.

En premier lieu, mes remerciements vont à mes directrices de thèse, Florence Le Ber et Maguelonne Teisseire, ainsi qu'à mes encadrantes de thèse, Agnès Braud et Sandra Bringay. Je suis vraiment ravi d'avoir été encadré et d'avoir travaillé avec vous pendant ces trois ans, à la fois pour vos inestimables conseils, vos relectures précieuses et vos encouragements, et pour la grande liberté de recherche que vous m'avez laissée. Assurément, je n'aurais pas pu avoir un meilleur encadrement et environnement de travail que celui vous m'avez donné.

Merci à Bruno Crémilleux et Amedeo Napoli pour avoir acceptés d'évaluer ma thèse. Être rapporteur est une tâche difficile qui demande beaucoup de disponibilité. Également, je remercie Danielle Levet et Philippe Usseglio-Polatera pour avoir accepté d'être examinateur lors de ma soutenance de thèse.

Mes remerciements vont aussi à Marianne Huchard et à Pascal Poncelet pour avoir fait partie de mon comité de suivi de thèse en évaluant l'avancement de mes travaux en fin de seconde année.

Cette thèse ayant été effectuée sur plusieurs sites, à Montpellier et Strasbourg, je tiens à remercier les responsables d'équipes et directeurs de laboratoires. Je remercie donc Nicolas Lachiche (BFO/ICube), Pascal Poncelet (TATOO/LIRMM), Jean-Philippe Tonneau (TETIS) et Jean-François Quéré (ENGEES) pour m'avoir accueilli et m'avoir offert un cadre de travail idéal, que ce soit d'un point de vue matériel ou intellectuel.

Un grand merci à tous mes collègues, du LIRMM à l'ENGEES en passant par la MTD. Tout d'abord merci à mes collègues doctorants, Juliane, Hugo, Juan, Nathalie, Flavien, Lilia, Claude, Hai, Marc et Louise mais aussi à Xavier, Dino, Éric, Fabio, Julien et Valérie pour l'ambiance, les pauses café et la bonne humeur de travail.

Également, je remercie toutes les personnes du projet ANR Fresqueau avec qui la collaboration a été des plus riches et intéressante ainsi que Corinne, Fred, Danielle et les bureaux d'études Aquabio et Aquascop pour le travail sur l'interprétation des données.

Je remercie particulièrement l'équipe TATOO du LIRMM. Mon parcours est devenu ce qu'il est car, lors de mon master, vous m'avez donné ma chance au travers de nombreux projets et stages, qui m'ont donné l'envie de continuer, d'aller plus loin dans la recherche et tout simplement de faire cette thèse.

Un remerciement spécial pour Laetitia, qui m'a supporté, soutenu et qui a été le meilleur des traiteurs pour mon pot de thèse. J'exprime aussi toute ma gratitude envers tous mes proches, ma famille et mes amis. Avoir pu régulièrement me changer les idées et m'échapper de la bulle thèse a indirectement, mais sûrement, contribué à la réussite de ces trois ans de recherche.

Enfin, je remercie l'ANR qui a financé les travaux de cette thèse dans le cadre du projet Fresqueau (ANR11_MONU14).

Table des matières

Table des figures

Liste des tableaux

CHAPITRE 1

Introduction

Selon le Planetoscope [1], chaque seconde plus de 29 000 gigaoctets de données circulent en ligne dans le monde, ce qui au total fait 912,5 exaoctets dans l'année. Pour l'Europe de l'ouest, il est prédit que le volume des données digitales va augmenter de plus de 30% chaque année jusqu'en 2020 [2]. Eric Schmidt, l'ancien PDG de Google, affirmait en 2010 qu'en deux jours nous générions autant de données que ce qu'a générées l'humanité jusqu'en 2003.

Les données produites et collectées, et de ce fait les bases de données qui les stockent, deviennent de plus en plus importantes et complexes. Nous assistons de plus en plus à la collecte automatique des données à l'aide de capteurs. Ainsi, la génération et le stockage de celles-ci sont omniprésents, comme en astronomie avec les satellites et les télescopes, ou bien encore dans nos smartphones avec la géolocalisation. Dans sa recherche du boson de Higgs, le CNRS a publié que le Grand Collisionneur de Hadrons (LHC) du CERN [3] produit plus de 15 pétaoctets de données chaque année [4].

Les enjeux industriels qui découlent de ce stockage massif de l'information sont nombreux. Un exemple, que nous connaissons tous, est la présence de plus en plus importante de la publicité sur internet. En 2012, le marché publicitaire en ligne français a atteint les 2,7 milliards d'euros net avec une croissance de 7% pour les liens sponsorisés. L'objectif est de cibler au mieux le profil des utilisateurs afin de leur proposer les annonces et publicités susceptibles de les intéresser. Dans ce contexte, le 14 avril 2014, Google a mis à jour les conditions générales d'utilisation (CGU) de Gmail. Il annonce officiellement aux utilisateurs que les e-mails reçus et envoyés sont scannés par des logiciels pour le ciblage publicitaire. Cette multinationale a développé des algorithmes dans l'objectif de traiter cette grande quantité de données

1. www.planetoscope.com
2. http ://www.emc.com/collateral/analyst-reports/idc-digital-universe-western-europe.pdf
3. http ://home.web.cern.ch/fr
4. https ://lejournal.cnrs.fr/articles/le-big-data-un-enjeu-economique-et-scientifique

textuelles complexes, issue de ses utilisateurs.

Devant cette explosion du volume de l'information stockée, identifier les informations utiles, les extraire et les analyser de manière automatique implique la définition de nouvelles méthodes. C'est dans ce contexte que s'est développée l'*Extraction de Connaissances dans les Données* (ECD). Ce domaine s'étend au-delà de la conception de nouvelles approches pour extraire des informations d'un jeu de données. L'ECD se réfère à un processus mettant en jeu un ensemble d'opérations effectuées sur les données. Ces dernières vont des requêtes dans une base de données à la restitution des informations extraites à l'utilisateur, qui peut les interpréter en éléments de connaissance. L'application d'algorithmes de fouille de données représente une seule de ces étapes. Ce processus respecte généralement un schéma décrit par la figure 1.1 [FPSSU96].

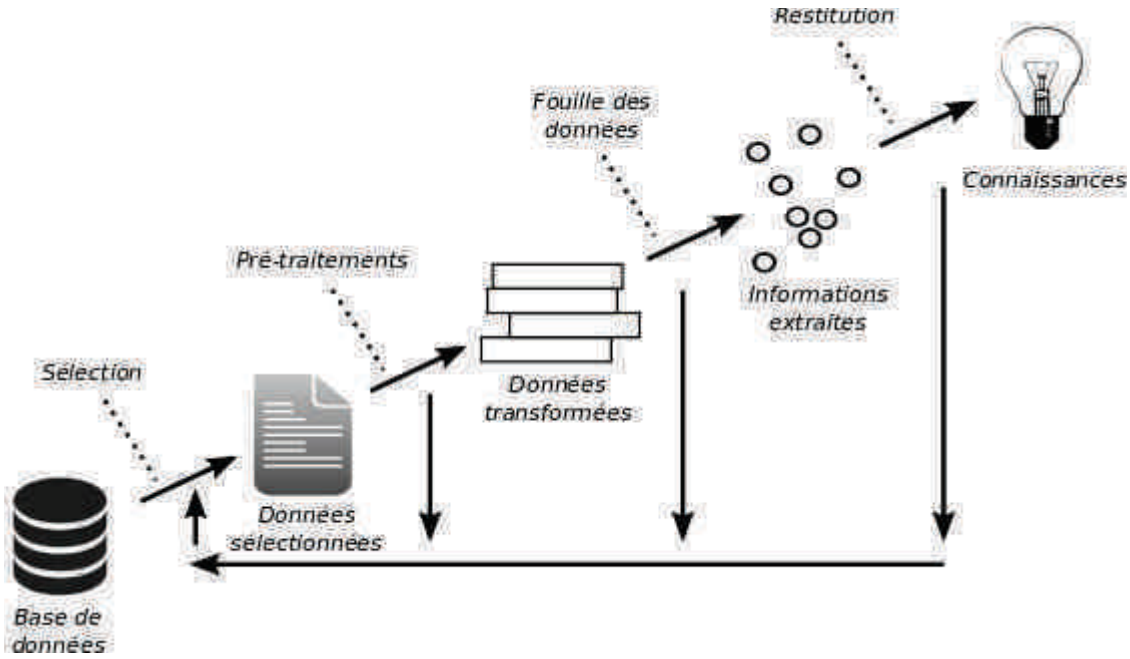


FIGURE 1.1 – Processus de fouille de données

Nous détaillons maintenant chacune de ces étapes :

Sélection des données Couramment effectuée à l'aide de requêtes, cette première étape consiste à sélectionner, dans une base ou un entrepôt de données, les informations relatives au problème pour lequel nous souhaitons construire de nouvelles connaissances.

Pré-traitements Les données sélectionnées sont souvent bruitées, de qualité hétérogène ou bien ne correspondent pas au format d'entrée des algorithmes

de fouille. Cette étape vise à nettoyer les données et à les transformer dans un format adéquat.

Fouille des données C'est l'étape centrale du processus. Les données sélectionnées et pré-traitées sont explorées avec un ou plusieurs algorithmes adaptés afin d'en extraire un ensemble d'informations. Celles-ci peuvent par exemple être un ensemble de motifs, de règles ou un regroupement par classes.

Restitution Les informations extraites ne sont souvent pas directement interprétables. En effet, les résultats des algorithmes peuvent être chargés en mémoire, stockés dans un fichier texte ou également affichés dans une console. Cette phase consiste à traiter le format de sortie des algorithmes pour restituer les résultats, les rendre facilement visualisables et analysables par les utilisateurs.

Ce schéma est très général, et les différentes étapes peuvent varier en fonction de nombreux critères. Le domaine d'application de la fouille de données touche à tous les secteurs. Nous pourrions même dire que partout où les données sont présentes, l'extraction de connaissances est possible. Les méthodes d'alignement de séquences ADN ont fortement contribué à la recherche en phylogénétique [OMHO94]. Dans un autre domaine, la protection des réseaux informatiques a pu être renforcée grâce aux méthodes de détection d'anomalies [LEK+03]. Dans le secteur des finances, nous avons assisté, depuis le début du XXIème siècle, à une multiplication d'algorithmes destinés à prédire le cours futur de la bourse [ET05, BMZ11].

Comme dans ce dernier exemple, c'est dans le domaine des données temporelles que s'inscrit cette thèse, et plus spécifiquement dans le domaine des séquences temporelles constituées de données symboliques. Les travaux sur la fouille de données temporelles sont nombreux et il existe de multiples manières de fouiller ces données, de les filtrer et de représenter les informations extraites. Dans ce contexte, une thématique importante est d'arriver à extraire un minimum d'éléments qui contiennent un maximum d'informations sur les données. Cet aspect constitue la principale motivation de cette thèse, présentée dans la section suivante.

1.1 Motivations

Comme nous l'avons dit précédemment, la complexité des données collectées n'a cessé d'augmenter. Ainsi, certaines données ne sont pas seulement définies par un paramètre auquel une valeur est attribuée, mais aussi par une multitude de dimensions associées, et en particulier la dimension temporelle. Dans ce manuscrit, nous proposons un ensemble d'approches méthodologiques basées sur un type de motifs, les motifs partiellement ordonnés, pour traiter les données temporelles.

Dans le cadre de données temporelles, ce type de motifs a été moins étudié que des approches à base de motifs séquentiels, par exemple. Pourtant son intérêt est significatif puisqu'il fournit à l'utilisateur certaines informations qu'une approche par motifs séquentiels ne peut pas directement fournir. Effectivement, les motifs partiellement ordonnés permettent de résumer efficacement les données temporelles ou séquentielles. L'extraction de tels motifs est cependant plus complexe que l'extraction de motifs séquentiels et nécessite la définition de nouvelles méthodes adaptées. Du fait de l'explosion combinatoire qu'implique leur extraction, l'utilisation de certaines propriétés sur ces motifs, pour en réduire le nombre, est pertinente, voire nécessaire. De telles propriétés permettent d'extraire un ensemble réduit de motifs, sans perte d'information. Même si les motifs partiellement ordonnés à eux seuls permettent de résumer efficacement les données, cette thèse est également motivée par d'autres aspects induits par l'extraction de tels motifs : (1) *peut-on efficacement filtrer un sous-ensemble intéressant et non redondant de l'information extraite ?* et (2) *peut-on utiliser la notion de structure partiellement ordonnée pour définir une vue synthétique unique d'une base de données séquentielles ?*

Ce travail répond à un besoin concret d'application dans le cadre de l'hydrobiologie. Les données collectées dans ce domaine sont conséquentes et sont principalement temporelles. Les experts du domaine, les hydrobiologistes, s'intéressent notamment aux liens existant entre deux catégories de données, les données concernant la biologie et celles concernant la physico-chimie, et cherchent à évaluer l'impact de la physico-chimie sur la dimension biologique. La complexité, l'hétérogénéité et le volume des données impliquées en font un cas d'application idéal pour la validation des méthodes proposées dans ce manuscrit.

1.2 Contributions

En accord avec les motivations et objectifs définis ci-avant, les contributions méthodologiques de cette thèse sont les suivantes :

Un algorithme d'extraction de motifs partiellement ordonnés clos. Nous présentons un nouvel algorithme, *OrderSpan*, pour l'extraction de motifs partiellement ordonnés, et plus spécifiquement de motifs partiellement ordonnés clos (également appelés motifs fermés). Nous proposons ainsi le premier algorithme qui permet d'extraire directement ce type de motifs dans n'importe quel type de base de données de séquences d'itemsets. De plus, nous montrons comment adapter simplement cet algorithme au cas de bases de données de séquences divisées en classes pour extraire seulement des motifs discriminants, i.e. des motifs qui sont plus fréquents dans une classe que dans les autres.

Une méthode pour filtrer k motifs selon plusieurs mesures d'intérêt. Le nombre de motifs extraits avec *OrderSpan*, et plus généralement avec les méthodes d'extraction de motifs, peut être volumineux. Bien qu'il existe des méthodes pour extraire ou filtrer les motifs selon un critère d'intérêt, l'utilisateur peut vouloir les filtrer selon une combinaison de plusieurs critères sans en privilégier un en particulier. Nous présentons pour cela une méthode itérative, qui permet de filtrer k motifs selon plusieurs dimensions d'intérêt pour l'utilisateur.

Un algorithme d'extraction de consensus discriminants. Plutôt que d'extraire un ensemble de motifs qui peut être volumineux, il peut être intéressant de proposer une méthode qui extrait une seule structure résumant toute une base de données. Nous proposons une approche fondée sur la notion de motifs partiellement ordonnés mais qui ne s'interprète pas comme telle. Cette méthode résume un ensemble de motifs séquentiels couvrant une base de données par une structure de données nommée consensus. Comme pour *OrderSpan*, cette approche est adaptée au cas de bases de données de séquences possédant plusieurs classes et permet ainsi d'extraire, pour chaque classe, un consensus discriminant, i.e., seule l'information spécifique à une classe est résumée dans le consensus.

Une mesure de dissimilarité entre séquences d'itemsets. L'approche par consensus, bien qu'étant une alternative possible à l'extraction de motifs,

est également intéressante dans le cadre de méthodes de *clustering* (ou de classification). En effet, le *clustering* consiste à créer des classes de manière non supervisée dans un jeu de données. Extraire un consensus pour chaque classe permet alors de les résumer et d'identifier les différences entre celles-ci. Une telle approche est intéressante en particulier lorsque le jeu de données est composé de milliers d'instances. Cependant, le *clustering* de séquences symboliques nécessite une mesure de distance adéquate. Nous avons adapté une mesure de dissimilarité très utilisée en traitement de séries temporelles, qui est tout à fait pertinente dans le cas de séquences d'itemsets représentant des données environnementales.

Cette thèse est financée dans le cadre du projet ANR11 MONU 14[5] Fresqueau[6]. Ce projet original vise à fournir aux hydrobiologistes de nouveaux outils d'analyse basés sur la fouille de données, qui ont pour but d'être utilisés comme des approches complémentaires aux méthodes statistiques, couramment utilisées dans ce domaine. Toutes les méthodes de fouille de données présentées dans ce manuscrit ont été appliquées à ces données réelles et les informations extraites ont été analysées par des experts du domaine représentés par les laboratoires de recherche TETIS[7] et LIVE[8], ainsi que les bureaux d'études Aquascop[9] et Aquabio[10]. De plus, une partie des approches a été implantée dans un logiciel de visualisation destiné à la restitution des données et des résultats de la fouille.

1.3 Plan du manuscrit

Le chapitre 2 présente un état de l'art sur les méthodes de fouille de motifs temporels, qui sont la base de cette thèse. Les motifs considérés sont les motifs séquentiels et les motifs partiellement ordonnés.

Les chapitres 3, 4 et 5 présentent les principaux apports méthodologiques. Le chapitre 3 introduit l'algorithme *OrderSpan*, qui est capable d'extraire directement la totalité des motifs partiellement ordonnés clos à partir d'une base de données de séquences. Un algorithme pour la sélection d'un sous-ensemble de k motifs est présenté dans le chapitre 4. Cette approche permet de combiner différents critères

5. Agence nationale de la recherche
6. http ://engees-fresqueau.unistra.fr/
7. http ://www.tetis.fr/
8. http ://imaville.u-strasbg.fr/
9. http ://www.aquascop.fr/
10. http ://www.aquabio-conseil.fr/

pour restituer un sous-ensemble de motifs. Le chapitre 5 s'intéresse à une méthode qui permet de résumer un jeu de données de séquences au travers d'un consensus partiellement ordonné discriminant.

L'application de ces différentes méthodes aux données hydrobiologiques est traitée dans les chapitres 6 et 7. Le chapitre 6 introduit le contexte hydrobiologique. Le chapitre 7 présente les différents résultats obtenus, leur analyse par les experts du domaine ainsi que l'implantation des différentes méthodes dans un logiciel de visualisation destiné aux hydrobiologistes.

Nous concluons cette thèse dans le chapitre 8, en présentant les multiples perspectives associées aux méthodologies proposées.

CHAPITRE 2

État de l'art sur la fouille de motifs temporels

Contents

2.1 Introduction

La découverte de motifs est un des axes de recherche les plus importants en fouille de données et repose sur la mise en évidence d'informations récurrentes dans des données. Les motifs couvrent un champ très large d'applications et ne sont pas réservés aux bases de données temporelles. En effet, la notion de motif a été introduite dans [AS94] avec les règles d'association. Dans le langage usuel, il existe plusieurs définitions pour un motif. Dans le dictionnaire Larousse, la définition qui s'en rapproche le plus est *Dessin, ornement, le plus souvent répété, sur un support quelconque*. Cela s'applique ici à l'art, au dessin et à la peinture. Dans le cadre de cette thèse et de manière plus générale dans le cadre de la fouille de données, un motif peut être défini comme une *Information, le plus souvent répétée et présentant un intérêt particulier, repérée dans une base de données*. Cette définition est très générale et un motif peut prendre différentes formes.

L'objectif de la fouille de motifs est d'aider l'utilisateur à améliorer sa compréhension des données en identifiant les comportements nouveaux, inattendus. Il existe de nombreux challenges inhérents à ce domaine :

1. Les bases de données étant de plus en plus volumineuses, les techniques de fouille de motifs sont soumises à des contraintes fortes telles que le temps de calcul ainsi que la consommation mémoire.
2. Très souvent le nombre de motifs extraits est particulièrement volumineux. Tous les motifs ne sont pas nécessairement intéressants pour l'utilisateur. Il est nécessaire de proposer des approches qui se focalisent sur la réduction du nombre de motifs extraits. Cette réduction peut s'effectuer grâce à des propriétés mathématiques, e.g. motifs clos, ou sur l'aspect qualitatif des motifs en se basant sur les besoins de l'utilisateur.
3. Les données devenant de plus en plus complexes, il est important d'extraire de nouveaux types de motifs au travers d'algorithmes adaptés.

Nous nous sommes intéressés à la fouille de motifs dans les bases de données temporelles divisées en classes, et en particulier à l'extraction de motifs partiellement ordonnés. L'utilisation de ces motifs est la motivation de cette thèse et comme nous allons le voir, ils restent actuellement peu étudiés dans le contexte de bases de données de séquences.

Dans ce chapitre, nous présentons tout d'abord des définitions générales sur les motifs. Nous étudions ensuite deux types de motifs temporels : les motifs séquentiels

et les motifs partiellement ordonnés. Nous décrivons également la méthode d'extraction de motifs séquentiels basée sur le paradigme *Pattern-Growth* avec projection de la base de données. Les motifs partiellement ordonnés et la méthode d'extraction expliquée dans ce chapitre sont la base de notre proposition dans le chapitre 3. Nous terminons avec le problème de la fouille de motifs dans le cadre de bases de données multiples, avec l'introduction d'une mesure statistique adaptée à cette problématique. Elle permet d'identifier des motifs dits discriminants. Cette dernière problématique est ensuite reprise et utilisée dans les chapitres méthodologiques 4 et 5.

2.2 Premières définitions sur les motifs

Il existe à ce jour une grande diversité de techniques de recherche de motifs adaptées à un contexte ou des données particulières. Nous pouvons par exemple citer la recherche d'itemsets [AS94, GZ04], de séquences [AS95, PHMA+04, MTP05], d'arbres [dJBC10] ou de graphes [WM03, CF06] ainsi que leurs nombreuses variantes appliquées au domaine spatial [LMP03, LHK+07, SCA+11], spatio-temporel [TG01, HZZ08, ASBF+12] ou multi-dimensionnel [PHP+01, PLL+10]. D'autres travaux utilisent des propriétés particulières des motifs pour ne conserver qu'une partie des résultats comme l'extraction de motifs par contraintes [GRS99, PHW07] ou bien l'extraction de motifs clos [PBTL99, YHN06] et maximaux [GZ01]. Ces approches sont fondées sur l'idée qu'un ensemble de propriétés fréquentes peuvent être utilisées comme une représentation d'un ensemble d'objets ou d'éléments. Les motifs recherchés peuvent être de nature différente mais suivent tous un même principe, qui est le fait qu'un motif soit identifiable dans un objet d'une base de données. Cette notion de présence, lorsqu'elle se réfère à plusieurs objets différents, induit la notion de fréquence d'un motif, c'est-à-dire à quel point un motif est répété sur la totalité des objets de cette base. Ainsi, plus un motif a une fréquence élevée, plus ce motif est représentatif de la base de données.

Pour illustrer simplement l'idée de motif, considérons la base de données du tableau 2.1. Celle-ci est composée de cinq lignes, où chaque ligne correspond à un ensemble d'achats effectués par un client (un panier).

En analysant cette base de données, nous observons par exemple que tous les clients ont acheté un *magazine*. Ainsi, le produit *magazine* peut être considéré comme un motif simple composé d'un seul achat, ou plus couramment appelé item. Cet item est présent dans chaque panier d'achats. Il est alors identifiable pour 100%

Client	Achats
C_1	céréales, magazine
C_2	chewing-gum, magazine, stylo
C_3	magazine, stylo
C_4	chewing-gum, magazine
C_5	céréales, magazine, stylo

TABLE 2.1 – Base de données d'achats

des clients. Nous pouvons également observer des motifs plus complexes composés d'ensembles d'achats comme {*magazine, stylo*} qui lui apparaît dans les paniers des clients C_2, C_3 et C_5. Ce motif est donc identifié pour 60% des clients. Comme nous le voyons, la fouille de motifs consiste à chercher des sous-ensembles d'informations qui sont représentatifs de sous-ensembles d'objets d'une base de données.

Nous donnons maintenant une définition plus formelle des motifs dans une base de données en nous basant sur la définition proposée dans [Rab11] et qui s'inspire des notations utilisées en analyse formelle de concept (AFC) [BM70, GW99]. Cette définition est valable pour tous les types de motifs. Supposons une base de données qui contient un ensemble de n objets $\mathcal{O} = \{O_1, O_2, \ldots, O_n\}$ décrits par un ensemble d'informations quelconques (attributs, séquences, graphes, etc.). À partir de cet ensemble d'objets, supposons maintenant un ensemble de p motifs $\mathcal{M} = \{M_1, M_2, \ldots, M_p\}$ identifiables dans ces objets. Nous appelons cet ensemble d'objets muni d'un ensemble de motifs un environnement d'extraction (définition 1).

Définition 1 ***(Environnement d'extraction)*** *On appelle* ***environnement d'extraction*** *le triplet $\mathcal{K} = (\mathcal{O}, \mathcal{M}, \mathcal{R})$, où $\mathcal{O} = \{O_1, O_2, \ldots, O_n\}$ est un ensemble de n* ***objets***, *$\mathcal{M} = \{M_1, M_2, \ldots, M_p\}$ est un ensemble de p* ***motifs*** *et $\mathcal{R}$ une* ***relation de support binaire*** *entre $\mathcal{O}$ et $\mathcal{M}$ tel que $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{M}$. Un objet $O \in \mathcal{O}$* ***supporte*** *un motif $M \in \mathcal{M}$ (ou M est supporté par O) s'il existe un couple (O, M) tel que $(O, M) \in \mathcal{R}$.*

Cette définition, bien que proche de celle de contexte formel en AFC, diffère de celle-ci par le fait qu'en analyse formelle de concept on ne parle pas de motifs mais d'attributs, où le but est d'identifier les sous-ensembles maximaux d'objets qui partagent ou ont en commun des sous-ensembles maximaux d'attributs. L'AFC traite de l'extraction de concepts formels dans le cadre de la théorie de Galois. Dans

notre cas, l'environnement d'extraction est utilisé comme cadre aux définitions qui suivent. Ainsi ici, la relation de support binaire permet d'identifier quels motifs apparaissent dans quels objets.

Pour illustrer cette définition, prenons une base de données $\mathcal{D}$ composée de dix objets tel que $\mathcal{D} = \{O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8, O_9, O_{10}\}$ et un ensemble $\mathcal{M}$ de six motifs tel que $\mathcal{M} = \{M_1, M_2, M_3, M_4, M_5, M_6\}$. L'environnement d'extraction donné par le tableau 2.2 indique le fait qu'un motif est supporté ou non par les objets de $\mathcal{D}$.

	M_1	M_2	M_3	M_4	M_5	M_6
O_1	×		×	×		×
O_2	×	×			×	×
O_3		×	×			
O_4	×		×		×	×
O_5	×			×		
O_6		×	×		×	
O_7		×			×	×
O_8				×		×
O_9	×		×		×	
O_{10}			×		×	

TABLE 2.2 – Exemple d'environnement d'extraction

Dans cet environnement d'extraction, lorsqu'un objet O_j supporte un motif M_i, la case à l'intersection de la ligne j et de la colonne i est marquée d'une croix ×. Ainsi, nous voyons par exemple que l'objet O_1 supporte les motifs M_1, M_3, M_4 et M_6 et que le motif M_2 est supporté par les objets O_2, O_3, O_6 et O_7.

Il est maintenant possible de définir le support d'un motif dans une base de données, qui correspond au nombre d'objets qui le supportent (définition 2).

Définition 2 ***(Support d'un motif)*** *Soit $\mathcal{K} = (\mathcal{O}, \mathcal{M}, \mathcal{R})$ un environnement d'extraction. Le* ***support*** *d'un motif $M \in \mathcal{M}$ dans $\mathcal{O}$ est égal au nombre d'objets $O \in \mathcal{O}$ tel que O et M sont en relation par $\mathcal{R}$:*

$$Support(M) = |\{O \in \mathcal{O} | (O, M) \in \mathcal{R}\}| \tag{2.1}$$

Reprenons l'environnement d'extraction donné par le tableau 2.2, le support du motif M_2 est $Support(M_2) = |\{O_2, O_3, O_6, O_7\}| = 4$ et le support du motif M_5 est

$Support(M_5) = |\{O_2, O_4, O_6, O_7, O_9, O_{10}\} = 6$. Le support donne le nombre exact d'objets qui supportent un motif, et il est également courant d'utiliser la fréquence plutôt que le support. La définition de la fréquence est donnée par la définition 3.

Définition 3 ***(Fréquence d'un motif)*** *Soit $\mathcal{K} = (\mathcal{O}, \mathcal{M}, \mathcal{R})$ un environnement d'extraction. La* ***fréquence*** *d'un motif $M \in \mathcal{M}$ dans $\mathcal{O}$ est définie de la manière suivante :*

$$Frequence(M) = \frac{Support(M)}{|\mathcal{O}|}$$

Ainsi, les fréquences des motifs M_2 et M_5 dans $\mathcal{D}$ sont :
- $Frequence(M_2) = \frac{Support(M_2)}{|\mathcal{D}|} = \frac{4}{10} = 40\%$
- $Frequence(M_5) = \frac{Support(M_5)}{|\mathcal{D}|} = \frac{6}{10} = 60\%$.

Généralement, dans les cas d'applications réels, extraire la totalité des motifs à partir d'une base de données entraîne une explosion combinatoire. Ainsi l'extraction des motifs est souvent limitée par un seuil de support minimum, noté θ, qui contraint l'extraction aux motifs qui ont un support supérieur à θ. Cela revient à stipuler une requête du type : "nous souhaitons extraire tous les motifs qui sont supportés par au minimum k objets de la base de données". Si nous reprenons l'exemple précédent, nous observons que tous les motifs $M \in \mathcal{M}$ ont un support supérieur ou égal à 3 dans $\mathcal{D}$, ou de manière équivalente une fréquence supérieure ou égale à 30% dans $\mathcal{D}$. Un motif dans une base de données est considéré comme fréquent si son support dépasse un seuil minimum θ fixé par l'utilisateur (définition 4).

Définition 4 ***(Motif fréquent)*** *Soient $\mathcal{K} = (\mathcal{O}, \mathcal{M}, \mathcal{R})$ un environnement d'extraction et θ une valeur entière tel que $0 < \theta \leq |\mathcal{O}|$. Un motif $M \in \mathcal{M}$ est un* ***motif fréquent*** *dans $\mathcal{O}$ si et seulement si $Support(M) \geq \theta$. θ est alors appelé seuil de* ***support minimum****.*

Nous allons maintenant présenter le problème de la recherche de motifs dans le cas des données séquentielles. Dans la section 2.3, nous étudions deux types de motifs extraits à partir de bases de données de séquences, les motifs séquentiels et les motifs partiellement ordonnés.

2.3 Motifs extraits à partir de données séquentielles

Les motifs tels que présentés dans l'exemple de la base de données du tableau 2.1 sont étudiés dans le cadre de la recherche d'itemsets (ensembles d'items) fréquents [AS94, GZ04]. Bien que pertinents dans les bases de données où les objets sont représentés par des ensembles, ils ne peuvent pas être extraits à partir de bases de données plus complexes composées de graphes ou bien de séquences. C'est cette dernière problématique que nous allons étudier. Les données séquentielles sont présentes dans de nombreux domaines comme le marketing [GB12], le génie logiciel [RWD+09] ou bien les données de santé [PCKR09, FBP+11, SPB+11]. Ces données peuvent être représentées par des séquences. Une séquence est une structure de données qui permet d'organiser un ensemble d'éléments grâce à une relation d'ordre entre ces éléments. Par exemple, cet ordre est défini par la temporalité dans le cas de données temporelles ou bien par la valeur d'expression des gènes dans le cas de données génomiques [FBP+11].

Pour illustrer cette notion, reprenons l'exemple d'achats effectués par un client. Le tableau 2.3 nous donne, pour un seul client, la liste des achats qu'il a effectués à différentes dates.

Date	Achats
11/10	céréales, lait
12/10	chewing-gum
20/10	magazine, stylo, cahier
01/11	chocolat

TABLE 2.3 – Liste d'achats d'un client

Dans cette liste, le client a par exemple acheté des céréales et du lait le 11 octobre, et du chocolat le 1er novembre. Nous pouvons ainsi construire une séquence temporelle d'achats pour ce client : ⟨(céréales,lait)(chewing-gum)(magazine,stylo,cahier)(chocolat)⟩. La notion d'ordre est ici définie par la temporalité. Soient deux dates d_α et d_β, si $d_\alpha < d_\beta$, alors le ou les achats qui correspondent à d_α sont positionnés, dans la séquence, avant le ou les achats qui correspondent à d_β.

Bien qu'il existe plusieurs types de séquences, nous nous intéressons aux séquences d'itemsets (cf. l'exemple précédent), où ce sont des ensembles d'éléments, ou items, qui sont ordonnés et non des éléments seuls. Ainsi, nous définissons tout d'abord la notion d'itemset (définition 5).

Définition 5 ***(Itemset)*** *Soit $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ un ensemble d'**items**. Un **itemset** IS est un ensemble non vide et non ordonné d'**items** noté $(I'_1, \ldots, I'_k)$ où $I'_i \in \mathcal{I}$.*

Par exemple, en reprenant la liste d'achats du tableau 2.3, l'ensemble des items est $\mathcal{I} = \{$céréales,lait,chewing-gum,magazine,stylo,cahier,chocolat$\}$ et l'ensemble des achats effectués le 20 octobre forme l'itemset (magazine,stylo,cahier). Nous pouvons maintenant définir les séquences composées d'itemsets (définition 6).

Définition 6 ***(Séquence)*** *Soit $\mathcal{IS}$ l'ensemble de tous les **itemsets** construits à partir d'un ensemble d'items $\mathcal{I}$. Une **séquence** S est une liste non vide d'**itemsets** notée $\langle IS_1 IS_2 \ldots IS_p \rangle$ où $IS_j \in \mathcal{IS}$.*

Nous illustrons cette définition par la base de données de séquences $\mathcal{D}$ donnée par le tableau 2.4. Cette base de données est composée de trois séquences S_1, S_2 et S_3 construites à partir de l'alphabet d'items $\mathcal{I} = \{a, c, d, e, f, g\}$.

ID	**Séquence**
S_1	$\langle (c,d)(a)(g)(d) \rangle$
S_2	$\langle (g)(c,d,e)(f)(a,e,g) \rangle$
S_3	$\langle (g)(d)(e)(f) \rangle$

TABLE 2.4 – Exemple de base de données de séquences

Prenons la séquence $S_1 = \langle (c,d)(a)(g)(d) \rangle$, l'itemset (c,d) précède l'itemset (g) lui même suivi par l'itemset (d). Cela signifie que $(c,d) < (g)$ et $(g) < (d)$. L'ordre dans une séquence est un **ordre total**. Ainsi, en prenant deux itemsets IS_α, IS_β au hasard dans une séquence, il est possible de déterminer si $IS_\alpha \leq IS_\beta$ ou $IS_\beta \leq IS_\alpha$. L'extraction de motifs à partir de telles bases de données est basée sur la notion de sous-séquence. Une sous-séquence S' d'une séquence S est une séquence de taille $|S'|$ avec $|S'| \leq |S|$ où l'ordre des éléments dans S' est identique à l'ordre des éléments dans S (définition 7).

Définition 7 ***(Sous-séquence)*** *Une **séquence** $S' = \langle IS'_1 IS'_2 \ldots IS'_p \rangle$ est une **sous-séquence** d'une autre **séquence** $S = \langle IS_1 IS_2 \ldots IS_m \rangle$, noté $S' \preceq_s S$, si $p \leq m$ et s'il existe des entiers $j_1 < j_2 < \ldots < j_k < \ldots < j_p$ tels que $IS'_1 \subseteq IS_{j_1}, IS'_2 \subseteq IS_{j_2}, \ldots, IS'_p \subseteq IS_{j_p}$.*

Prenons la séquence $S = \langle (g)(e)(f) \rangle$. Cette séquence est une sous-séquence des séquences S_2 et S_3 car dans S_2 et S_3 $(g) < (e)$, $(g) < (f)$ et $(e) < (f)$. Cette définition

des sous-séquences sert de base aux méthodes d'extraction de motifs séquentiels et de motifs partiellement ordonnés. Ces deux types de motifs se différencient par l'ordre défini sur les éléments. Un motif séquentiel se base sur un **ordre total** entre les itemsets (Sous-section 2.3.1) alors qu'un motif partiellement ordonné se base sur un **ordre partiel** entre les itemsets (Sous-section 2.3.3).

2.3.1 Motifs séquentiels

Les motifs séquentiels ont été introduits dans [AS95] comme une extension des itemsets fréquents. Il s'agit des sous-séquences fréquentes qui sont contenues dans une base de données de séquences. Dans la littérature, il existe de nombreux travaux qui ont étudié l'extraction ou bien l'utilisation de tels motifs. Tout d'abord, de nombreux algorithmes ont été proposés pour extraire de manière exhaustive tous les motifs séquentiels contenus dans une base de données de séquences [AS95, Zak01, AFGY02, PHMA$^+$04]. D'autres algorithmes se sont intéressés aux propriétés des motifs clos [YHA03, WH04] (étudiés dans la suite de ce chapitre), maximaux [RPT06, FVWT13] ou bien encore a une représentation condensée de ces motifs [PC09]. Également, bien qu'adaptés aux donnés temporelles, les motifs séquentiels ont été étendus pour capturer, en plus de l'information temporelle, de la connaissance spatiale [SBF$^+$12].

Les motifs séquentiels sont eux-mêmes des séquences associées à un support dans une base de données de séquences. Une séquence supporte un motif séquentiel si celui-ci en est une sous-séquence. Ainsi, dans le contexte d'un environnement d'extraction avec une base de données de séquences et un ensemble de motifs séquentiels, nous définissons la relation de support associée (définition 8) sur la base de la définition 7.

Définition 8 ***(Relation de support pour les motifs séquentiels)*** *Soit $\mathcal{K} = (\mathcal{S}, \mathcal{MS}, \mathcal{R})$ un environnement d'extraction avec $\mathcal{S}$ une base de données de séquences et $\mathcal{MS}$ un ensemble de motifs séquentiels. La relation de support $\mathcal{R}$ entre l'ensemble des séquences $\mathcal{S}$ et l'ensemble des motifs $\mathcal{MS}$ est définie de la manière suivante :*

$$\mathcal{R} = \{(S, MS) \in \mathcal{S} \times \mathcal{MS} | MS \preceq_s S\} \tag{2.2}$$

Cette définition nous permet d'utiliser ainsi toutes les définitions générales sur les motifs présentées dans la section 2.2, dont la notion de motif séquentiel fréquent.

Par exemple, avec une valeur de support minimum $\theta = 2$, l'ensemble des motifs séquentiels fréquents extraits à partir de la base de données $\mathcal{D}$ du tableau 2.4 est

donné par le tableau 2.5 avec pour chaque motif son support et sa fréquence. Comme il y a trois séquences dans la base de données exemple, dans le tableau les motifs séquentiels sont rassemblés en deux groupes, comprenant les motifs de support 3 et les motifs de support 2.

Motifs séquentiels	**Support**	**Fréquence**
$\langle(d)\rangle$,$\langle(g)\rangle$,$\langle(g)(d)\rangle$	3	100%
$\langle(a)\rangle$,$\langle(e)\rangle$,$\langle(f)\rangle$,$\langle(c)\rangle$,$\langle(c,d)\rangle$,$\langle(c)(a)\rangle$,$\langle(c)(g)\rangle$, $\langle(c,d)(a)\rangle$,$\langle(c,d)(g)\rangle$,$\langle(g)(e)\rangle$,$\langle(g)(f)\rangle$,$\langle(g)(e)(f)\rangle$, $\langle(g)(d)(f)\rangle$,$\langle(g)(d)(e)\rangle$	2	66.66%

TABLE 2.5 – Motifs séquentiels extraits à partir de la base de données du tableau 2.4 avec un support minimum $\theta = 2$

À un tel support minimum, plus de 17 motifs séquentiels sont extraits des séquences de $\mathcal{D}$. Par exemple, le motif $\langle(g)(d)\rangle$ est inclus dans la totalité des séquences alors que le motif $\langle(g)(f)\rangle$ est inclus dans les séquences S_2 et S_3. Le support de $\langle(g)(d)\rangle$ est donc de 3 et celui de $\langle(g)(f)\rangle$ de 2. Nous observons également que les motifs $\langle(g)(f)\rangle$ et $\langle(g)(e)(f)\rangle$ sont tous les deux inclus dans le même ensemble de séquences $\{S_2, S_3\}$ et que $\langle(g)(f)\rangle \preceq_s \langle(g)(e)(f)\rangle$. $\langle(g)(f)\rangle$ est donc redondant par rapport à $\langle(g)(e)(f)\rangle$ car : (1) les deux motifs sont supportés par le même ensemble de séquences et (2) $\langle(g)(e)(f)\rangle$ est plus spécifique que $\langle(g)(f)\rangle$. Pour sélectionner les motifs non-redondants dans un ensemble de motifs, nous introduisons la notion de motifs séquentiels clos [YHA03, WH04]. Ces motifs sont très utilisés car ils permettent de réduire considérablement le nombre de motifs séquentiels extraits sans perte d'information. Cette propriété est définie par la définition 9.

Définition 9 ***(Motifs séquentiels clos)*** *Un* ***motif séquentiel*** M_S *est* ***clos*** *s'il n'existe pas de* ***motif séquentiel*** M'_S *tel que* $M_S \preceq_s M'_S$ *avec* $Support(M_S) = Support(M'_S)$.

Appliqué à l'exemple précédent, nous obtenons l'ensemble des motifs séquentiels clos présenté dans le tableau 2.6.

Comme nous le voyons, cette propriété permet de passer de 17 motifs extraits à seulement 6. Cette différence entre le nombre de motifs séquentiels et de motifs séquentiels clos dépend du jeu de données sur lequel la méthode est appliquée.

Motifs séquentiels clos	**Support**	**Fréquence**
$\langle(g)(d)\rangle$	3	100%
$\langle(c,d)(a)\rangle$,$\langle(c,d)(g)\rangle$,$\langle(g)(e)(f)\rangle$, $\langle(g)(d)(f)\rangle$,$\langle(g)(d)(e)\rangle$	2	66.67%

TABLE 2.6 – Motifs séquentiels clos extraits à partir de la base de données du tableau 2.4 avec un support minimum $\theta = 2$

2.3.2 Extraction de motifs séquentiels

Brièvement énumérés en introduction de cette section, les algorithmes de fouille de motifs séquentiels, bien que basés sur des techniques différentes, suivent tous un même schéma général :

1. Extraire l'ensemble de toutes les séquences fréquentes de taille 1 (qui ne contiennent qu'un item).
2. À partir des séquences extraites à l'étape précédente, extraire toutes les séquences de taille $k + 1$.
3. L'étape précédente est ensuite réitérée récursivement jusqu'à l'extraction de tous les motifs séquentiels.

L'extraction des motifs séquentiels, et des motifs en général, s'arrête lorsque qu'il n'existe plus de motif à extraire dont le support est supérieur ou égal au seuil de support minimum fixé par l'utilisateur. Ce schéma est général et les méthodes utilisent ensuite différentes techniques pour générer les motifs de taille $k + 1$. Pour éviter de générer un motif plusieurs fois, les méthodes considèrent couramment l'ordre lexicographique des items dans les itemsets. Par exemple, l'itemset (a, c) (selon la définition 5) peut être représenté par (a, c) ou bien (c, a). Cependant, dans le cadre de l'extraction de motifs, seul l'itemset (a, c) sera généré.

Les premiers algorithmes [AS95, Zak01, AFGY02] exploitent par exemple le paradigme d'extraction *Générer-Élaguer*, qui consiste à générer, à partir d'un ensemble de motifs de taille k, tous les motifs possibles de taille $k + 1$ par génération combinatoire. Les motifs générés sont ensuite testés sur la base de données pour calcul de leur support. Si ce support est inférieur au seuil de support minimum, le motif est élagué, sinon il est stocké pour le calcul des motifs de taille $k + 2$.

Les approches de type *Générer-Élaguer* s'avèrent cependant coûteuses. Il n'est pas rare qu'un volume immense de motifs soit généré pour au final en élaguer la majorité. Ainsi, d'autres algorithmes [PHMA$^+$04, YHA03, WH04] se sont appuyés

sur le paradigme *Pattern-Growth* qui permet, à chaque étape, de générer seulement des motifs fréquents sans génération de motifs candidats.

Dans [PHMA+04], est proposé l'algorithme *PrefixSpan*. Les motifs sont générés selon le paradigme *Pattern-Growth* et la base de données est récursivement divisée par projection des séquences grâce aux propriétés des préfixes fréquents. Cette approche s'est avérée très performante en comparaison des approches de type *Générer-Élaguer*. C'est cette dernière méthode qui nous intéresse plus particulièrement et nous la détaillons maintenant.

Utilisation du paradigme Pattern-Growth avec projection par préfixes fréquents

Pour illustrer cette technique, prenons l'arbre de la figure 2.1. Celui-ci présente l'espace de recherche complet qui correspond à l'extraction des motifs séquentiels depuis la base de données du tableau 2.4, avec un support minimum θ de 2.

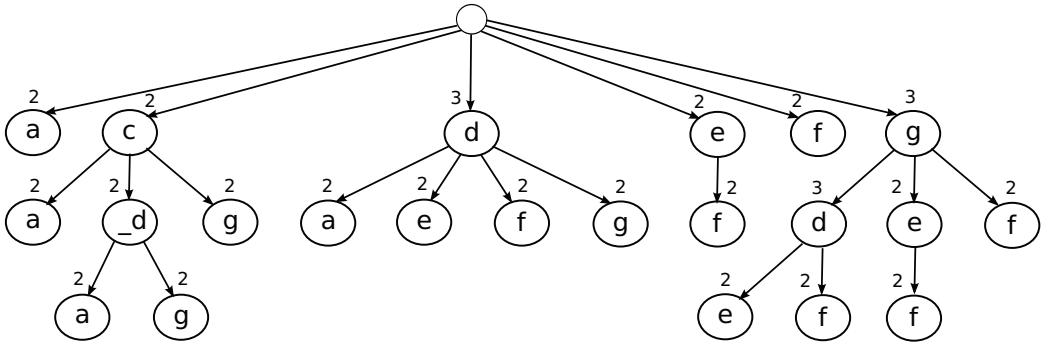


FIGURE 2.1 – Arbre de recherche des motifs séquentiels généré à partir de la base de données de séquences du tableau 2.4 avec un support minimum de $\theta = 2$

Chaque sommet représente un motif séquentiel et, en commençant par le sommet racine, il est possible d'énumérer l'ensemble complet des motifs séquentiels supportés par les séquences de la base de données avec un support minimum θ de 2. Le nombre apparaissant au dessus de chaque sommet est le support qui correspond au motif séquentiel représenté par ce même sommet. Par exemple, prenons le sous-espace de recherche qui commence par le sommet étiqueté par c, en dessous la racine. Il est possible d'en extraire les motifs séquentiels suivants avec leur support associé : $\langle(c)\rangle : 2$, $\langle(c,d)\rangle : 2$, $\langle(c,d)(a)\rangle : 2$, $\langle(c,d)(g)\rangle : 2$, $\langle(c)(a)\rangle : 2$ et $\langle(c)(g)\rangle : 2$. Dans cet ensemble, les motifs séquentiels clos sont : $\langle(c,d)(a)\rangle$ et $\langle(c,d)(g)\rangle$.

Cet espace de recherche est exploré grâce des opérations d'extension sur les séquences. Prenons une séquence S et un item α, l'opération $S \diamond \alpha$ concatène l'item

α à S. Cependant, la fouille de motifs séquentiels dans les bases de données de séquences d'itemsets nécessite deux types d'extensions, qui sont la *I-Extension* et la *S-Extension* [AFGY02].

L'opération de *I-Extension* concatène l'item α dans le dernier itemset de S (définition 10). Cette opération considère l'ordre lexicographique des items dans les itemsets, que nous notons $<_{lexico}$.

Définition 10 ***(I-Extension dans l'ordre lexicographique)*** *Soient une séquence $S = \langle IS_1 IS_2 ... \ IS_p \rangle$ et un item α, avec $\forall I \in IS_p, I <_{lexico} \alpha$. La I-Extension de S avec α est notée $S \diamond_i \alpha$, avec $S \diamond_i \alpha = \langle IS_1 IS_2 ... \ IS_p \cup (\alpha) \rangle$.*

Par exemple $\langle (c) \rangle \diamond_i d$ donne la nouvelle séquence $\langle (c, d) \rangle$.

L'opération de *S-Extension* concatène l'item α dans un nouvel itemset inséré après le dernier itemset S (définition 11).

Définition 11 ***(S-Extension)*** *Soient une séquence $S = \langle IS_1 IS_2 ... \ IS_p \rangle$ et un item α. La S-Extension de S avec α est notée $S \diamond_s \alpha$ avec $S \diamond_s \alpha = \langle IS_1 IS_2 ... \ IS_p(\alpha) \rangle$.*

Par exemple $\langle (c, d) \rangle \diamond_s g$ donne la nouvelle séquence $\langle (c, d)(g) \rangle$.

Dans la figure 2.1, l'opération de *I-Extension* est différenciée de l'opération de *S-Extension* par les sommets qui commencent par le symbole '_'.

À partir de ces opérations d'extensions entre une séquence et un item, nous les généralisons maintenant au cas de deux séquences. Soient deux séquences $S = \langle IS_1 IS_2 ... \ IS_p \rangle$ et $S' = \langle IS'_1 IS'_2 ... \ IS_m \rangle$, $S \diamond S'$ signifie que S est concaténée avec S', avec $S \diamond_i S' = \langle IS_1 IS_2 ... \ IS_p \cup IS'_1 IS'_2 ... \ IS_m \rangle$ et $S \diamond_s S' = \langle IS_1 IS_2 ... \ IS_p IS'_1 IS'_2 ... \ IS_m \rangle$. Cela nous permet maintenant de définir la notion de préfixe et de suffixe d'une séquence (définition 12).

Définition 12 ***(Préfixe et suffixe d'une séquence)*** *Soient trois séquences S, S' et S''. Si $S = S' \diamond S''$ alors S' est un **préfixe** de S et S'' est un **suffixe** de S.*

Par exemple $\langle (c) \rangle$ et $\langle (c, d) \rangle$ sont des préfixes de $\langle (c, d)(a)(g)(d) \rangle$. De la même manière, $\langle (d) \rangle$ et $\langle (g)(d) \rangle$ en sont des suffixes.

En utilisant ces définitions, le paradigme *Pattern-Growth* avec utilisation des préfixes fréquents permet de diviser récursivement la base de données avec un système de projections. De telles projections sont construites à partir d'un motif séquentiel fréquent précédemment extrait, appelé préfixe fréquent (l'initialisation de l'algorithme débute avec un préfixe vide). La projection d'une séquence selon un préfixe fréquent est donnée par la définition 13.

Définition 13 ***(Séquence projetée)*** *Soient S une séquence et M_S un motif séquentiel fréquent. La projection de S selon M_S est notée $S|_{M_S}$ tel que $S = S' \diamond S|_{M_S}$, avec S' étant le préfixe de taille minimale de S contenant M_S, i.e. $M_S \preceq_s S'$ et $\nexists S''$ tel que $M_S \preceq_S S''$ et $S'' \prec_s S'$.*

Par exemple, prenons la séquence $\langle(a,b)(b,c)(a,c)(b)\rangle$. Si nous projetons cette séquence selon un préfixe fréquent $\langle(a)(c)\rangle$, noté $\langle(a,b)(b,c)(a,c)(b)\rangle|_{\langle(a)(c)\rangle}$, cela consiste à identifier le suffixe qui succède la première apparition de la sous-séquence $\langle(a)(c)\rangle$ et qui est $\langle(a,c)(b)\rangle$. De la même manière, $\langle(a,b)(b,c)(a,c)(b)\rangle|_{\langle(a)\rangle} = \langle(\_b)(b,c)(a,c)(b)\rangle$ et $\langle(a,b)(b,c)(a,c)(b)\rangle|_{\langle(a)(b)\rangle} = \langle(\_c)(a,c)(b)\rangle$. Dans le dernier exemple, la projection s'effectue à l'item b du second itemset, et non du premier, car dans le préfixe fréquent l'item b succède séquentiellement l'item a. Cette dernière séquence projetée commence avec le symbole "_" pour pouvoir prendre en considération, ensuite, une possible I-Extension avec l'item c.

En généralisant à l'ensemble de séquences d'une base $\mathcal{D}$, il est ainsi possible de construire une projection $\mathcal{D}|_{M_S}$ à partir d'un préfixe fréquent M_S. Soit $\mathcal{D}$ la base de données du tableau 2.4, $\mathcal{D}|_{\langle(g)\rangle} = \{\langle(d)\rangle, \langle(c,d,e)(f)(a,e,g)\rangle, \langle(d)(e)(f)\rangle\}$ et $\mathcal{D}|_{\langle(g)(d)\rangle} = \{\$, \langle(\_e)(f)(a,e,g)\rangle, \langle(e)(f)\rangle\}$. Pour cette dernière projection, le symbole \$ signifie que le préfixe fréquent $\langle(g)(d)\rangle$ est inclus dans la première séquence de la base de données, mais que le suffixe obtenu par la projection est une séquence vide.

En parcourant les séquences projetées de cette base de données projetée, nous trouvons les items e et f, également appelés occurrences, avec un support égal à 2. Ce qui permet ensuite de récursivement projeter $\mathcal{D}$ avec les préfixes fréquents $\langle(g)(d)(e)\rangle$ et $\langle(g)(d)(f)\rangle$. Nous obtenons finalement les bases projetées $\mathcal{D}|_{\langle(g)(d)(e)\rangle} = \{\langle\rangle, \langle(\_g)\rangle, \langle(f)\rangle\}$ et $\mathcal{D}|_{\langle(g)(d)(f)\rangle} = \{\langle\rangle, \langle(a,e,g)\rangle, \$\}$.

L'efficacité de cette méthode est due à la division récursive de la base de données, que permet l'utilisation des préfixes fréquents. D'autres optimisations, comme la pseudo-projection de la base de données, influent de manière importante sur le temps de calcul et la consommation mémoire. Elles ne sont pas présentées ici mais elles sont détaillées dans l'article de *PrefixSpan* [PHMA+04].

2.3.3 Motifs partiellement ordonnés

En plus des motifs séquentiels, il est possible d'extraire un autre type de motif à partir de données séquentielles : les motifs partiellement ordonnés. Ils ont été introduits plus récemment que les motifs séquentiels et ont été particulièrement utilisés dans la fouille d'épisodes fréquents [MTV97, ZLC10, TC12]. Cependant, ces

approches permettent d'extraire des motifs partiellement ordonnés dans une seule et longue séquence. Leur utilisation dans le cadre de bases de données de séquences a par contre été étudiée par [CG05, PWL+06].

Dans [CG05], l'auteur présente un premier algorithme pour l'extraction de motifs partiellement ordonnés clos dans les bases de données séquentielles. Cet algorithme est divisé en deux phases : (1) un algorithme d'extraction de motifs séquentiels clos, tels que *CloSpan* [YHA03] ou *BIDE* [WH04], est tout d'abord utilisé ; et (2) une opération de post-traitement est effectuée pour transformer des ensembles de motifs séquentiels clos qui sont supportés par un même ensemble de séquences en motifs partiellement ordonnés.

La fouille de bases de données de chaînes de caractères a été étudiée dans [PWL+06]. Les auteurs ont proposé l'algorithme *Frecpo* qui permet d'extraire efficacement les motifs partiellement ordonnés clos. La méthode consiste tout d'abord à transformer chaque séquence de la base de données en un ensemble d'arcs entre les items grâce à la propriété de fermeture transitive sur les séquences. Par exemple, la séquence $\langle(c)(d)(f)\rangle$ est transformée en ensemble d'arcs $\{(c,d)(c,f)(d,f)\}$. Ensuite, à partir de cette base transformée, les ensembles d'arcs fréquents clos sont extraits. Ces ensembles peuvent alors facilement être transformés en motifs partiellement ordonnés clos.

Cependant, ces deux méthodes ont certaines limites qui sont discutées dans le chapitre 3. Elles sont la motivation de la méthode que nous proposons, qui vise a extraire efficacement l'ensemble complet des motifs partiellement ordonnés clos.

Extraire ces motifs est plus difficile que d'extraire des motifs séquentiels, car cela entraîne l'exploration d'un espace de recherche plus important. Alors que chaque motif séquentiel est une sous-séquence fréquente de la base de données, un motif partiellement ordonné rend compte d'un ensemble de sous-séquences fréquentes supportées par un même ensemble de séquences. Par exemple, reprenons la base de données de séquences et les motifs extraits précédemment. Les motifs séquentiels clos $\langle(c,d)(a)\rangle$ et $\langle(c,d)(g)\rangle$ sont tous les deux supportés par les mêmes séquences $S_1 = \langle(c,d)(a)(g)(d)\rangle$ et $S_2 = \langle(g)(c,d,e)(f)(a,e,g)\rangle$. Ces motifs co-existent alors dans la base de données. À moins d'effectuer un post-traitement potentiellement coûteux sur les motifs séquentiels extraits, il n'est pas possible, avec les méthodes d'extraction de motifs séquentiels classiques, de savoir si deux motifs séquentiels co-existent dans une base de données. Cependant, les motifs partiellement ordonnés permettent de capturer cette information en regroupant les motifs qui sont supportés par les mêmes ensembles de séquences. En effet, en analysant les motifs séquentiels

$\langle (c,d)(a) \rangle$ et $\langle (c,d)(g) \rangle$, nous observons que $(c,d) < (a)$ et que $(c,d) < (g)$, mais les itemsets (a) et (g) sont ordonnés différemment dans S_1 et S_2. Si l'on considère les séquences S_1 et S_2, les itemsets qui composent ces deux motifs sont donc partiellement ordonnés. Pour résumer cette information, nous utilisons une séquence partiellement ordonnée (représentée par un graphe acyclique orienté) :

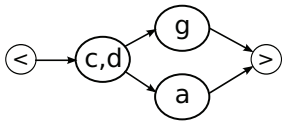


FIGURE 2.2 – Exemple de séquence partiellement ordonnée qui résume les motifs $\langle (c,d)(a) \rangle$ et $\langle (c,d)(g) \rangle$

Les séquences partiellement ordonnées ont un sommet de début et un sommet de fin qui correspondent aux chevrons $<$ et $>$ usuellement utilisés dans la notation des séquences telles que définies par la définition 6. La définition 14 définit les séquences partiellement ordonnées.

Définition 14 ***(Séquence partiellement ordonnée)***
Une ***séquence partiellement ordonnée*** *est un ensemble d'****itemsets*** *qui peut être représenté par un* ***graphe acyclique orienté*** $G = (\mathcal{V}, \mathcal{A}, \Sigma_\mathcal{V}, l_\mathcal{V})$ *munis d'un ordre partiel* $(\mathcal{V}, \leq)$*, où :*

- $\mathcal{V}$ *est l'ensemble des* ***sommets*** *et* $\mathcal{A}$ *est l'ensemble des* ***arcs*** *où* $\mathcal{A} = \{(u,v) \in < \text{ avec } u, v \in \mathcal{V}\}$
- $\Sigma_\mathcal{V}$ *est un alphabet composé d'****itemsets*** *représentant les valeurs possibles des* ***sommets***
- $l_\mathcal{V} : \mathcal{V} \rightarrow \Sigma_\mathcal{V}$ *est la* ***fonction*** *donnant l'étiquette des* ***sommets***
- *Pour chaque paire de sommets* $u, v \in \mathcal{V}$*,* $u < v$ *s'il y a un* ***chemin*** *de* u *à* v*. Par contre, s'il n'y pas de* ***chemin*** *de* u *à* v *ou de* v *à* u*, alors ces* ***itemsets*** *ne sont pas comparables*
- *Chaque* ***chemin*** *dans le graphe est une* ***séquence*** *telle que définie par la définition 6. L'ensemble des chemins de* G *est écrit* $\mathcal{P}_G$

Pour pouvoir utiliser la notion de séquence partiellement ordonnée comme motif, nous définissons maintenant la notion d'inclusion d'une séquence partiellement ordonnée dans une séquence d'une base de données (définition 15).

Définition 15 ***(Inclusion d'une séquence partiellement ordonnée dans une séquence)***

*Soient G une **séquence partiellement ordonnée**, $\mathcal{P}_G$ l'ensemble des chemins dans G et S une **séquence**. G est incluse dans S si $\forall P_G \in \mathcal{P}_G, P_G \preceq_s S$. Cette inclusion est notée $G \preceq_{gs} S$.*

Cette définition s'illustre facilement avec la séquence partiellement ordonnée de la figure 2.2 où les deux chemins, $\langle(c,d)(a)\rangle$ et $\langle(c,d)(g)\rangle$ sont des sous-séquences de S_1 et S_2. Cette séquence partiellement ordonnée est incluse dans les séquences S_1 et S_2. Comme pour la définition de la relation de support dans le contexte des motifs séquentiels, nous définissons la relation de support associée aux motifs partiellement ordonnés (définition 16).

Définition 16 ***(Relation de support pour les motifs partiellement ordonnés)*** *Soit $\mathcal{K} = (\mathcal{S}, \mathcal{MG}, \mathcal{R})$ un environnement d'extraction avec $\mathcal{S}$ une base de données de séquences et $\mathcal{MG}$ un ensemble de motifs partiellement ordonnés. La relation de support $\mathcal{R}$ entre l'ensemble des séquences $\mathcal{S}$ et l'ensemble des motifs $\mathcal{MG}$ est définie de la manière suivante :*

$$\mathcal{R} = \{(S, MG) \in \mathcal{S} \times \mathcal{MG} | MG \preceq_{gs} S\} \tag{2.3}$$

Nous donnons dans la figure 2.3 un ensemble (non exhaustif) de motifs partiellement ordonnés supportés par les séquences S_1 et S_2. Ici nous ne donnons pas l'ensemble des motifs partiellement ordonnés possibles avec un support minimum $\theta = 2$ car cet ensemble est trop volumineux. Nous voyons qu'il est possible d'extraire de nombreux motifs partiellement ordonnés à partir de ces deux séquences. L'espace de recherche de ces motifs est bien plus important que celui des motifs séquentiels et consiste à rechercher les combinaisons de motifs séquentiels possibles pour obtenir des motifs partiellement ordonnés. Par exemple, le motif de la figure 2.3e est supporté par les séquences S_1 et S_2 car les deux séquences qui composent ce motif partiellement ordonné, $\langle(a)\rangle$ et $\langle(g)(d)\rangle$, sont toutes les deux supportées par S_1 et S_2 (définition 7).

Il est cependant possible d'avoir différentes représentations équivalentes lorsque l'on cherche à synthétiser un ensemble de motifs séquentiels dans une séquence partiellement ordonnée. Les représentations qui nous intéressent sont les représentations minimales, c'est-à-dire celles qui synthétisent un ensemble de motifs séquentiels en un nombre de sommets et d'arcs minimaux. Prenons le motif partiellement ordonné M_d de la figure 2.3d, celui-ci n'est pas minimal car le chemin $\langle(d)\rangle$ est inclus dans les chemins $\langle(c,d)(g)\rangle$ et $\langle(c,d)(a)\rangle$. Une représentation minimale de M_d est donnée

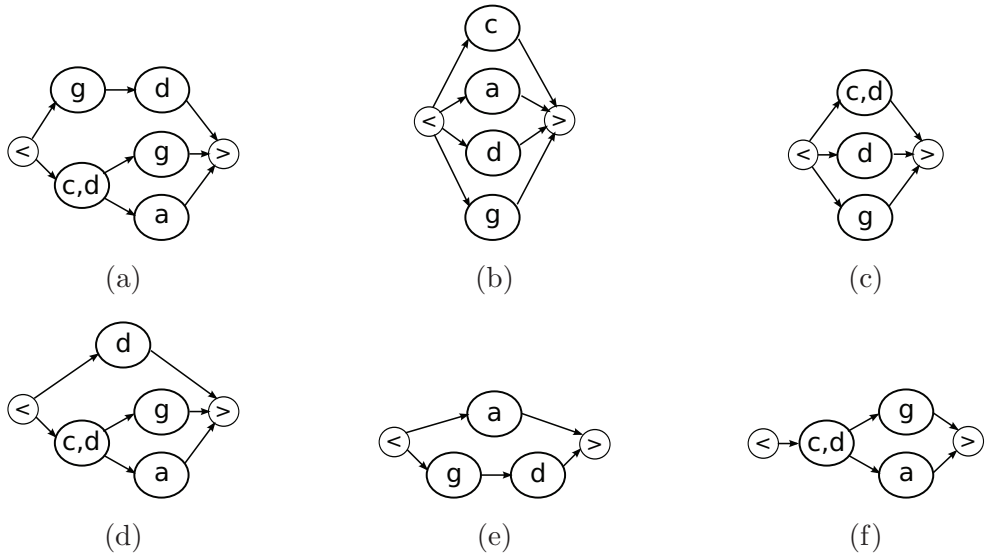


FIGURE 2.3 – Exemple de motifs partiellement ordonnés supportés par les séquences S_1 et S_2

par le motif M_f de la figure 2.3f. Pour définir cette notion de séquence partiellement ordonnée minimale, nous donnons la définition de sous-séquence partiellement ordonnée (définition 17).

Définition 17 ***(Sous-séquence partiellement ordonnée)***
Soient G et G' deux ***séquences partiellement ordonnées*** *avec $\mathcal{P}_G$ et $\mathcal{P}'_G$ leurs ensembles de chemins respectifs. G' est* ***une sous-séquence partiellement ordonnée*** *de G, noté $G' \preceq_g G$, si $\forall P'_G \in \mathcal{P}'_G, \exists P_G \in \mathcal{P}_G$ tel que $P'_G \preceq_s P_G$.*

Par exemple, prenons les motifs partiellement ordonnés des figures 2.3b et 2.3d, notés M_b et M_d respectivement. M_b est une sous-séquence partiellement ordonnée de M_d, car pour chaque chemin dans M_b ($\langle(a)\rangle$, $\langle(c)\rangle$, $\langle(d)\rangle$ et $\langle(g)\rangle$), il existe une séquence dans M_d qui l'inclue, donc $M_b \preceq_g M_d$.

À partir de cette définition, nous voyons que M_d et M_f sont équivalents car $M_d \preceq_g M_f$ et $M_f \preceq_g M_d$. Nous notons cette équivalence $M_d \equiv_g M_f$. Mais le nombre de sommets dans M_d est supérieur au nombre de sommets dans M_f. Cela nous amène à la notion de séquence partiellement ordonnée minimale (définition 18).

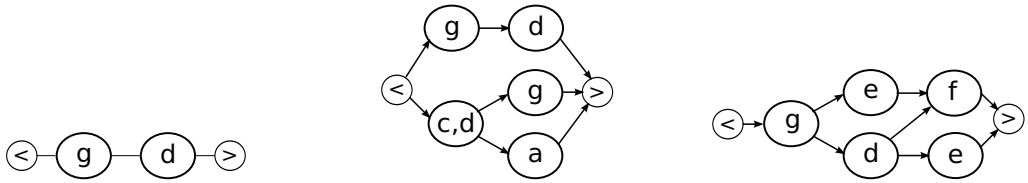
Définition 18 ***(Séquence partiellement ordonnée minimale)***
Soit $G = (\mathcal{V}, \mathcal{A}, \Sigma_\mathcal{V}, l_\mathcal{V})$ une ***séquence partiellement ordonnée****, G est* ***une séquence partiellement ordonnée minimale*** *s'il n'existe pas de séquence partiellement ordonnée $G' = (\mathcal{V}', \mathcal{A}', \Sigma'_\mathcal{V}, l'_\mathcal{V})$ tel que $G \equiv_g G'$ avec $|\mathcal{V}'| < |\mathcal{V}|$ ou $|\mathcal{A}'| < |\mathcal{A}|$.*

Ces motifs nous fournissent ainsi une information détaillée concernant l'ordre des éléments dans les séquences. Cette information supplémentaire entraîne une explosion du nombre de motifs partiellement ordonnés par rapport aux nombres de motifs séquentiels dans une même base de données. Dans un contexte applicatif réel, le nombre trop volumineux de motifs partiellement ordonnés est une contrainte. Cependant, comme pour les motifs séquentiels, il est possible de définir la notion de motif partiellement ordonné clos qui réduit considérablement la taille des résultats (définition 19).

Définition 19 ***(Motif partiellement ordonné clos)*** *Un* ***motif partiellement ordonné*** M_G *est* ***clos*** *s'il n'existe pas de* ***motif partiellement ordonné*** M'_G *tel que* $M_G \preceq_g M'_G$ *avec* $M_G \not\equiv_g M'_G$ *et* $Support(M_G) = Support(M'_G)$.

Reprenons les motifs présentés dans la figure 2.3 qui sont supportés par les séquences S_1 et S_2. Nous observons que les motifs des figures 2.3b, 2.3c, 2.3d, 2.3e et 2.3f sont inclus dans le motif M_a de la figure 2.3a. De plus, si nous reprenons les séquences S_1 et S_2, nous observons qu'il n'existe pas d'autre motif M_G supporté par ces séquences tel que $M_a \preceq_g M_G$. M_a est donc un motif partiellement ordonné clos sur S_1 et S_2. Les motifs partiellement ordonnés clos possèdent une propriété intéressante : il est possible de n'extraire qu'un seul motif partiellement ordonné clos pour un sous-ensemble de séquences de la base de données, où chaque chemin du motif partiellement ordonné clos est un motif séquentiel clos [CG05]. L'ensemble des motifs séquentiels clos supportés par un même ensemble de séquences est ainsi représenté par un seul motif partiellement ordonné clos. Par exemple, les motifs séquentiels clos $\langle (g)(d) \rangle$, $\langle (c,d)(g) \rangle$ et $\langle (c,d)(a) \rangle$ supportés par S_1 et S_2 sont synthétisés par le motif partiellement ordonné M_a de la figure 2.3a. Les figures 2.4a, 2.4b et 2.4c nous montrent les trois motifs partiellement ordonnés clos qui couvrent la base de données du tableau 2.4 avec un support minimum $\theta = 2$. La légende donne l'ensemble des séquences de la base de données qui supportent le motif de la figure.

Alors que le nombre de motifs partiellement ordonnés explose en comparaison du nombre de motifs séquentiels, le nombre de motifs partiellement ordonnés clos peut être au contraire inférieur au nombre de motifs séquentiels clos (dans notre exemple, trois motifs partiellement ordonnés clos contre six motifs séquentiels clos).



(a) $\{S_1, S_2, S_3\}$ (b) $\{S_1, S_2\}$ (c) $\{S_2, S_3\}$

FIGURE 2.4 – Motifs partiellement ordonnés clos extraits à partir de la base de données du tableau 2.4 avec un support minimum $\theta = 2$

2.4 Motifs d'intérêt

Les définitions précédentes introduisent la notion de fouille de motifs dans des bases de données temporelles. Nous nous intéressons maintenant à une problématique indépendante du type de la base de données (temporelle ou autre), mais liée à la notion de classe ou de partition d'une base de données. Par exemple, c'est typiquement le cas des jeux de données qui mettent en jeu des problématiques de classification. Comme nous l'avons vu précédemment, les motifs sont décrits par la notion de support. Si l'on considère une partition en classes de la base de données, un motif aura un support différent dans chacune d'entre elles. Ainsi, dans le cadre d'une exploration à l'aide de motifs, chaque classe d'une base de données partitionnée peut être décrite par des motifs qui sont plus fréquents dans une classe que dans les autres. Bien qu'il existe de nombreuses mesures d'intérêt qui ont été proposées dans le cadre des motifs [GH06], nous nous sommes particulièrement intéressés à l'extraction de motifs plus fréquents dans une classe que dans les autres.

Pour extraire de tels motifs, [DL99] a introduit la notion de motifs discriminants, également appelé motifs émergents. Un motif discriminant est caractérisé par une valeur statistique appelée *growth rate*, qui peut être traduite en français par taux d'émergence, qui est calculée par le rapport entre les fréquences dans les différentes classes d'une base de données. Cette mesure peut s'appliquer à tous les types de motifs car elle se base sur leur fréquence (dans notre cas les motifs séquentiels et les motifs partiellement ordonnés). La définition 20 définit cette mesure.

Définition 20 ***(Growth rate)***
Soient un motif M et deux classes $\mathcal{C}$ et $\mathcal{C}'$, le growth rate de M dans $\mathcal{C}$ par rapport à $\mathcal{C}'$, noté $GR(M, \mathcal{C}, \mathcal{C}')$, est le rapport de fréquences défini par :

$$\begin{cases} \text{- } 0, \textit{ si } Frequence_{\mathcal{C}}(M) = 0 \textit{ et } Frequence_{\mathcal{C}'}(M) = 0 \\ \text{- } \infty, \textit{ si } Frequence_{\mathcal{C}}(M) \neq 0 \textit{ et } Frequence_{\mathcal{C}'}(M) = 0 \\ \text{- } \frac{Frequence_{\mathcal{C}}(M)}{Frequence_{\mathcal{C}'}(M)}, \textit{ sinon} \end{cases} \tag{2.4}$$

Pour illustrer cette approche, prenons deux classes d'une base de données d'objets, avec $\mathcal{C} = \{O_1, O_2, O_3, O_4, O_5, O_6\}$ et $\mathcal{C}' = \{O'_1, O'_2, O'_3, O'_4\}$, et un ensemble de quatre motifs $\{M_1, M_2, M_3, M_4\}$ extraits à partir de ces classes. Les tableaux 2.5a et 2.5b donnent le contexte d'extraction (cf. définition 1) de chaque classe.

	M_1	M_2	M_3	M_4
O_1	×	×	×	×
O_2	×	×		
O_3		×	×	
O_4	×		×	
O_5	×			×
O_6		×		

(a)

	M_1	M_2	M_3	M_4
O'_1	×		×	×
O'_2	×			×
O'_3				×
O'_4	×		×	×

(b)

FIGURE 2.5 – Environnements d'extraction des classes (a) $\mathcal{C}$ et (b) $\mathcal{C}'$

Pour chaque motif, analysons leur support dans les deux classes. Le motif M_3 a une fréquence de 50% dans $\mathcal{C}$ et dans $\mathcal{C}'$, il est donc porteur d'une information qui a la même distribution dans chaque classe. Au contraire, M_4 a une fréquence de 33,33% dans $\mathcal{C}$ et une fréquence de 100% dans $\mathcal{C}'$. Ce motif est donc bien plus fréquent dans $\mathcal{C}'$. Le motif M_2 est présent à 66,66% dans $\mathcal{C}$ et à 0% dans $\mathcal{C}'$. M_2 est donc très intéressant puisqu'il est seulement représenté dans la classe $\mathcal{C}$. Le *growth rate* permet de mettre efficacement en évidence cet aspect au travers du rapport de fréquence. Le tableau 2.7 renseigne, pour chaque motif, sa valeur de *growth rate* dans $\mathcal{C}$ et $\mathcal{C}'$.

Pour un motif donné, une valeur de *growth rate* strictement supérieure à 1 dans une classe signifie que ce motif est plus fréquent dans celle-ci. Plus la valeur est forte, plus le motif est fréquent. Par exemple, M_4 a un *growth rate* de 3 dans $\mathcal{C}'$ par rapport à $\mathcal{C}$, ce qui signifie qu'il est trois fois plus fréquent dans $\mathcal{C}'$ que dans $\mathcal{C}$. Les motifs les plus intéressants sont ceux qui sont de petite taille en nombre

Motif	**Fréquence dans $\mathcal{C}$**	**Fréquence dans $\mathcal{C}'$**	**Growth rate dans $\mathcal{C}$**	**Growth rate dans $\mathcal{C}'$**
M_1	66,67%	75%	0,88	**1,13**
M_2	66,67%	0%	∞	0
M_3	50%	50%	1	1
M_4	33,34%	100%	0,33	**3**

TABLE 2.7 – Valeurs de *growth rate* de M_1, M_2, M_3 et M_4

d'items et qui ont un *growth rate* infini dans une classe, car ils sont alors fortement représentatifs de cette classe tout en limitant le sur-apprentissage. Ces motifs sont appelés *Jumping Patterns*. Finalement, on peut dire qu'un motif M est plus fréquent dans une classe $\mathcal{C}$ que dans une autre classe $\mathcal{C}'$ si $GR(M, \mathcal{C}, \mathcal{C}') > 1$. Ce motif est alors dit **discriminant**.

Comme pour l'extraction de motifs où un seuil de support minimum θ est utilisé, l'extraction de motifs discriminants peut être seuillée par un seuil minimal de discrimination ρ [DL99]. Il est alors possible de limiter l'extraction des motifs aux plus discriminants. Ce seuil minimal de discrimination est une valeur comprise entre 1 et l'infini, i.e. mettre un seuil entre 0 et 1 reviendrait à extraire des motifs non-discriminants. Plus la valeur du seuil est grande, plus les motifs extraits sont discriminants.

Cette mesure est adaptée au cas de bases de données composées de seulement deux classes. Nous proposons, dans le chapitre 4, une méthode pour appliquer cette mesure dans le cas de bases de données composées de n classes.

2.5 Discussion

Comme nous l'avons vu dans ce chapitre, la notion de motif est sous-entendue par la recherche d'informations fréquentes et d'intérêt dans des bases de données. Nous nous sommes en particulier intéressés aux motifs pouvant être extraits à partir de bases de données temporelles, qui sont les motifs séquentiels et les motifs partiellement ordonnés.

Cependant, l'extraction de tels motifs reste une tâche difficile. Il faut considérer l'explosion combinatoire qui peut s'avérer être un véritable problème lorsque les données deviennent complexes ou bien trop volumineuses. En effet, passer du problème de l'extraction d'itemsets fréquents à l'extraction de motifs séquentiels

entraîne une explosion de l'espace de recherche. Il en est de même pour l'extraction de motifs partiellement ordonnés qui, basée sur l'extraction de motifs séquentiels, implique de nouvelles approches pour être efficace sur un espace de recherche bien plus important.

De plus, dans le cadre de bases de données composées de classes, la fouille de motifs ne consiste pas seulement à extraire l'information fréquente d'une base de données, mais aussi à identifier les motifs qui sont plus fréquents dans une classe que dans les autres.

Les méthodes existantes pour la fouille de motifs partiellement ordonnés ont certaines limites qui ont motivé nos approches. Ainsi, cette thèse se focalise sur la proposition de méthodes efficaces pour extraire et filtrer des structures partiellement ordonnées, sous la forme de motifs ou de résumés, dans le cadre de bases de données divisées en classes.

CHAPITRE 3

OrderSpan : une nouvelle approche pour l'extraction de motifs partiellement ordonnés clos

Contents

3.1 Introduction

Comme présenté dans le chapitre 2, la fouille de motifs partiellement ordonnés clos reste peu étudiée et est une tâche difficile. Les approches existantes ont certaines limites que nous avons essayé de résoudre.

Dans cette optique, ce chapitre présente notre méthode appelée *OrderSpan*. Cet algorithme a pour but l'extraction de l'ensemble complet des motifs partiellement ordonnés clos qui couvrent une base de données de séquences d'itemsets. Nous nous sommes basés sur le paradigme *Pattern-Growth* et la projection de la base de données avec les préfixes fréquents ainsi que sur une optimisation qui a fait ses preuves en fouille de motifs séquentiels clos [YHA03].

Avant de présenter l'approche, nous présentons la motivation de notre travail en étudiant les limites des approches existantes. Ensuite, nous présentons notre méthode organisée en deux étapes principales (sections 3.3.1 et section 3.3.2). Celles-ci utilisent les propriétés sur les préfixes et les suffixes des séquences. Dans la section 3.3.3, nous montrons qu'il est possible d'adapter les optimisations déjà proposées dans le cadre de la fouille de motifs séquentiels clos. Ensuite, nous donnons la complexité de l'approche dans la section 3.4. Pour terminer, dans la section 3.5, nous montrons les expérimentations effectuées sur des jeux de données réels, couramment utilisés par la communauté. Nous comparons les performances de la version de base et de la version optimisée ainsi que les performances vis-à-vis de l'existant.

3.2 Motivations

L'utilisation des motifs partiellement ordonnés est pertinente dans le cadre des bases de données de séquences. Cependant, comme présenté précédemment, pour fouiller de telles bases, les chercheurs se sont principalement focalisés sur la fouille de motifs séquentiels. Or, les motifs partiellement ordonnés permettent de fournir à l'utilisateur une information différente et plus complète. Cependant, extraire de tels motifs nécessite la définition de méthodes plus complexes. Ainsi, pour fouiller de tels motifs, des approches [CG05, PWL$^+$06] ont été proposées. Néanmoins, à notre connaissance, les méthodes existantes sont limitées à des bases de données de séquences particulières ou ne permettent pas d'extraire l'ensemble complet des motifs partiellement ordonnés clos comme nous allons l'expliquer dans la section suivante.

3.2.1 Limites des approches existantes

La méthode proposée dans [PWL+06] est limitée car elle permet seulement d'extraire l'ensemble des motifs partiellement ordonnés clos sur un cas particulier de bases de données de séquences : (1) ces séquences ne sont pas composées d'itemsets mais seulement d'items et (2) un même item ne peut pas se répéter plusieurs fois dans une même séquence. Cela réduit considérablement les champs d'applications de cette méthode puisque de nos jours, les bases de données temporelles permettent de stocker plusieurs variables pour une même date (itemset), e.g. une station météorologique mesure plusieurs indicateurs à une même date. De plus, une variable peut se répéter plusieurs fois au cours d'une même séquence, e.g. la même valeur pour un indicateur peut être mesurée à des dates différentes.

Dans [CG05], l'approche permet d'extraire des motifs partiellement ordonnés sur n'importe quel type de base de données de séquences mais elle n'en extrait pas la totalité. Ceci est du au fait que les motifs partiellement ordonnés clos sont générés à partir d'un ensemble de motifs séquentiels clos, extraits au préalable avec l'algorithme BIDE [WH04], qui sont supportés par les mêmes ensembles de séquences. Ci-dessous, nous démontrons cette limite en nous basant sur un contre-exemple.

Soit un ensemble $\mathcal{M}_S$ de motifs séquentiels clos supportés par un ensemble $\mathcal{S}$ de séquences de la base de données. Le motif partiellement ordonné clos généré à partir de cet ensemble de motifs séquentiels clos est nécessairement supporté par $\mathcal{S}$ [CG05]. Dans cette approche, pour chaque ensemble $\mathcal{S}$ de séquences de la base de données qui supporte un motif partiellement ordonné clos calculé à partir d'un ensemble $\mathcal{M}_S$ de motifs séquentiels clos, il existe au moins un motif séquentiel $M_S \in \mathcal{M}_S$ qui est clos sur $\mathcal{S}$. Or, il existe des motifs partiellement ordonnés clos sur un ensemble $\mathcal{S}$ de séquences tel qu'il n'existe pas de motif séquentiel $M_S \in \mathcal{M}_\mathcal{S}$ qui soit clos sur $\mathcal{S}$. Pour illustrer cela, considérons la base de données de séquences du tableau 3.1 et l'ensemble des motifs séquentiels clos extraits à partir de celle-ci dans le tableau 3.2, avec un support minimum de 2.

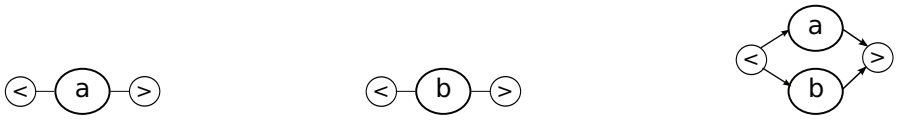
ID	**Séquence**
S'_1	$\langle(a)\rangle$
S'_2	$\langle(a)(b)\rangle$
S'_3	$\langle(b)(a)\rangle$
S'_4	$\langle(b)\rangle$

TABLE 3.1 – Exemple d'une base minimale de séquences

Séquences qui supportent le motif	**Motif séquentiel clos**
$\{S'_1, S'_2, S'_3\}$	$\langle(a)\rangle$
$\{S'_2, S'_3, S'_4\}$	$\langle(b)\rangle$

TABLE 3.2 – Motifs séquentiels clos extraits de la base du tableau 3.1 avec un support minimum $\theta = 2$

De la même manière, prenons les trois motifs partiellement ordonnés clos extraits à partir de cette même base et donnés par la figure 3.1. L'approche proposée dans [CG05] n'est pas capable d'extraire le motif partiellement ordonné M'_{G_3} puisque dans le tableau 3.2, nous voyons qu'il n'existe pas de motif séquentiel clos sur l'ensemble de séquences $\{S'_2, S'_3\}$.



(a) M'_{G_1} sur $\{S'_1, S'_2, S'_3\}$ (b) M'_{G_2} sur $\{S'_2, S'_3, S'_4\}$ (c) M'_{G_3} sur $\{S'_2, S'_3\}$

FIGURE 3.1 – Motifs partiellement ordonnés clos extraits de la base du tableau 3.1 avec un support minimum $\theta = 2$

3.2.2 Contribution

La méthode présentée dans ce chapitre a été conçue de manière à corriger les limites des approches présentées précédemment (voir sections 2.3.3 et 3.2.1) :

— Tous les types de séquences peuvent être fouillés, i.e. les séquences composées d'itemsets dans lesquelles un même item peut se répéter plusieurs fois.

— Selon un seuil de fréquence/support minimum, l'ensemble complet des motifs partiellement ordonnés est extrait.

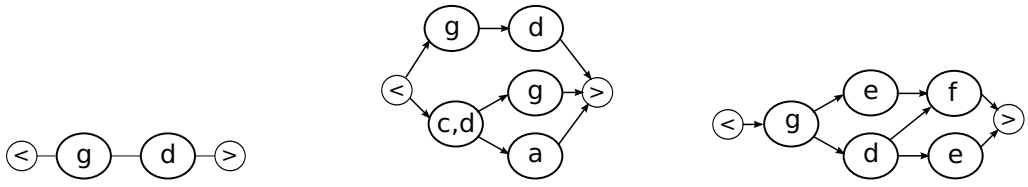
L'approche proposée utilise le paradigme *Pattern-Growth* introduit dans *PrefixSpan* [PHMA+04] et utilisé dans de nombreux autres algorithmes de fouille de motifs. Nous avons également adapté l'optimisation proposée dans *CloSpan* [YHA03] à l'extraction de motifs partiellement ordonnés clos.

3.3 Présentation de l'algorithme

Dans le chapitre sur l'état de l'art, nous avons mis en évidence le lien entre les motifs séquentiels et les motifs partiellement ordonnés. Il est ainsi possible de transformer la problématique de la fouille de motifs séquentiels en une problématique de fouille de motifs partiellement ordonnés. L'algorithme proposé dans ce chapitre, *OrderSpan*, est basé sur les méthodes d'extraction qui ont fait leurs preuves en fouille de motifs séquentiels. Avant de présenter l'algorithme et pour aider dans la compréhension de la méthode proposée, nous rappelons dans le tableau 3.3 et les figures 3.2 et 3.3 la base de données exemple, les motifs partiellement ordonnés clos ainsi que l'arbre de recherche utilisés dans le chapitre 2.

ID	**Séquence**
S_1	$\langle(c,d)(a)(g)(d)\rangle$
S_2	$\langle(g)(c,d,e)(f)(a,e,g)\rangle$
S_3	$\langle(g)(d)(e)(f)\rangle$

TABLE 3.3 – Exemple de base de données de séquences



(a) M_{G_1} sur $\{S_1, S_2, S_3\}$ (b) M_{G_2} sur $\{S_1, S_2\}$ (c) M_{G_3} sur $\{S_2, S_3\}$

FIGURE 3.2 – Motifs partiellement ordonnés clos extraits à partir de la base de données du tableau 3.3 avec un support minimum $\theta = 2$

Dans le chapitre 2, nous avons présenté le fonctionnement du paradigme *Pattern-Growth* ainsi que la projection de la base de données avec les préfixes fréquents. *OrderSpan* reprend ce même principe qui est étendu aux motifs partiellement ordonnés clos. Cette adaptation entraîne la nécessité de mettre en place une structure de données interne à l'algorithme qui étend la structure de données des séquences partiellement ordonnées. La raison est la suivante : pour être efficace durant la génération des motifs partiellement ordonnés, les opérations de I-Extension et de S-Extension (cf. chapitre 2) sont maintenant considérées comme deux types d'arcs

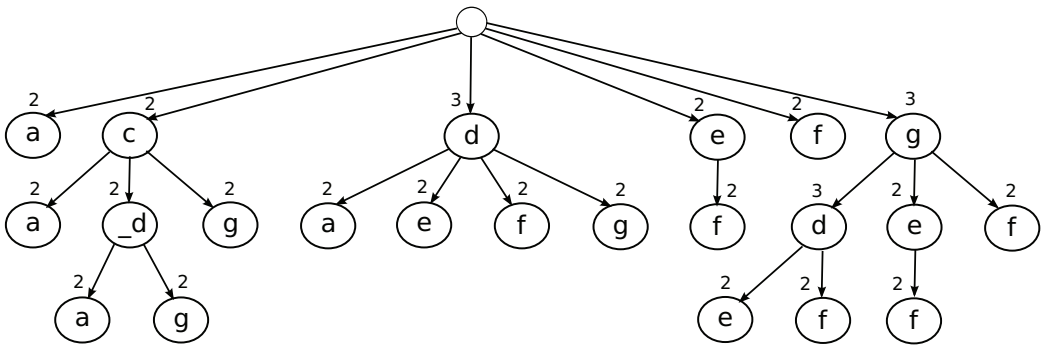


FIGURE 3.3 – Arbre de recherche des motifs séquentiels généré à partir de la base de données de séquences du tableau 3.3 avec un support minimum de $\theta = 2$

différents. Les sommets ne sont alors plus composés d'itemsets mais d'items. La définition 21 définit cette nouvelle structure de données.

Définition 21 ***(Séquence partiellement ordonnée étendue)***
Une ***séquence partiellement ordonnée étendue*** *est un ensemble d'****items*** *qui peut être représenté par un* ***graphe acyclique orienté*** $G^{exp} = (\mathcal{V}, \mathcal{A}, \Sigma_{\mathcal{V}}, \Sigma_{\mathcal{A}}, l_{\mathcal{V}}, l_{\mathcal{A}})$ *munis d'un ordre partiel* $(\mathcal{V}, \leq)$*, où :*

- $\mathcal{V}$ *est l'ensemble des* ***sommets*** *et* $\mathcal{A}$ *est l'ensemble des* ***arcs*** *où* $\mathcal{A} = \{(u, v) \in <$ *, avec* $u, v \in \mathcal{V}\}$
- $\Sigma_{\mathcal{V}}$ *est un alphabet composé d'****items*** *représentant les valeurs possibles des* ***sommets*** *et* $\Sigma_{\mathcal{A}} = \{I\text{-}Extension, S\text{-}Extension\}$
- $l_{\mathcal{V}} : \mathcal{V} \to \Sigma_{\mathcal{V}}$ *et* $l_{\mathcal{A}} : \mathcal{A} \to \Sigma_{\mathcal{A}}$ *sont deux fonctions donnant respectivement les étiquettes des* ***sommets*** *et des* ***arcs***
- *et soit deux sommets* $u, v \in \mathcal{V}$, $l_{\mathcal{A}}(u, v) =$ "$I\text{-}Extension$" *signifie que les deux sommets appartiennent au même itemset* $\langle(uv)\rangle$ *et* $l_{\mathcal{A}}(u, v) =$ "$S\text{-}Extension$" *signifie que les deux sommets appartiennent à une séquence* $\langle(u)(v)\rangle$

Nous illustrons cette définition en prenant comme exemple le motif partiellement ordonné M_{G_2}, présenté par la figure 3.2b. La figure 3.4 donne la version étendue de ce motif.

Cette transformation divise chaque itemset en autant de sommets qu'il y a d'items qui le composent. Cela induit un nouveau type d'arc pour symboliser l'appartenance de plusieurs items à un même itemset (I-Extension). Dans la figure, ces arcs sont en pointillés pour les distinguer des opérations de S-Extension. Dans un itemset, les items ne sont normalement pas ordonnés entre eux. Cependant, le paradigme *Pattern-Growth* nécessite que les items dans les itemsets soient ordonnés

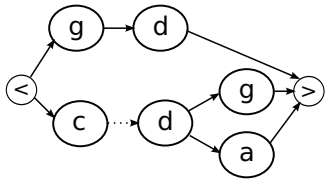


FIGURE 3.4 – Version étendue du motif partiellement ordonné M_{G_2}

par leur valeur lexicographique. Par exemple, dans la figure 3.4, l'itemset transformé est représenté par (c, d) ou (d, c). Comme $c < d$ selon l'ordre lexicographique, alors l'arc (c, d) avec $l_{\mathcal{A}}(c, d) = I\text{-}Extension$ est ajouté à la version étendue de M_{G_2}. Pour rappel, utiliser l'ordre lexicographique items dans les itemsets garantit de générer les motifs partiellement ordonnés dans un ordre donné et empêche la génération d'un même motif plusieurs fois.

Un motif partiellement ordonné clos sur un ensemble de séquences $\mathcal{S}$ contient l'ensemble des motifs séquentiels qui couvrent $\mathcal{S}$ [CG05]. Nous nous basons sur cet aspect dans la première étape de l'algorithme. Tout d'abord, regardons de plus près l'espace de recherche donné par la figure 3.3. Nous observons que chaque motif partiellement ordonné clos est représenté par un sous-arbre de l'espace de recherche des motifs séquentiels. Par exemple, pour le motif M_{G_2}, le sous-arbre qui représente ce motif dans l'espace de recherche est donné par la figure 3.5.

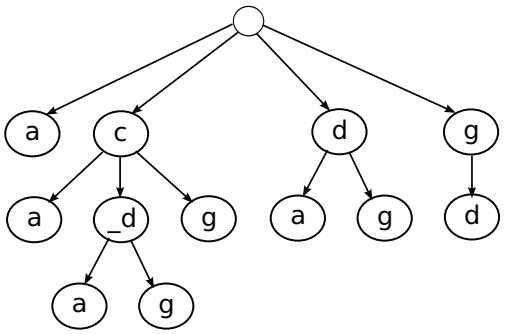


FIGURE 3.5 – Sous-arbre de l'espace de recherche couvrant les séquences S_1 et S_2

Ce sous-arbre contient l'ensemble complet des motifs séquentiels qui sont supportés par les séquences S_1 et S_2 de la base de données du tableau 3.3. Pour chaque motif séquentiel M_S dans l'arbre, il existe au moins un chemin S dans M_{G_2} tel que $M_S \preceq_s S$. Pour finir, pour chaque sous-ensemble de séquences d'une base de données, il existe un sous-arbre de l'espace de recherche qui représente tous les motifs séquentiels qui sont supportés par ce sous-ensemble de séquences.

3.3.1 ForwardTreeMining : extraction de sous arbres préfixés

La première étape de l'algorithme est fondée sur cette notion de sous-arbre dans l'espace de recherche qui couvre un sous-ensemble de séquences. Ainsi, l'objectif est d'extraire en premier lieu, sous la forme de motifs partiellement ordonnés, tous les sous-arbres de l'espace de recherche qui correspondent aux différents sous-ensembles de séquences de la base de données. Cette opération est appelée *ForwardTreeMining*. Les motifs partiellement ordonnés extraits dans cette étape n'ont pas encore leur forme finale, leur contenu est équivalent à celui du sous-arbre de l'espace de recherche correspondant. Reprenons l'exemple des séquences S_1 et S_2, l'opération *ForwardTreeMining* extrait le motif partiellement ordonné clos de la figure 3.6. Ce motif est équivalent au sous-arbre de recherche donné par la figure 3.5.

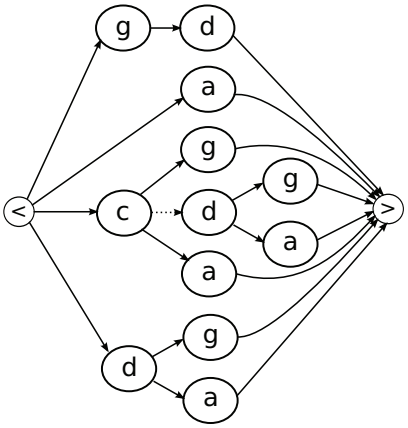


FIGURE 3.6 – Le motif partiellement ordonné équivalent au sous-arbre de l'espace de recherche sur S_1 et S_2 après l'opération *ForwardTreeMining*

Sur la base des préfixes fréquents [PHMA+04] pour l'exploration et la projection de la base de données, *OrderSpan* extrait l'ensemble complet des sous-arbres de l'espace de recherche. Cet algorithme est tout d'abord initialisé avec le sous-arbre, sous la forme d'un motif partiellement ordonné, qui couvre toutes les séquences de la base de données. Ce premier motif peut être vide si aucune séquence n'est commune à toute la base. Il est ensuite utilisé pour récursivement extraire tous les motifs partiellement ordonnés qui couvrent des ensembles de séquences $\mathcal{S}$ dont la taille $|\mathcal{S}|$ est supérieure ou égale à un support minimum θ. À partir d'un motif partiellement ordonné sur un ensemble de séquences $\mathcal{S}$, l'opération permet alors de découvrir de nouveaux motifs partiellement ordonnés clos sur des ensembles de séquences $\mathcal{S}'$ tel que $\mathcal{S}' \subset \mathcal{S}$.

Durant la construction d'un motif sur un ensemble de séquences $\mathcal{S}$, $\mathcal{S}$ est tout d'abord projeté avec les préfixes fréquents de taille 1 (racine de l'espace de recherche). De nouveaux sommets sont alors au fur et à mesure ajoutés au motif et l'ensemble $\mathcal{S}$ récursivement projeté. Pour un sommet, lorsqu'une extension (I-Extension ou S-Extension) est possible à partir d'un item fréquent, deux cas sont possibles :

— l'item fréquent a un support dans la base projetée égal à $|\mathcal{S}|$ (l'item est présent dans chaque séquence projetée de $\mathcal{S}$). Dans ce cas le motif est étendu (ajout d'un sommet au motif).
— l'item fréquent a un support dans la base projetée inférieur à la taille de l'ensemble $|\mathcal{S}|$. Il existe alors un motif partiellement ordonné supporté par un ensemble $\mathcal{S}'$ tel que $\mathcal{S}' \subset \mathcal{S}$. Dans ce cas, l'opération *ForwardTreeMining* est récursivement appelée sur l'ensemble $\mathcal{S}'$ pour générer le motif partiellement ordonnés clos sur cet ensemble.

Cependant, deux motifs partiellement ordonnés clos qui couvrent respectivement des ensembles de séquences $\mathcal{S}$ et $\mathcal{S}'$ peuvent être parents (dans le processus d'exploration) d'un même motif qui couvre un ensemble de séquences $\mathcal{S}''$, avec $\mathcal{S}'' \subset \mathcal{S}$ et $\mathcal{S}'' \subset \mathcal{S}'$. Alors, lorsqu'un nouveau motif peut potentiellement être extrait sur un ensemble de séquences $\mathcal{S}$, l'algorithme s'assure qu'un motif n'a pas déjà été extrait sur $\mathcal{S}$. Cela empêche la génération d'un même motif plusieurs fois.

Cette vérification peut être coûteuse. Pour être efficace, nous utilisons une structure de données qui est un ensemble d'ensembles de séquences pour lesquelles un motif partiellement ordonné clos a déjà été extrait. Cette structure permet, à chaque fois qu'un nouveau motif est découvert, de vérifier si ce motif n'a pas déjà été extrait en vérifiant si l'ensemble des séquences qui le supportent est inclus dans cette structure. Cet ensemble d'ensembles de séquences est muni d'un ordre sur les ensembles de séquences, eux-mêmes ordonnés. Soit $\mathcal{M}_G$ l'ensemble des motifs précédemment extraits, vérifier si un nouveau motif à déjà été extrait se fait alors de manière dichotomique en $\Theta(\log(|\mathcal{M}_G|))$.

L'algorithme 1 décrit ce processus.

Nous détaillons chaque étape de l'algorithme. **Lignes [1-4]** : un graphe acyclique orienté vide est créé pour le motif partiellement ordonné clos. Chaque sommet du graphe contient : (1) une étiquette concernant l'information extraite (un item) et (2) la base de données projetée sur l'ensemble $\mathcal{S}$ en fonction du préfixe fréquent, i.e

Algorithm 1: ForwardTreeMining

Données: $\mathcal{S}$ un ensemble de séquences, θ un support minimum, $ListSet$ l'ensemble des ensembles de séquences sur lesquels un motif a été extrait

Résultat: M_G un motif partiellement ordonné

```
1  M_G = graphe vide ;
2  M_G.begin.database = base projetée sur S;
3  NodeQueue = liste vide de sommets;
4  ajouter le sommet M_G.begin à NodeQueue;
5  tant que NodeQueue n'est pas vide faire
6      Node = dépiler un sommet de NodeQueue;
7      FreqItems = récupérer les items fréquents dans la base projetée
       Node.database selon θ;
8      pour chaque Item dans FreqItems tel que Support(Item) = |S| faire
9          Node' = nouveau sommet;
10         Node'.database = base projetée de Node.database selon Item;
11         étendre le sommet N avec N' ;
12         ajouter N' à NodeQueue;
13     fin
14     pour chaque Item dans FreqItems tel que Support(Item) < |S| faire
15         si ListSet ne contient pas l'ensemble des séquences qui supportent
           Item alors
16             S' = ensemble des séquences qui supportent Item;
17             ajouter S' à ListSet;
18             ForwardTreeMining(S', θ, ListSet) ;
19
20     fin
21 fin
22 Retourner M_G;
```

du motif séquentiel qui correspond au chemin depuis le sommet racine. Une liste *NodeQueue* est initialisée avec le sommet de début du motif partiellement ordonné (racine de l'espace de recherche). **Lignes [5-21]** : *NodeQueue* est vide lorsqu'il n'y a plus de sommet à étendre, i.e. le motif partiellement ordonné est extrait. **Lignes [8-13]** : le motif est étendu quand un item fréquent est supporté par toutes les séquences de $\mathcal{S}$. **Lignes [14-20]** : lorsqu'un item fréquent a un support inférieur à la taille de l'ensemble $|\mathcal{S}|$, si aucun motif n'a été extrait sur l'ensemble $\mathcal{S}'$, un nouveau est alors généré en appelant récursivement l'opération *ForwardTreeMining* sur $\mathcal{S}'$. **Lignes [22]** : le motif est retourné.

3.3.2 MergingSuffixTree : fusion des sous-arbres

Dans l'étape précédente, nous avons utilisé la propriété des préfixes fréquents pour extraire l'ensemble complet des sous-arbres de l'espace de recherche fréquents, sous la forme de motifs partiellement ordonnés. Cette seconde étape a pour but, à partir de chaque motif, de supprimer les chemins redondants et de fusionner les sommets qui font référence aux mêmes suffixes. Dans les motifs extraits avec l'opération *ForwardTreeMining*, nous observons de nombreuses redondances. Par exemple, considérons le motif de la figure 3.6. Celui-ci contient l'ensemble des motifs séquentiels supportés par les séquences S_1 et S_2. Nous observons que certains de ces motifs séquentiels ne sont pas clos, ce qui génère de l'information redondante. Par exemple, la séquence $\langle(c)(g)\rangle$ est incluse dans la séquence $\langle(c,d)(g)\rangle$. De même la séquence $\langle(a)\rangle$ est incluse dans la séquence $\langle(c,d)(a)\rangle$. Ainsi, ce motif partiellement ordonné clos a encore la forme du sous-arbre de recherche associé composé de chemins et de sommets redondants. L'opération présentée maintenant a deux avantages : (1) elle supprime la redondance due aux motifs séquentiels non-clos et (2) elle génère en même temps le motif partiellement ordonné clos en fusionnant les sommets.

À partir du sommet de fin des motifs partiellement ordonnés obtenus suite à la première opération, et de manière récursive, nous fusionnons les sommets qui possèdent une même étiquette en utilisant les propriétés sur les suffixes des séquences. En effet, fusionner ces sommets conserve l'ordre des items dans chaque chemin (séquence) du motif partiellement ordonné. Par exemple, les séquences $\langle(c)(a)\rangle$ et $\langle(c,d)(a)\rangle$ ont toutes les deux pour suffixe la séquence $\langle(a)\rangle$. Dans le motif, les sommets représentant ce suffixe sont alors fusionnés en un seul sommet étiqueté par a. Cette opération récursive supprime automatiquement la redondance due aux

motifs séquentiels non-clos, qui sont d'une certaine manière absorbés par les clos. Cette opération est appelée *MergingSuffixTree*. La figure 3.7 fournit un exemple d'application de cette opération sur le motif partiellement ordonné de la figure 3.6.

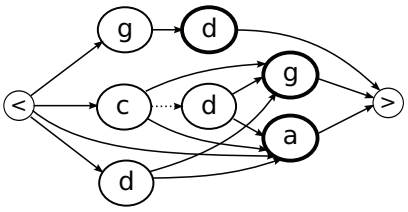


FIGURE 3.7 – Opération de fusion sur le motif partiellement ordonné couvrant les séquences S_1 et S_2 (figure 3.6)

Dans cette figure, l'opération *MergingSuffixTree* a été ici seulement appliquée sur les sommets parents du sommet de fin. Les sommets représentant le suffixe $\langle(a)\rangle$ et étiquetés par a sont fusionnés tout en conservant les arcs de leurs sommets parents. Cette opération est également appliquée aux sommets étiquetés par d et g, qui correspondent respectivement aux suffixes $\langle(d)\rangle$ et $\langle(g)\rangle$. Pour le suffixe $\langle(d)\rangle$, le cas est trivial puisqu'il n'existe qu'un seul sommet d parent du sommet de fin. Une fois qu'un ensemble de sommets faisant référence à un même suffixe a été fusionné en un sommet v, cette opération est appliquée récursivement aux sommets parents de v. Ainsi, appliqué aux sommets parents de a, tous les sommets ayant pour label c ou d sont fusionnés chacun à leur tour car ils correspondent respectivement aux suffixes $\langle(c)(a)\rangle$ et $\langle(d)(a)\rangle$. Dans un motif partiellement ordonné, un sommet v peut faire référence à de multiples suffixes représentés par les arcs sortants de v. Par exemple, le sommet d peut faire référence aux suffixes $\langle(d)(a)\rangle$ et $\langle(d)(g)\rangle$. Pour garder l'ordre entre les items, il est important de noter que deux sommets ayant la même étiquette sont fusionnés seulement s'ils font exactement référence au même ensemble de suffixes dans le motif.

Maintenant, considérons le suffixe $\langle(c)(a)\rangle$. Conserver les arcs depuis les sommets parents du sommet fusionné peut récursivement entraîner la génération d'une redondance due à la transitivité dans le motif partiellement ordonné. Dans la figure 3.7, l'arc partant du sommet c vers le sommet fusionné a est redondant puisque l'ordre entre ces deux sommets est déjà contenu dans le chemin représenté par la séquence $\langle(c,d)(a)\rangle$. Donc, après avoir fusionné un sommet avec l'opération *MergingSuffixTree*, nous vérifions la redondance des arcs venant des sommets parents avec la propriété 1.

Propriété 1 ***(Arc redondant)***
Soient $G = (\mathcal{V}, \mathcal{A}, \Sigma_{\mathcal{V}}, \Sigma_{\mathcal{A}}, l_{\mathcal{V}}, l_{\mathcal{A}})$ une séquence partiellement ordonnée étendue avec deux arcs $\alpha, \gamma \in \mathcal{V}$, un chemin allant de α à γ représenté par une séquence S avec I_1 le premier item de S et I_n le dernier item de S, et trois arcs $(\alpha, \gamma), (\alpha, I_1)$, $(I_n, \gamma) \in \mathcal{A}$ tel que $l_{\mathcal{A}}(\alpha, \gamma) = k$, $l_{\mathcal{A}}(\alpha, I_1) = l$ et $l_{\mathcal{A}}(I_n, \gamma) = m$. L'arc (α, γ) est redondant si $\alpha \diamond_k \gamma \preceq_s \alpha \diamond_l S \diamond_m \gamma$.

La figure 3.8 illustre cette propriété.

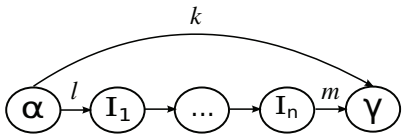


FIGURE 3.8 – Transitivité

Considérons un sommet fusionné v. Élaguer la redondance due à la transitivité implique de vérifier la propriété 1 pour chaque paire de sommets parents de v. Nous détaillons ce processus dans l'algorithme *TransitivityPruning* (algorithme 2) qui est appelé sur chaque sommet v venant d'être fusionné.

Algorithm 2: TransitivityPruning

Entrées: v un sommet fusionné, M_G un motif partiellement ordonné

```
1 EdgeToRemove = ensemble de sommets vide;
2 pour chaque Edge (arc) entrant de v faire
3 |   si Edge est redondant selon la propriété 1 alors
4 |   |   ajouter Edge à EdgeToRemove;
5 |
6 fin
7 pour chaque Edge dans EdgeToRemove faire
8 |   supprimer Edge de M_G;
9 fin
```

Lignes [1-6] : Pour chaque arc entrant d'un sommet fusionné v, nous vérifions si cet arc est redondant en utilisant la propriété 1. **Lignes [7-9]** : Chaque arc redondant est supprimé du motif partiellement ordonné clos.

En appliquant l'opération *MergingSuffixTree* sur un motif partiellement ordonné nous supprimons tous les chemins redondants, i.e. toutes les séquences non-closes, et nous fusionnons les sommets qui représentent un même suffixe.

3.3.3 Optimisation de l'espace de recherche

Les deux opérations présentées dans les sections 3.3.1 et 3.3.2 sont suffisantes pour extraire l'ensemble des motifs partiellement ordonnés clos à partir d'une base de données de séquences. Pour chaque motif, le coût des opérations *ForwardTreeMining* et *MergingSuffixTree* est proportionnel à la taille du sous-arbre correspondant dans l'espace de recherche. Plus un sous-arbre est de taille importante, plus la génération du motif partiellement ordonné clos correspondant est coûteuse.

Soit $\mathcal{S}$ un sous-ensemble de séquences d'une base de données. Étant donné que le sous-arbre de l'espace de recherche sur $\mathcal{S}$ contient l'ensemble des motifs séquentiels supportés par les séquences de $\mathcal{S}$, celui-ci peut dans certains cas être inutilement volumineux et redondant. Le pire des cas concerne le calcul d'un arbre de recherche sur une seule séquence (possible avec support minimum de θ=1). Dans un tel cas, le motif partiellement ordonné clos est équivalent à la séquence elle même, alors que l'arbre de recherche utilisé pour le générer contient l'ensemble des sous-séquences sur $\mathcal{S}$. Considérons la séquence $S_3 = \langle(g)(d)(e)(f)\rangle$ du tableau 3.3, l'opération *ForwardTreeMining* appliquée à cette seule séquence donne le motif partiellement ordonné de la figure 3.9.

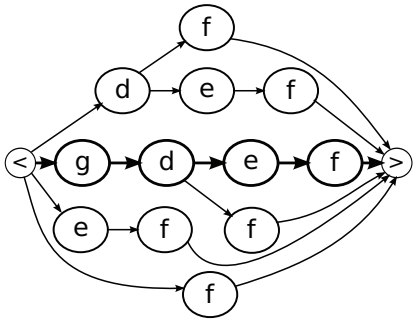


FIGURE 3.9 – Le motif partiellement ordonné extrait à partir de la séquence S_3 après l'opération *ForwardTreeMining*

Appliquer sur ce motif l'opération *MergingSuffixTree* génère le motif partiellement ordonné clos de la figure 3.10.

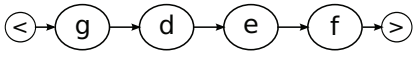


FIGURE 3.10 – Motif partiellement ordonné clos extrait à partir de la séquence S_3 après l'opération *MergingSuffixTree*

L'arbre de recherche est de taille importante en comparaison du motif partielle-

ment ordonné clos final puisque dans celui-ci, nous retrouvons la séquence S_3. Dans le motif de la figure 3.9, le motif partiellement ordonné clos représente la branche la plus profonde de l'arbre de recherche. Nous voyons ici qu'il est inutile d'explorer les autres branches puisqu'elles sont redondantes.

Ce problème a déjà été étudié dans l'algorithme *CloSpan* [YHA03] dans le cadre de la fouille de motifs séquentiels clos. Les auteurs proposent une méthode visant à éviter l'exploration de certaines branches en les élaguant pendant l'exploration de l'espace de recherche. Ils utilisent la somme du nombre d'items dans les séquences projetées pour efficacement élaguer l'espace de recherche. Grâce à ce calcul, les auteurs ont défini la notion de bases projetées équivalentes. En considérant deux préfixes fréquents et les bases projetées à partir de ceux-ci, cette astuce permet de vérifier efficacement si les bases projetées sont identiques. Si tel est le cas, exactement les mêmes sous-arbres de l'espace de recherche seront générés à partir de ces deux projections.

Nous illustrons ceci avec l'exemple donné par la figure 3.11. Durant le processus de fouille, la base de données est projetée pour chaque sommet dans le but d'étendre le motif partiellement ordonné. Plusieurs de ces projections sont équivalentes. Par exemple, les bases projetées qui correspondent aux préfixes fréquents $\langle(d)\rangle$ et $\langle(c,d)\rangle$ sont identiques car $\mathcal{D}|_{\langle(d)\rangle} = \mathcal{D}|_{\langle(c,d)\rangle} = \{\langle(a)(g)(d)\rangle, \langle(\_e)(f)(a,e,g)\rangle\}$. Elles possèdent exactement le même ensemble d'items fréquents qui sont $\langle(a)\rangle$ et $\langle(g)\rangle$.

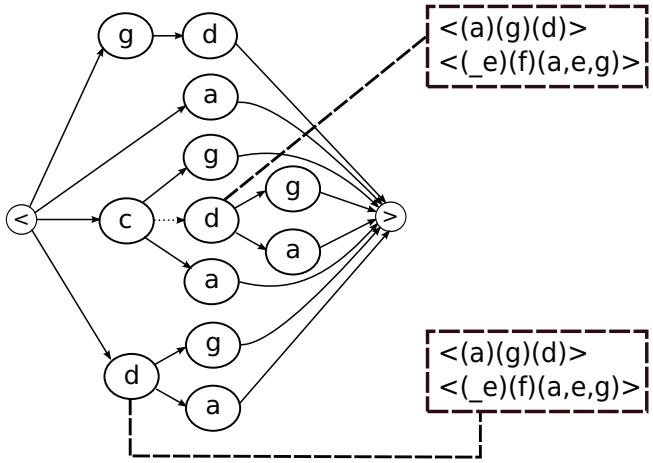


FIGURE 3.11 – Exemple de bases projetées équivalentes

Il n'est alors pas nécessaire d'explorer les deux branches qui correspondent aux préfixes fréquents $\langle(d)\rangle$ et $\langle(c,d)\rangle$.

Notre optimisation est basée sur cette observation. Sur la base des travaux dans [YHA03], nous introduisons dans cette partie l'adaptation de cette propriété au cas des motifs partiellement ordonnés clos. Comme nous allons le voir, elle permet de générer un motif partiellement ordonné pré-fusionné à l'issue de l'opération *ForwardTreeMining*. En effet, durant l'exploration de l'espace de recherche, les branches considérées comme équivalentes sont fusionnées. Intuitivement, vérifier l'équivalence de deux bases projetées est coûteux en temps car les séquences doivent être parcourues. Mais en se basant sur l'approche proposée dans *CloSpan*, l'équivalence de deux bases projetées peut être vérifiée en regardant seulement la longueur (en nombre d'items) des séquences projetées qui les composent. Pour une séquence projetée S, sa taille en items est notée $length(S)$. Nous définissons ci-dessous l'équivalence entre deux séquences projetées (définition 22).

Définition 22 ***(Équivalence de séquences projetées [YHA03])***
Soient S une séquence et α, β deux préfixes fréquents. Les séquences projetées $S|_\alpha$ et $S|_\beta$ sont équivalentes si $length(S|_\alpha) = length(S|_\beta)$.

Quand une séquence S est projetée en fonction d'un préfixe fréquent, la projection obtenue est un suffixe de S. Deux projections d'une même séquence qui ont la même longueur font nécessairement référence au même suffixe (preuve dans [YHA03]). À partir de cette définition, nous définissons l'équivalence entre deux bases de séquences projetées (définition 23).

Définition 23 ***(Équivalence de bases projetées)***
Soient $\mathcal{D}$ une base de données de séquences et α, β deux préfixes fréquents. Les bases projetées $\mathcal{D}|_\alpha$ et $\mathcal{D}|_\beta$ sont équivalentes si $\forall S \in \mathcal{D}$, $S|_\alpha$ et $S|_\beta$ sont équivalentes.

Avec cette propriété, il est maintenant possible d'efficacement vérifier si deux bases projetées sont équivalentes durant le processus d'extension du motif partiellement ordonné clos. Notre approche est différente de celle proposée dans *CloSpan*. Dans cette dernière, soit α et β deux préfixes fréquents, l'équivalence entre deux bases projetées est seulement vérifiée si $\alpha \preceq_s \beta$ ou $\beta \preceq_s \alpha$. Ceci est dû à la manière dont *CloSpan* explore l'espace de recherche. Appliquer exactement la même optimisation aux motifs partiellement ordonnés nous limiterait alors à une plus petite partie de l'espace de recherche. Dans notre approche, l'équivalence d'une base projetée dans un motif partiellement ordonné est vérifiée avec toutes les autres bases

projetées de ce même motif. Lorsqu'une extension d'un sommet v est effectuée avec un item fréquent I, un des deux cas suivants arrive :

— il n'y a pas d'autre sommet dans le motif tel que la base projetée est équivalente à une base existante, alors un nouveau sommet v' étiqueté par I est créé et un arc (correspondant au type d'extension) entre v and v' est ajouté ;

— il existe un autre sommet v'' dans le motif tel que la base projetée est équivalente à une base existante, alors un arc (correspondant au type d'extension) entre v and v'' est ajouté.

Pour illustrer cela, considérons le motif dans la figure 3.11. Avec l'optimisation, les deux sommets étiquetés par d sont représentés par un seul sommet. Cette optimisation, effectuée durant l'opération *ForwardTreeMining*, donne comme résultat le motif pré-fusionné donné par la figure 3.12.

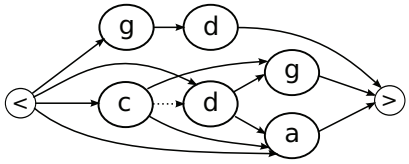


FIGURE 3.12 – Exemple de motif partiellement ordonné optimisé

Dans cet exemple, le nombre de sommets du motif à la fin de l'opération *ForwardTreeMining* est réduit de manière significative et nous observons que cette optimisation génère de la redondance due à la transitivité. Cependant, la propriété 1 est adaptée à un tel cas. Supprimer la redondance due à la transitivité générée par l'opération *MergingSuffixTree* revient à supprimer la redondance due à la transitivité générée par l'optimisation proposée. Elle est donc automatiquement supprimée lors de l'application de l'opération *MergingSuffixTree*.

La figure 3.13 synthétise le processus global d'*OrderSpan* en illustrant les interactions entre les différentes parties. Dans cette figure, les opérations spécifiques à *OrderSpan* sont mises en pointillés. Nous présentons les expérimentations dans la section 3.5

3.4 Complexité

Nous présentons maintenant une étude de la complexité des opérations *ForwardTreeMining* et *MergingSuffixTree*. Pour simplifier l'étude, ces deux opérations sont notées FTM et MST. La complexité qui est donnée ici considère seulement le pire

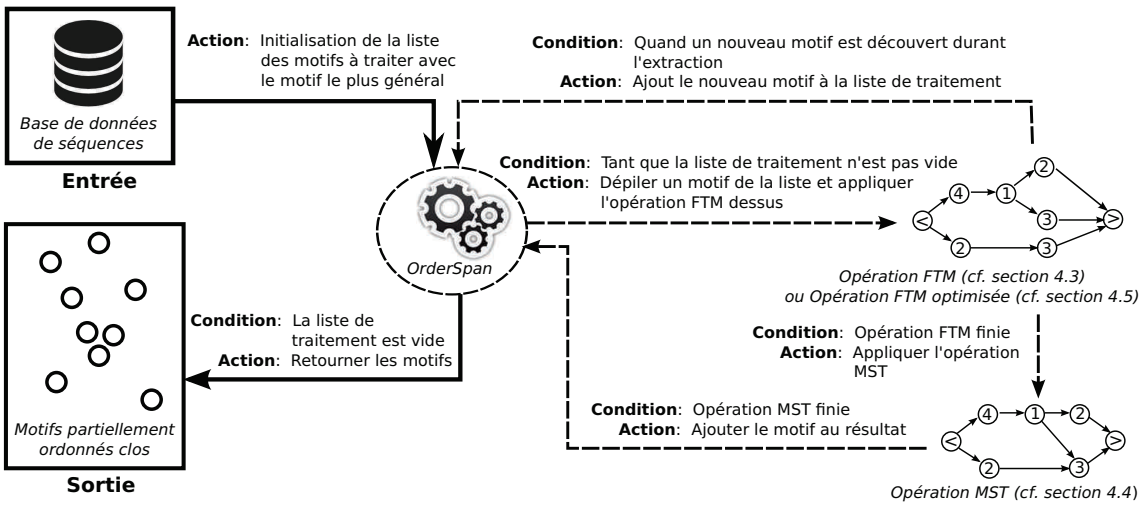


FIGURE 3.13 – Processus global

des cas. Nous ne considérons pas l'optimisation présentée dans la section 3.3.3, car le pire des cas reste le même : s'il n'y a pas de bases projetées équivalentes durant le processus, l'approche est équivalente à la version non-optimisée car aucun sommet n'est alors fusionné.

Pour résumer, soit un ensemble de séquences $\mathcal{S}$, l'opération FTM calcule l'arbre $M_{G_T}(\mathcal{V}_T, \mathcal{E}_T)$ des motifs séquentiels supportés par toutes les séquences dans $\mathcal{S}$, avec $\mathcal{V}_T$ l'ensemble des sommets et $\mathcal{E}_T$ l'ensemble des arcs. L'opération MST fusionne les sommets et arcs redondants dans M_{G_T}.

La complexité de l'opération FTM est très proche de la complexité de *PrefixSpan* puisque cela consiste à extraire tous les motifs séquentiels supportés par un ensemble de séquences. La pire complexité pour *PrefixSpan* est donnée par la formule $\Theta((2 \cdot |\mathcal{I}|)^{|S_{max}|})$ [Con06], avec $\mathcal{I}$ l'ensemble des items de la base de données de séquences et S_{max} la séquence la plus longue de la base de données. $(2 \cdot |\mathcal{I}|)^{|S_{max}|}$ correspond au nombre maximal de motifs pouvant être extraits. Dans notre cas, cette formule représente le pire des cas pour l'opération FTM sur un ensemble de séquences $\mathcal{S}$. Cela se traduit par la taille maximale, en nombre de sommets et noté $\mathcal{V}_{T_{max}}$, du motif M_{G_T} pouvant être calculé sur $\mathcal{S}$. Donc $\mathcal{V}_{T_{max}} = (2 \cdot |\mathcal{I}|)^{|S_{max}|}$. La complexité de l'opération FTM est donnée par l'équation 3.1.

$$\Theta((2 \cdot |\mathcal{I}|)^{|S_{max}|}) \tag{3.1}$$

L'opération MST appliquée sur M_{G_T} est équivalente à un algorithme de parcours en profondeur ou en largeur dans un arbre car chaque sommet de M_{G_T} est exploré

une seule fois. Un algorithme de parcours d'arbre, en profondeur ou en largeur, a une complexité linéaire en fonction du nombre de sommets, soit $\Theta(|\mathcal{V}_T|)$. Dans le pire des cas, comme $\mathcal{V}_{T_{max}} = (2 \cdot |\mathcal{I}|)^{|S_{max}|}$, alors la complexité de l'opération *MST* est la même que celle de l'opération *FTM* donnée par l'équation 3.1.

À partir de la complexité des opérations *FTM* et *MST*, nous donnons une complexité globale dans le pire des cas. Comme de nombreux algorithmes d'extraction de motifs, le problème dépend de la taille du résultat, i.e. du nombre de motifs extraits. Ainsi, la complexité globale est donnée par l'équation 3.2 en fonction de la taille de $\mathcal{M}_{G_{max}}$, i.e. l'ensemble maximal des motifs partiellement ordonnés clos pouvant être extraits.

$$\begin{aligned} &\Theta(|\mathcal{M}_{G_{max}}|) \cdot (\Theta(FTM) + \Theta(MST)) \\ &\qquad = \Theta(|\mathcal{M}_{G_{max}}| \cdot 2 \cdot (2 \cdot |\mathcal{I}|)^{|S_{max}|}) \end{aligned} \tag{3.2}$$

Évaluer $|\mathcal{M}_{G_{max}}|$ selon un seuil de support minimum θ Puisqu'il existe un seul motif partiellement ordonné clos pour un ensemble de séquences, nous pouvons estimer le nombre maximal de motifs clos pouvant être extraits d'une base de données de séquences $\mathcal{D}$, avec $|\mathcal{D}|$ le nombre de séquences dans $\mathcal{D}$. En fonction d'un support minimum θ, le nombre maximal de motifs clos pouvant être extraits de $\mathcal{D}$, noté $|\mathcal{M}_{G_{max}}|$, est équivalent au nombre de sous-ensembles de séquences $\mathcal{S} \subseteq \mathcal{D}$ tel que $|\mathcal{S}| \geq \theta$. Si $\theta = 1$, le cas est trivial. Le nombre de sous-ensembles est donné par $2^{|\mathcal{D}|}$. Comme nous ne considérons pas l'ensemble vide $\{\}$, $|\mathcal{M}_{G_{max}}| = 2^{|\mathcal{D}|} - 1$. Si $1 < \theta \leq |\mathcal{S}|$, le nombre de sous-ensembles de séquences est donné par la formule 3.3, qui calcule la somme des coefficients binomiaux $\binom{|\mathcal{D}|}{k}$ de k=θ à $|\mathcal{D}|$, qui donne, à chaque itération, le nombre de k-combinaisons de séquences dans $\mathcal{D}$.

$$|\mathcal{M}_{G_{max}}| = \sum_{k=\theta}^{|\mathcal{D}|} C_{|\mathcal{D}|}^{k} \tag{3.3}$$

3.5 Expérimentations

Des tests ont été effectués sur des jeux de données de séquences avec différentes caractéristiques. L'algorithme a été développé en C++. Les expérimentations ont été lancées sur un ordinateur portable équipé d'un processeur Intel Core i7 avec 8 Go de mémoire vive, s'exécutant sur Debian stable 7.0. Dans ces expérimentations,

nous étudions les performances de l'algorithme en comparant les versions avant et après optimisation.

Pour évaluer la performance de l'approche, nous avons sélectionné trois jeux de données : *Gazelle*, *Kosarak* et *Sign*. Ils ont été précédemment utilisés dans de nombreux articles sur la fouille de motifs séquentiels pour l'évaluation des performances [WH04, YHA03, PKSG05]. Ils sont accessibles en ligne [1] et peuvent être téléchargés dans le format SPMF, qui est un format qui représente de manière simple les séquences d'itemsets. Nous présentons maintenant chaque jeu de données :

Gazelle Ce premier jeu de données fut initialement proposé pour la KDD cup 2000 [2]. Il contient des données d'achats et de navigation venant du site Gazelle [3]. C'est un jeu de données peu dense qui décrit le comportement de milliers d'utilisateurs, où chaque séquence ordonne les pages Internet qui ont été explorées par un utilisateur.

Kosarak Ce très grand jeu de données contient des séquences de navigation d'utilisateurs anonymes fournies par un site d'information Hongrois. Peu d'informations sont disponibles sur ce jeu de données [4]. À cause du nombre important de séquences dans ce jeu (presque un million), nos expérimentations ont été effectuées sur un sous ensemble de 70 000 séquences.

Sign Ce dernier jeu de données est fourni par l'université de Boston [5]. Il contient des enregistrements de discours avec un segment de video associé, où chaque enregistrement correspond à un ensemble de gestes de la langue des signes américaine et à des expressions faciales.

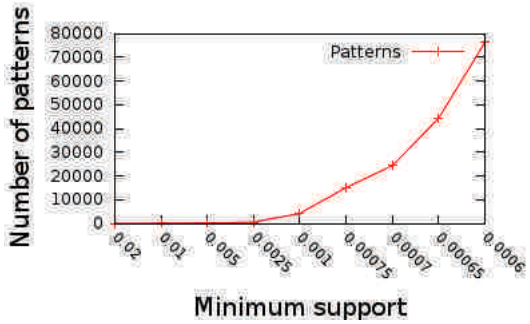
Ces trois jeux de données tests ont été sélectionnés car ils sont différents en terme de densité, de taille et d'hétérogénéité. Pour chacun d'entre eux, nous avons calculé (tableau 3.4) différentes statistiques comme le nombre total d'items et la longueur moyenne des séquences.

Nous donnons dans la figure 3.14 une idée du nombre de motifs partiellement ordonnés clos extraits en fonction du support minimum pour chaque jeu de données.

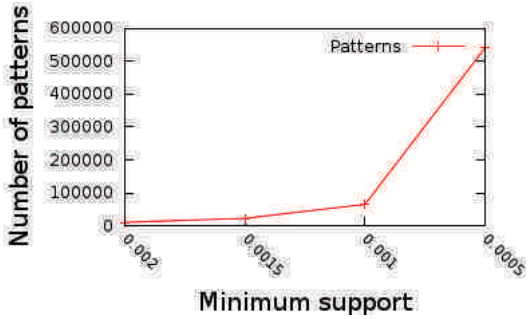
1. http ://www.philippe-fournier-viger.com/spmf/index.php ?link=datasets.php
2. http ://www.kdd.org/kdd-cup-2000-online-retailer-website-clickstream-analysis
3. https ://www.gazelle.com/
4. http ://fimi.ua.ac.be/data/
5. http ://cs-people.bu.edu/panagpap/Research/ asl_mining.htm

Jeu de données	**Sequences**	**Alphabet**	**Longueur moyenne des séquences**
Gazelle	59,601	497	2.51
Kosarak	70,000	21,144	7.98
Sign	730	267	51.99

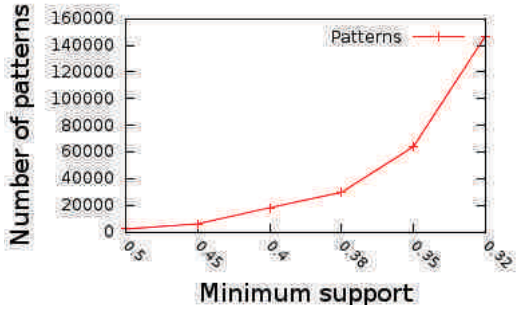
TABLE 3.4 – Statistiques sur les jeux de données



(a) Gazelle



(b) Kosarak

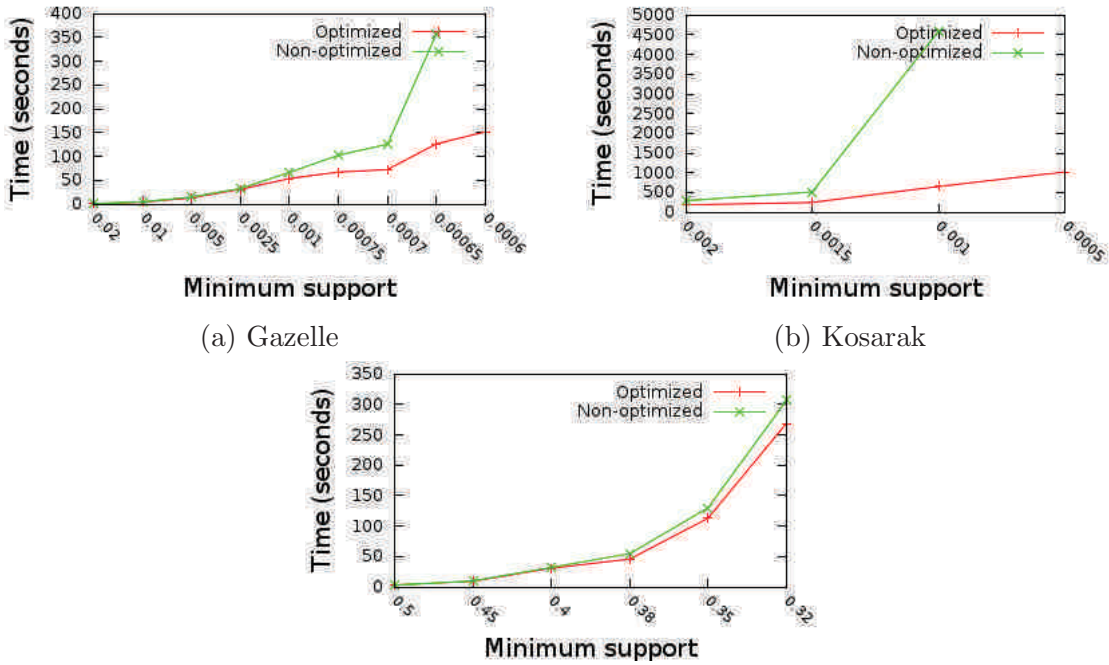


(c) Sign

FIGURE 3.14 – Nombre de motifs partiellement ordonnés clos en fonction du seuil de fréquence minimale

Dans ces expérimentations, nous avons représenté le seuil minimum d'extraction par la fréquence minimale plutôt que le support minimum. Par exemple, le jeu de données *Gazelle* contient 59 601 séquences. Une fréquence minimale égale à 0,001 signifie que nous avons extrait tous les motifs partiellement ordonnés qui sont supportés par au moins 60 séquences (support minimum) car : 59601 × 0,001 = 59,601. Comme dans toutes les problématiques de fouille de motifs, plus le seuil de fréquence minimale diminue, plus le nombre de motifs partiellement ordonnés clos augmente. Par exemple, pour le jeu de données *Sign*, 29 662 motifs partiellement ordonnés clos sont extraits avec un seuil de fréquence minimale de 0,38, et 147 004 sont extraits avec un seuil de fréquence minimale de 0,32.

Dans la figure 3.15, nous comparons le temps d'exécution de l'algorithme avec et sans optimisation pour les trois jeux de données (figures 3.15a, 3.15b et 3.15c). Nous observons que plus le seuil de fréquence minimale diminue, plus le gain obtenu avec l'optimisation est important. Par exemple, pour le jeu de données *Gazelle* avec un seuil de fréquence minimale de 0,005, la version optimisée est 1,14 fois plus rapide. Avec un seuil de fréquence minimale de 0,00065, elle est 2,83 fois plus rapide.



(a) Gazelle

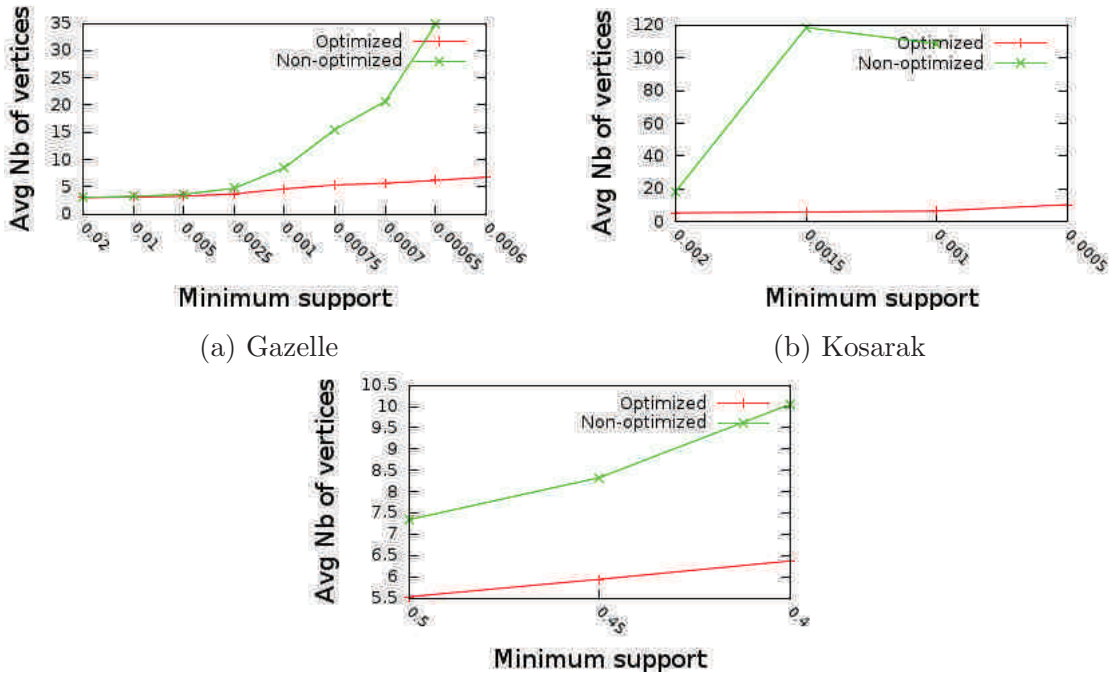
(b) Kosarak

(c) Sign

FIGURE 3.15 – Temps de calcul en fonction du seuil de fréquence minimale

Nous nous sommes aperçu que la différence de temps d'extraction entre les deux approches était liée à la différence du nombre moyen de sommets dans les mo-

tifs partiellement ordonnés avant l'application de l'opération *MergingSuffixTree*. Par exemple, dans le jeu de données *Kosarak* avec un seuil de fréquence minimale de 0,001, le nombre moyen de sommets dans la version non optimisée est 16,64 fois plus important que dans la version optimisée avec un temps d'exécution 3,84 fois plus faible. Dans le jeu de données *Sign*, nous observons que la différence du nombre moyen de sommets est faible entre les deux versions, e.g. 1,40 fois plus faible à un seuil de fréquence minimale de 0,45. Ce qui se traduit par un temps d'exécution 1,08 fois plus faible. La figure 3.16b illustre un comportement intéressant avec la décroissance du nombre moyen de sommets pour la version de base à un seuil de fréquence minimale de 0,001. Une hypothèse est que les motifs partiellement ordonnés clos ayant une fréquence comprise entre 0,0015 et 0,001 sont en moyenne plus petits que ceux ayant une fréquence supérieure à 0,0015.



(a) Gazelle

(b) Kosarak

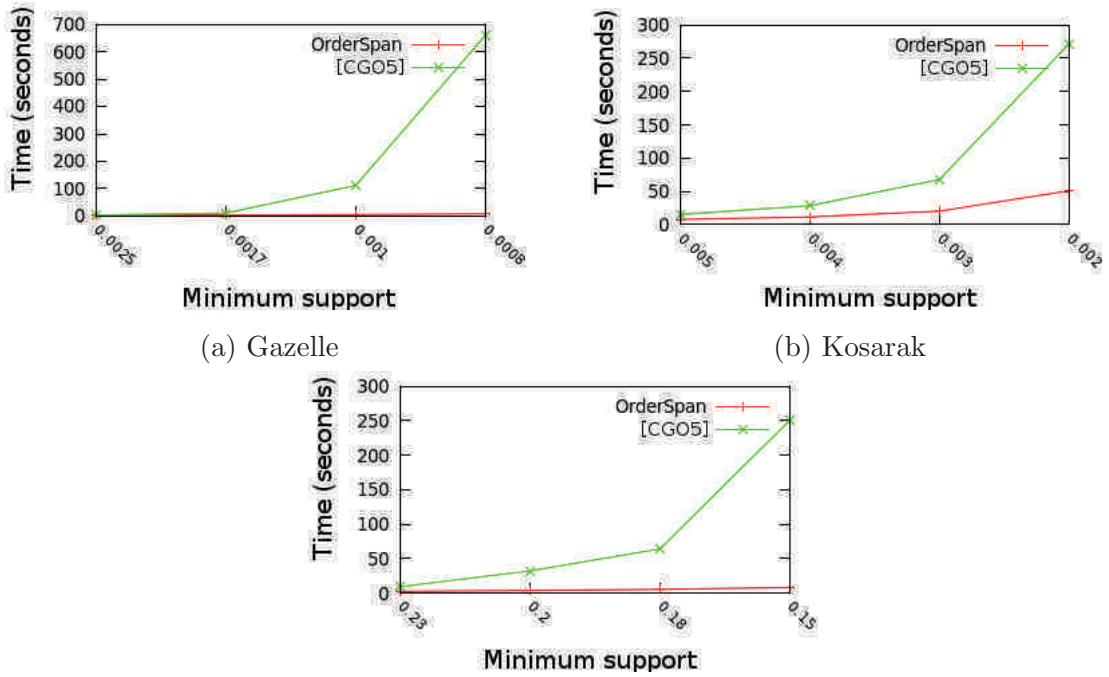
(c) Sign

FIGURE 3.16 – Nombre moyen de sommets en fonction du seuil de fréquence minimale

Pour la version non-optimisée, les expérimentations pour le jeu de données *Gazelle* au seuil de fréquence minimale de 0,0006, et pour le jeu de données *Kosarak* au seuil de fréquence minimale de 0,0005 ne sont pas présentées à cause d'un temps de calcul trop important.

L'efficacité de la version optimisée dépend pour beaucoup du jeu de données. Par exemple, le jeu de données *Sign* est moins sensible à l'optimisation que les jeux de données *Gazelle* et *Kosarak*.

Nous avons également comparé l'efficacité de notre approche avec celle proposée dans [CG05]. Cette dernière, comme expliqué dans la section 3.2.1, est un post-traitement appliqué à un ensemble de motifs séquentiels clos précédemment extraits avec l'algorithme d'extraction de motifs séquentiels clos *BIDE* [WH04]. Et contrairement à *OrderSpan*, elle ne permet pas d'extraire la totalité des motifs partiellement ordonnés clos. Ainsi, pour effectuer une comparaison viable, nous avons effectué une modification sur *OrderSpan* pour extraire le même ensemble de motifs partiellement ordonnés que l'approche dans [CG05]. Une telle comparaison à l'avantage de comparer un algorithme qui extrait directement les motifs partiellement ordonnés clos (*OrderSpan*) à un algorithme qui utilise un post-traitement (méthode dans [CG05]). Les figures 3.17a, 3.17b et 3.17c nous fournissent les résultats obtenus sur les trois jeux de données précédemment utilisés. Dans ces expérimentations, nous avons utilisé la version optimisée d'*OrderSpan*.



(a) Gazelle

(b) Kosarak

(c) Sign

FIGURE 3.17 – Comparaison entre *OrderSpan* et l'approche dans [CG05]

Prenons le jeu de données *Kosarak* avec un seuil de fréquence minimale de 0,003, l'algorithme dans [CG05] s'exécute en 67,97 secondes et *OrderSpan* 20,37 secondes. Sur le jeu de données *Gazelle* avec un seuil de fréquence minimale de 0,001, l'algorithme dans [CG05] s'exécute en 111,09 secondes et *OrderSpan* 5,43 secondes. Sur ce dernier exemple *OrderSpan* est 20 fois plus rapide. Ces résultats montrent ainsi que fouiller les motifs partiellement ordonnés directement dans les données est bien plus efficace que d'effectuer un post-traitement sur un ensemble de motifs séquentiels. La raison est la suivante, lorsque le nombre de motifs séquentiels clos devient important (plusieurs milliers), une approche par post-traitement devient trop lourde car une comparaison pour de nombreuses paires de motifs est nécessaire.

3.6 Synthèse

Dans ce chapitre nous avons présenté l'algorithme *OrderSpan*. Il permet d'extraire l'ensemble complet des motifs partiellement ordonnés clos et peut s'appliquer à n'importe quel type de séquences, i.e aux séquences composées d'itemsets contenants des items pouvant se répéter plusieurs fois.

L'approche est à la fois basée sur les propriétés des préfixes fréquents dans les bases de données de séquences (opération *ForwardTreeMining*) et des propriétés sur les suffixes fréquents (opération *MergingSuffixTree*). Le processus d'extraction est ensuite amélioré en adaptant la propriété des bases projetées équivalentes, utilisée en fouille de motifs séquentiels clos, durant l'opération *ForwardTreeMining* pour élaguer l'espace de recherche.

Les expérimentations montrent le gain obtenu avec l'optimisation, qui est fortement dépendant du jeu de données utilisé, mais dans tous les cas les performances sont meilleures. De plus, les résultats montrent qu'extraire directement les motifs partiellement ordonnés dans les données est bien plus efficace qu'une approche basée sur un post-traitement appliqué à un ensemble de motifs séquentiels clos.

CHAPITRE 4

Sélection de k motifs d'intérêt

4.1 Introduction

Les travaux qui traitent de méthodes pour l'extraction de motifs sont très nombreux. De nouveaux algorithmes sont constamment développés pour surpasser les anciens, que se soit en terme de performances en temps de calcul ou bien en terme de consommation mémoire. Ainsi, il n'est pas rare d'avoir la possibilité d'extraire jusqu'à plusieurs centaines de milliers de motifs en quelques secondes ou minutes. Cependant, lorsque ces algorithmes sont appliqués à des données réelles, une même problématique revient : le nombre de motifs extraits est bien trop volumineux. Leur restitution et leur analyse par les utilisateurs, souvent externes au domaine de la fouille de données, est alors difficile. Dans ce contexte : *comment identifier les motifs les plus intéressants lorsque plusieurs milliers ont été extraits ?*

La notion d'intérêt est propre à chaque problématique. La nôtre s'inscrit dans le cadre de la fouille de motifs partiellement ordonnés dans des bases de données partitionnées en plusieurs classes. Pour considérer le problème du volume important de motifs extraits à analyser, nous définissons dans ce chapitre l'algorithme *Pattern-Balanced*, qui permet de filtrer les k motifs partiellement ordonnés clos discriminants les plus intéressants. Nous basons la notion d'intérêt d'un motif sur une combinaison de trois critères. En effet, après avoir effectué certains tests et après discussions avec les utilisateurs finaux, nous avons identifié trois critères pour la sélection des motifs partiellement ordonnés intéressants : la fréquence, la discrimination et la non-redondance. Même si la méthode présentée ici est appliquée spécifiquement aux motifs partiellement ordonnés clos, elle peut s'appliquer à tout type de motif dans le contexte d'une base de donnée partitionnée en classes.

Tout d'abord, nous présentons les motivations de ce travail dans la section 4.2. Ensuite, la section 4.3 introduit la définition du *generalized growth rate* qui adapte la notion de *growth rate* [DL99] au cas de bases de données composées de n classes.

Nous avons ainsi pu adapter *OrderSpan* pour la fouille de motifs discriminants. La problématique de l'identification de k motifs d'intérêt est illustrée sur un cas réel dans la section 4.4. L'algorithme *Pattern-Balanced* est ensuite présenté dans la section 4.5 et pour finir, des expérimentations sont présentées dans la section 4.7.

4.2 Motivations

Plusieurs auteurs ont travaillé sur l'extraction des k motifs les plus intéressants comme alternative à l'extraction de l'ensemble complet des motifs. Cependant, comme nous allons le voir, leurs objectifs sont différents des nôtres.

Dans [HWLT02] et [TYH05], les auteurs proposent deux algorithmes qui permettent d'extraire les *Top-k* motifs les plus fréquents d'une taille supérieure à un seuil de longueur minimale. L'avantage de ces méthodes est qu'elles ne prennent pas en paramètre un seuil de support minimum, souvent difficile à déterminer. Toutefois, elles nécessitent un autre paramètre, la taille minimale d'un motif. Pour pallier ce problème, d'autres auteurs [DF07, Den14] ont travaillé sur les *Top-rank-k* motifs. Ces approches visent tout simplement à extraire les k motifs les plus fréquents dans une base de données, sans considérer leur longueur. Dans le contexte des motifs discriminants, et plus spécifiquement celui des *Jumping patterns* (les motifs qui sont seulement supportés par les objets d'une seule classe), une approche pour extraire les k *Jumping patterns* les plus supportés par une classe en comparaison d'une autre est proposée dans [TW08].

Nous voyons que toutes les approches proposées se focalisent principalement sur un seul critère à la fois. Certaines sélectionnent les k motifs les plus fréquents alors que d'autres filtrent les k motifs les plus discriminants au travers des *Jumping patterns*. Ainsi, la problématique est la suivante : *comment faire lorsque plusieurs critères sont d'intérêt pour les utilisateurs ?*

En effet, dans notre cas, nous avons identifié les trois critères suivants :

1. **Fréquence :** elle est définie par le nombre de séquences dans chaque classe de la base de données qui supportent un motif en fonction du nombre total de séquences dans ces classes. Les utilisateurs finaux sont intéressés par les motifs les plus fréquents dans chaque classe de la base de données car ces motifs décrivent mieux une classe donnée que les motifs avec une faible fréquence.
2. **Discrimination :** ce critère est dans notre cas défini par la valeur de *growth rate* dans chaque classe de la base de données. Pour rappel, cette mesure cal-

cule un rapport sur la fréquence d'un motif dans une classe par rapport à une autre (voir chapitre 2). Les utilisateurs finaux sont généralement intéressés par les motifs les plus discriminants dans une classe, i.e. avec une forte valeur de *growth rate* pour cette classe, correspondant aux motifs qui sont plus fréquents dans une classe par rapport à une autre.

3. **Redondance :** en fouille de motifs, il est fréquent de trouver un ensemble de motifs porteurs d'informations proches. Cela est souvent lié au fait que ces motifs sont supportés par des ensembles de séquences de la base de données qui sont très proches. Ainsi, nous définissons la redondance par le fait que deux motifs soient supportés par un nombre élevé de séquences communes. Réduire cette redondance dans les résultats permet d'améliorer l'analyse des motifs. Plus la redondance entre deux motifs est faible, plus ils sont susceptibles de donner une information différente sur la base.

Ainsi, à partir de plusieurs milliers de motifs, le résultat idéal est un sous-ensemble de k motifs qui sont à la fois les plus fréquents et les plus discriminants, tout en restituant toute la diversité de l'information extraite. Néanmoins, comme nous allons le voir, sélectionner un tel sous-ensemble de motifs d'intérêt dans plusieurs milliers de motifs n'est pas une tâche évidente.

4.3 Définitions préalables

Avant d'introduire la méthode de sélection des k motifs, nous présentons la définition du *generalized growth rate*. Cette mesure d'intérêt étend la mesure du *growth rate* à plus de deux classes. Cette définition permet alors de rechercher des ensembles de motifs discriminants dans n classes d'une base de données. Cette définition nous a permis de proposer une adaptation d'*OrderSpan* à la fouille de motifs partiellement ordonnés clos discriminants. Ces deux aspects sont nécessaires à l'application de l'algorithme *Pattern-Balanced* aux motifs partiellement ordonnés clos discriminants (voir section 4.5.2).

4.3.1 *Generalized growth rate*

Le *growth rate* s'applique très bien lorsque l'on a deux classes dans une base de données. Cependant, il est fréquent d'avoir un nombre n de classes. Le *growth rate* pouvant seulement être obtenu à partir du rapport entre deux fréquences, il est nécessaire de l'adapter au cas d'un nombre de classes supérieur à deux. Une

possibilité est d'utiliser une approche basée sur une version globale du *growth rate*, que nous appelons *generalized growth rate*. L'idée est de calculer le *growth rate* d'un motif dans une classe en fonction de la valeur de fréquence la plus forte dans les autres classes. La définition 24 introduit cette mesure généralisée.

Définition 24 *Soient un motif M, une classe $\mathcal{C}$ et un ensemble d'autres classes $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, le generalized growth rate de M dans $\mathcal{C}$ par rapport à $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, noté GGR($M$,$\mathcal{C}$,$\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$), est calculé de la manière suivante :*

$$\begin{cases} \text{- } 0, \textit{ si } Frequence_{\mathcal{C}}(M) = 0 \textit{ et } max(Frequence_{\mathcal{C}_1}(M), \\ Frequence_{\mathcal{C}_2}(M), \ldots, Frequence_{\mathcal{C}_n}(M)) = 0 \\ \text{- } \infty, \textit{ si } Frequence_{\mathcal{C}}(M) \neq 0 \textit{ et } max(Frequence_{\mathcal{C}_1}(M), \\ Frequence_{\mathcal{C}_2}(M), \ldots, Frequence_{\mathcal{D}_n}(M)) = 0 \\ \text{- } \frac{Frequence_{\mathcal{C}}(M)}{max(Frequence_{\mathcal{C}_1}(M), Frequence_{\mathcal{C}_2}(M), \ldots, Frequence_{\mathcal{C}_n}(M))}, \textit{ sinon} \end{cases} \quad (4.1)$$

Cette version généralisée du *growth rate* permet de savoir, pour un motif M, si celui-ci est plus fréquent dans une classe par rapport à toutes les autres classes de la base. Si un motif M a une valeur de *generalized growth rate* de 5 dans une classe, cela signifie qu'il est au moins cinq fois plus fréquent dans cette classe que dans les autres. Une valeur infinie dans une classe signifie que le motif n'est supporté par aucun objet des autres classes. Comme pour le *growth rate* classique, une valeur strictement supérieure à 1 signifie qu'un motif est discriminant. Un motif ne peut être discriminant que dans une seule classe à la fois.

4.3.2 *OrderSpan* pour la fouille de motifs discriminants

À partir de la définition du *generalized growth rate* (définition 4.1), nous avons pu adapter *OrderSpan* pour la fouille de motifs discriminants dans n classes. Cette adaptation est un processus en deux étapes :

1. extraire, pour chaque classe d'une base de données de séquences, l'ensemble des motifs partiellement ordonnés clos selon un support minimum θ ;
2. parmi les motifs partiellement ordonnés clos extraits dans chaque classe, filtrer seulement les motifs discriminants, i.e. avec une valeur de *generalized growth rate* supérieure à 1.

Nous donnons maintenant l'algorithme *DiscriminantOrderSpan* associé à ce processus.

Algorithm 3: DiscriminantOrderSpan

Entrées: $\mathcal{D}$ une base de données composées de plusieurs classes, θ un support minimum

Sorties: $\mathcal{MC}_{G_{Discr}}$ l'ensemble des ensembles de motifs partiellement ordonnés clos discriminants dans chaque classe de $\mathcal{D}$

```
1  MC_{G_Discr} = ∅;
2  pour chaque Classe C dans D faire
3      M_G = ∅ # l'ensemble des motifs extraits dans C;
4      M_G = OrderSpan(C, θ);
5      M_{G_Discr} = ∅ # l'ensemble des motifs discriminants dans C;
6      pour chaque Motif M_G dans M_G faire
7          si M_G est discriminant, i.e. une valeur de generalized growth rate
           supérieure à 1, dans C par rapport aux autres classes dans D alors
8              Ajouter M_G à M_{G_Discr};
9
10     fin
11     Ajouter l'ensemble des motifs discriminants M_{G_Discr} à MC_{G_Discr};
12 fin
13 Retourner MC_{G_Discr};
```

Ligne 1 : Initialisation de $\mathcal{MC}_{G_{Discr}}$, la liste contenant les motifs discriminants pour chaque classe de la base de données $\mathcal{D}$. **Lignes 2-4** : Pour chaque classe $\mathcal{C}$, l'ensemble des motifs partiellement ordonnés clos $\mathcal{M}_G$ est généré. **Lignes 5-10** : Initialisation de $\mathcal{M}_{G_{Discr}}$, la liste des motifs discriminants dans $\mathcal{C}$. Ensuite, pour chaque motif M_G dans $\mathcal{M}_G$, on vérifie s'il est discriminant. Si tel est le cas, il est ajouté à $\mathcal{M}_{G_{Discr}}$. **Ligne 11** : $\mathcal{M}_{G_{Discr}}$ est ajouté à $\mathcal{MC}_{G_{Discr}}$, l'ensemble des ensembles de motifs discriminants pour chaque classe. **Ligne 13** : $\mathcal{MC}_{G_{Discr}}$ est retourné.

4.4 Exemple illustratif

Pour démontrer toute la complexité de sélectionner k motifs intéressants selon la fréquence, la discrimination et la non-redondance, nous illustrons ce problème avec un exemple concret. Nous avons utilisé le jeu de données *Fresqueau* qui est décrit en détail dans le chapitre 6. Brièvement, ce jeu de données contient trois sous-jeux

de données composés de plusieurs classes relatives à la qualité biologique du milieu aquatique. Ces sous-jeux de données sont *IBGN*, *IBD* et *IPR*. Et chaque sous-jeu de données est composé de cinq classes dont le nom des classes est une couleur : *Bleu*, *Vert*, *Jaune*, *Orange* et *Rouge*. Les items dans les séquences de ce jeu de données sont des mesures physico-chimiques de l'eau. Pour plus d'informations sur les données ainsi que pour des exemples de motifs, le lecteur est invité à consulter les chapitres 6 et 7.

Pour cet exemple illustratif, nous avons extrait avec *DiscriminantOrderSpan* l'ensemble complet des motifs partiellement ordonnés clos discriminants dans la classe rouge de l'*IBGN* avec un seuil de fréquence minimale de 10%. La valeur de discrimination a été calculée en utilisant le *generalized growth rate* (définition 24) car chaque sous-jeu de données (ici l'*IBGN*) est composé de cinq classes.

La figure 4.1 montre la projection d'informations concernant cet ensemble de motifs discriminants extraits. La projection de notre exemple est obtenue en appliquant un algorithme de type multidimentional scaling [BG97], qui permet de représenter la dissimilarité entre plusieurs paires de motifs dans un espace géométrique en deux dimensions. Les motifs sont représentés graphiquement par des points munis de coordonnées pour être visualisés. Dans notre cas, les points représentent des motifs et la dissimilarité entre les motifs est obtenue en calculant la distance de *Hamming* [Ham50] sur les séquences qui les supportent (expliqué ci-dessous).

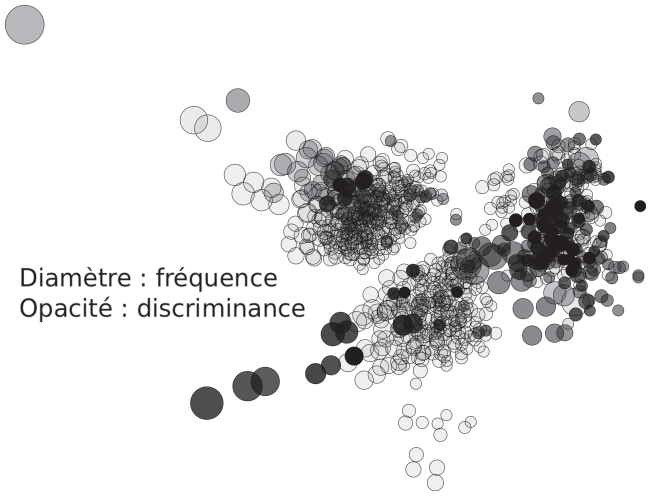


FIGURE 4.1 – Informations sur les motifs partiellement ordonnés clos discriminants extraits de la classe rouge de l'*IBGN* du jeu de données *Fresqueau*

Dans cet espace à deux dimensions, chaque cercle représente un motif discriminant. Son diamètre est proportionnel à la fréquence du motif dans la classe rouge de

l'*IBGN* et son opacité est proportionnelle à sa discrimination dans cette même classe par rapport aux autres (*generalized growth rate*). Plus deux cercles sont éloignés l'un de l'autre, plus les motifs correspondants sont supportés par des ensembles différents de séquences de la classe. Pour représenter cet aspect, nous avons codé l'inclusion des motifs dans une classe par un code binaire. Par exemple, supposons que la classe rouge de l'*IBGN* contienne six séquences. Un motif représenté par le code binaire 011101 signifie qu'il est supporté par toutes les séquences de la classe rouge de l'*IBGN*, à l'exception de la première et de la cinquième séquence (la valeur 1 signifie que le motif est supporté, 0 sinon).

Pour calculer la distance entre les codes binaires de deux motifs, nous avons utilisé la distance de *Hamming*. À partir de deux vecteurs composés de valeurs binaires, cette distance permet de calculer le nombre d'éléments qui diffèrent entre ces deux vecteurs. Elle est introduite par la définition 25.

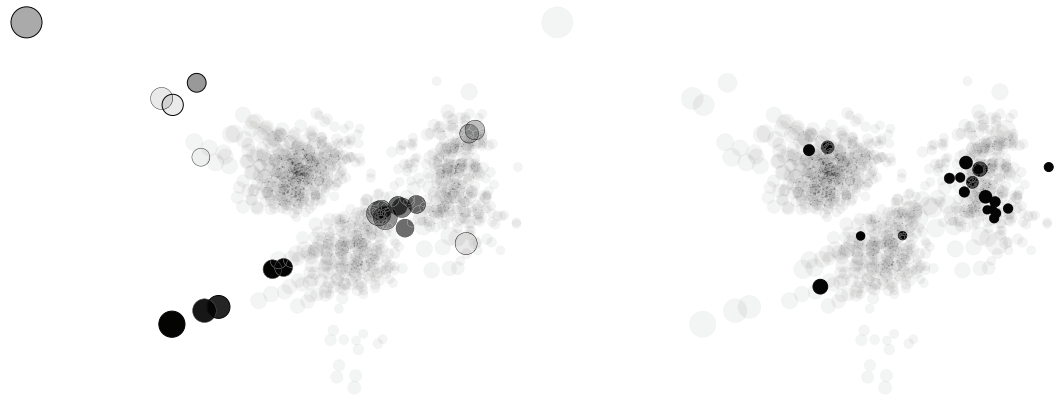
Définition 25 ***Distance de Hamming*** *Soient $\Sigma = \{0, 1\}$ un alphabet binaire, deux vecteurs binaires A et B à n valeurs dans Σ et $\oplus$ désignant le ou exclusif. La distance de Hamming s'écrit :*

$$Hamming(A, B) = \sum_{i=1}^{n} (A_i \oplus B_i) \tag{4.2}$$

Par exemple, la distance de *Hamming* entre les codes binaires 010101 et 011100 est de 2 car ces deux codes sont différents en deux endroits, aux troisième et sixième indices.

Dans la projection, nous observons que les motifs les plus fréquents sont rarement les plus discriminants et inversement, les motifs les plus discriminants sont rarement les plus fréquents. Également, nous voyons que de nombreux motifs sont regroupés en clusters, à peu près trois clusters dans l'exemple. Cela signifie que dans l'ensemble des motifs partiellement ordonnés clos discriminants extraits, nombreux sont les motifs qui sont supportés par des ensembles de séquences proches.

Maintenant, à partir de cet ensemble de motifs extraits, l'objectif est de sélectionner les k motifs qui maximisent la fréquence et la discrimination, tout en étant peu redondants. Prenons les figures 4.2a et 4.2b. Basées sur la figure 4.1, elles représentent respectivement la sélection des 20 motifs les plus fréquents et la sélection des 20 motifs les plus discriminants dans la classe rouge de l'*IBGN*. Les motifs qui n'ont pas été sélectionnés sont floutés.



(a) (b)

FIGURE 4.2 – (a) Les 20 motifs les plus fréquents et (b) les 20 motifs les plus discriminants

Nous observons ici que les 20 motifs les plus fréquents sont différents des 20 motifs les plus discriminants et inversement.

Ces deux méthodes de sélection simples présentent le même inconvénient. Nous voyons dans les projections qu'elles sélectionnent des motifs qui sont parfois très similaires (faible distance de *Hamming* de leurs codes binaires), ce qui fait qu'ils sont potentiellement redondants car supportés par des ensembles de séquences proches dans la classe rouge de l'*IBGN*.

4.5 Combinaison de plusieurs dimensions d'analyse

Pour améliorer la sélection des motifs, nous avons choisi de combiner les critères de fréquence, de discrimination et de non-redondance, afin de calculer un score d'intérêt pour chaque motif. Les valeurs de fréquence et de discrimination sont propres à chaque motif. Par contre, la valeur de non-redondance se calcule en fonction des autres motifs qui ont été déjà sélectionnés. Soient $\mathcal{M}_{G_{Discr}}$ un ensemble de motifs extraits et $\mathcal{M}_{G_{Selec}}$ un ensemble de motifs sélectionnés. La problématique est la suivante : comment trouver le meilleur motif $M_G \in \mathcal{M}_{G_{Discr}}$ a ajouter à $\mathcal{M}_{G_{Selec}}$, i.e. qui soit le moins redondant, le plus fréquent et le plus discriminant ? Pour cela, nous calculons pour chaque motif $M_G \in \mathcal{M}_{G_{Discr}}$ un score, entre 0 et 1, qui combine les scores des différents critères d'intérêt.

Avant de présenter l'algorithme de sélection, nous normalisons chaque critère d'intérêt.

4.5.1 Normalisation

Afin de donner le même poids à chaque critère, nous avons choisi de les normaliser pour calculer une valeur entre 0 et 1 pour chacun d'eux. Plus la valeur est proche de 1, plus le motif est intéressant pour le critère donné. La fréquence est déjà un score entre 0 et 1 où un motif qui a une fréquence égale à 0,2 signifie qu'il est supporté par 20% des données d'une classe. Seul le *generalized growth rate* et le calcul de la distance de *Hamming* sont normalisés.

Normalisation du *generalized growth rate*

La version normalisée du *generalized growth rate* d'un motif M_G dans une classe $\mathcal{C}$ par rapport à un ensemble d'autres classes $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, notée $NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})$, est définie par l'équation 4.3.

$$\begin{cases} \text{- 0 si } GGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}) = 0 \\ \text{- 1, si } GGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})) = \infty \\ \text{- } 1 - \frac{1}{GGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}))}, \text{ sinon} \end{cases} \tag{4.3}$$

Plus la valeur du *generalized growth rate* tend vers l'infini, plus sa version normalisée tend vers 1. Cette dernière est égale à 1 si la valeur du *generalized growth rate* est égale à l'infini.

Normalisation de la distance de *Hamming*

La valeur de la distance de *Hamming* entre les codes binaires de deux motifs est bornée par le nombre total de séquences dans les données. Par exemple, si la classe est composée de six séquences, le nombre maximal d'indices où les codes sont potentiellement différents est de 6. Le code binaire d'un motif M_G est noté $Code(M_G)$. Soit $Hamming(Code(M_{G_1}), Code(M_{G_2}))$ la distance de *Hamming* entre les codes binaires de deux motifs M_{G_1} et M_{G_2}, avec $\mathcal{S}$ l'ensemble total de séquences dans la classe où les motifs sont discriminants. La distance de *Hamming* normalisée, notée $NH(Code(M_{G_1}), Code(M_{G_2}))$, est définie par l'équation 4.4.

$$NH(Code(M_{G_1}), Code(M_{G_2})) = \frac{Hamming(Code(M_{G_1}), Code(M_{G_2}))}{|\mathcal{S}|} \tag{4.4}$$

Ainsi, plus la distance de *Hamming* entre les codes binaires de deux motifs est importante, plus le score de la distance normalisée est proche de 1. Au maximum, toutes les valeurs dans les codes sont différentes et alors $Hamming(Code(M_{G_1}), Code(M_{G_2}))$ $= |\mathcal{S}|$ et $NH(Code(M_{G_1}), Code(M_{G_2})) = 1$.

4.5.2 L'algorithme Pattern-Balanced : sélection des k motifs les plus équilibrés

Maintenant que chaque critère est normalisé, nous présentons un algorithme qui permet, à partir d'un ensemble $\mathcal{M}_{G_{Discr}}$ de motifs dicriminants extraits et d'un paramètre k fourni en entrée, de sélectionner un ensemble de k motifs qui sont à la fois fréquents, discriminants et peu redondants entre eux. À cause du nombre de motifs qui peut être extrait et de la combinatoire qui en découle, l'algorithme proposé est une heuristique incrémentale de type glouton. À partir d'un ensemble de motifs extraits, il effectue k itérations pour choisir à chaque fois un nouveau motif à ajouter dans la sélection. Les deux principales opérations de cet algorithme sont :

1. l'initialisation de l'ensemble des motifs sélectionnés ;
2. chaque itération, où l'ajout d'un nouveau motif se base sur l'ensemble des motifs précédemment sélectionnés.

Nous donnons maintenant le pseudo code de l'algorithme *Pattern-Balanced* (algorithme 4).

Lignes 1-4 : Initialisation de la liste des motifs sélectionnés $\mathcal{M}_{G_k}$ avec un motif M_G sélectionné dans la liste des motifs extraits $\mathcal{M}_{G_{Discr}}$. Suppression du motif M_G dans $\mathcal{M}_{G_{Discr}}$. **Lignes 5-9** : boucle qui itère $k-1$ fois. À chaque itération un nouveau motif M_G est sélectionné dans $\mathcal{M}_{G_{Discr}}$ et inséré dans $\mathcal{M}_{G_k}$. M_G est ensuite supprimé de $\mathcal{M}_{G_{Discr}}$. La sélection effectuée à l'initialisation est différente de celle effectuée dans la boucle. Ces deux étapes de sélection sont maintenant présentées.

Initialisation

Cette première étape consiste à initialiser l'ensemble résultat $\mathcal{M}_{G_k}$ en choisissant un premier motif dans l'ensemble des motifs extraits $\mathcal{M}_{G_{Discr}}$. Comme à cette étape $\mathcal{M}_{G_k}$ est vide, son initialisation ne prend pas en compte le critère de redondance puisque qu'aucun motif n'a précédemment été sélectionné. Seuls les critères de fréquence et de discrimination sont alors considérés. Un rang est donné à chaque motif $M_G \in \mathcal{M}_{G_{Discr}}$, selon un score calculé à partir de la formule de l'équation 4.5,

Algorithm 4: Pattern-Balanced

Entrées: $\mathcal{M}_{G_{Discr}}$ un ensemble de motifs discriminants dans une classe $\mathcal{C}$ par rapport à un ensemble d'autres classes $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, k le nombre de motifs à sélectionner

Sorties: $\mathcal{M}_{G_k}$ un ensemble de motifs sélectionnés

$\mathcal{M}_{G_k} = \emptyset$;
$M_G = init\_selection(\mathcal{M}_{G_{Discr}}, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})$;
ajouter M_G dans $\mathcal{M}_{G_k}$;
supprimer M_G de $\mathcal{M}_{G_{Discr}}$;
pour i de 1 à $k-1$ **faire**
    $M_G = selection(\mathcal{M}_{G_{Discr}}, \mathcal{M}_{G_k}, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})$;
    ajouter M_G dans $\mathcal{M}_{G_k}$;
    supprimer M_G de $\mathcal{M}_{G_{Discr}}$;
fin
retourner $\mathcal{M}_{G_k}$;

avec $Frequence_{\mathcal{C}}(M_G)$ la fréquence dans la classe $\mathcal{C}$ et $NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})$ la normalisation du *generalized growth rate* dans la classe $\mathcal{C}$.

$$init\_score(M_G) = \frac{Frequence_{\mathcal{C}}(M_G) \times NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})}{Frequence_{\mathcal{C}}(M_G) + NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\})} \times 2 \quad (4.5)$$

Puisque les critères sont normalisés entre 0 et 1, la valeur retournée par cette équation est également comprise entre 0 et 1 grâce au facteur multiplicatif de 2. Une telle équation à l'avantage de pénaliser les motifs qui ont par exemple une très faible valeur de fréquence (motifs rares) ou bien une très faible valeur de *generalized growth rate* (motif peu discriminant). Ainsi, seront par exemple privilégiés les motifs qui ont des valeurs moyennes et équilibrées, à d'autres motifs qui ont un score excellent pour un des deux critères, mais très faible pour le second.

Le motif ayant le plus fort score est sélectionné pour initialiser $\mathcal{M}_{G_k}$.

Itération

Après initialisation, la liste des motifs sélectionnés $\mathcal{M}_{G_k}$ est itérativement complétée en ajoutant cette fois-ci le critère de redondance. L'objectif est maintenant de sélectionner dans $\mathcal{M}_{G_{Discr}}$ un motif qui est à la fois fréquent, discriminant et également peu redondant par rapport aux motifs précédemment sélectionnés.

Pour cela, nous calculons pour chaque motif $M_G \in \mathcal{M}_{G_{Discr}}$ un score de non-redondance en se basant sur l'équation 4.4. Ce score est égal à la valeur minimale de la distance de *Hamming* normalisée entre le code binaire de M_G et les codes de l'ensemble des motifs sélectionnés $\mathcal{M}_{G_k}$.

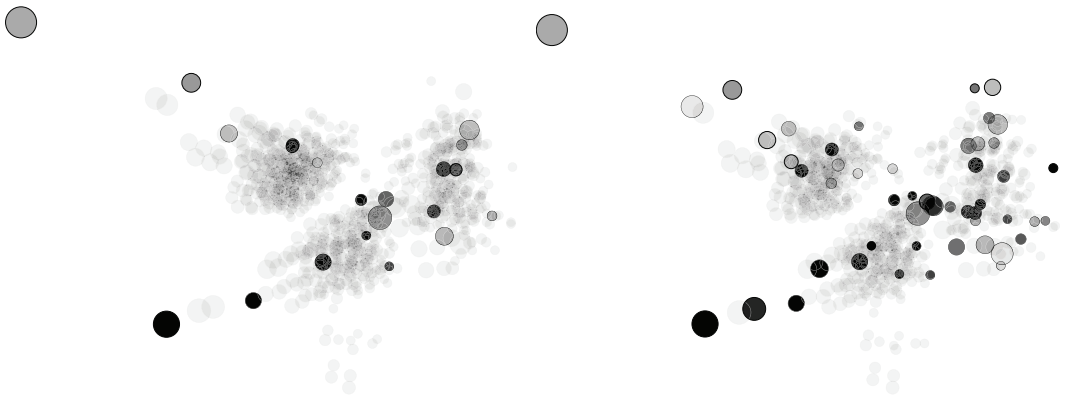
Nous notons $NR(M_G, \mathcal{M}_{G_k})$ le score de non-redondance défini par $NR(M_G, \mathcal{M}_{G_k}) = min\{NH(Code(M_G), Code(M'_G))|M'_G \in \mathcal{M}_{G_k}\}$. L'ajout de ce nouveau critère dans l'équation 4.5 donne l'équation 4.6.

$$score(M_G, \mathcal{M}_{G_k}) = \frac{Frequence_{\mathcal{C}}(M_G) \times NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}) \times NR(M_G, \mathcal{M}_{G_k})}{Frequence_{\mathcal{C}}(M_G) + NGGR(M_G, \mathcal{C}, \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}) + NR(M_G, \mathcal{M}_{G_k})} \times 3 \tag{4.6}$$

Alors que l'équation 4.5 servait à maximiser les motifs avec une forte fréquence et une forte discrimination, cette nouvelle équation permet également de maximiser la distance minimale avec les motifs déjà sélectionnés. Cette approche est en fin de compte proche d'une approche par clustering, méthode qui consiste a regrouper les éléments les plus proches, excepté que nous souhaitons à l'inverse sélectionner l'élément (motif) le plus distant. Tout comme pour l'équation 4.5, la formule 4.6 privilégie les motifs qui ont de bonnes valeurs pour les trois critères à la fois, et le facteur multiplicatif de 3 permet d'obtenir une valeur entre 0 et 1.

Nous illustrons l'approche en reprenant l'exemple de la figure 4.1 avec les motifs extraits à partir de la classe IBGN^{Rouge} du jeu de données *Fresqueau*. Les figures 4.3a et 4.3b montrent les tests effectués en sélectionnant les 20 et les 50 motifs les plus équilibrés selon la procédure de sélection définie ci-avant.

Nous observons que les motifs sélectionnés avec l'algorithme *Pattern-Balanced* sont plus diversifiés (répartis dans la projection) et donc moins redondants que la sélection basée sur les motifs les fréquents ou les plus discriminants. Nous observons également qu'ils sont équilibrés entre motifs fréquents et motifs discriminants (cf. figures 4.2a et 4.2b). Pour finir, nous observons qu'augmenter la valeur du paramètre k permet de capturer un peu plus la diversité d'un ensemble de motifs extraits tout en maîtrisant la redondance.



(a) (b)

FIGURE 4.3 – (a) Les 20 et (b) les 50 motifs les plus équilibrés

4.6 Complexité

Cet algorithme a une complexité de l'ordre de :

$$\sum_{i=0}^{k-1} i \times (\mathcal{M}_{G_{Discr}} - i) \times |C| + (\mathcal{M}_{G_{Discr}} - i) \tag{4.7}$$

où $\mathcal{M}_{G_{Discr}}$ est le nombre de motifs discriminants extraits d'une classe C, et $k > 0$ le nombre de motifs à sélectionner parmi eux. À chaque étape, les i motifs précédemment sélectionnés sont comparés deux à deux aux $(\mathcal{M}_{G_{Discr}} - i)$ motifs restants en calculant la distance de *Hamming* qui est linéaire en fonction de $|C|$ (taille des codes binaires). Ensuite, les $(\mathcal{M}_{G_{Discr}} - i)$ motifs restants sont parcourus pour sélectionner le motif avec le meilleur score.

4.7 Expérimentations

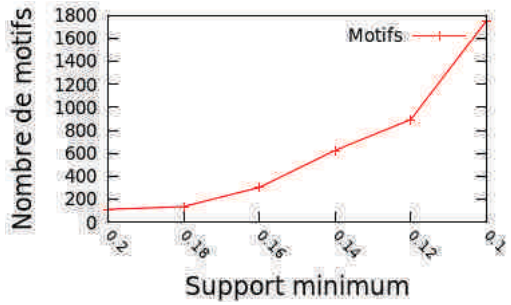
Pour tester les performances de cette approche, nous l'avons comparée avec une méthode alternative basée sur la notion de clustering, ici l'algorithme des k-médoïdes [KR09]. Proche de la méthode des k-moyennes, cet algorithme de classification non supervisée consiste à diviser les observations (séquences) en k partitions (clusters) où chaque observation est rattachée à un médoïde. Un médoïde représente l'observation la plus centrale d'une partition.

Ainsi, pour obtenir un résultat proche de notre méthode, nous appliquons une approche du type k-médoïdes en calculant k clusters, où chaque cluster est un en-

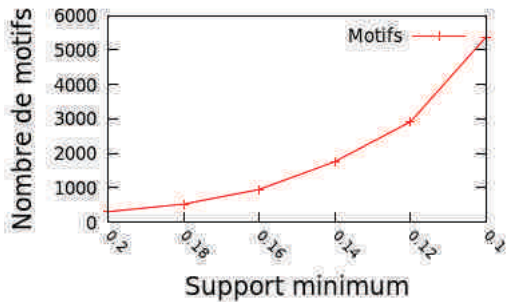
semble de motifs proches par la distance de *Hamming* entre leurs codes binaires. Une fois les k clusters générés, le score donné par l'équation 4.5 est calculé pour chaque motif d'un cluster afin de sélectionner le motif qui maximise ce score. Au final, cela revient à sélectionner k motifs qui maximisent fréquence et discrimination, à partir de k clusters qui permettent de minimiser la redondance.

Comme pour notre exemple illustratif, les expérimentations sont effectuées sur le jeu de données *Fresqueau*. Pour évaluer les performances de notre approche, les expérimentations sont effectuées sur la classe jaune des trois sous-jeux de données biologiques qui sont *IBGN*, *IBD* et *IPR*. Comme nous l'avons dit dans la section 4.4, plus de détails sur ces jeux de données sont fournis dans le chapitre 6. Pour ces expérimentations, le nombre k de motifs sélectionnés est fixé à 15. Les seuils de fréquence minimums utilisés dans les trois jeux de données vont de 0,2 à 0,1.

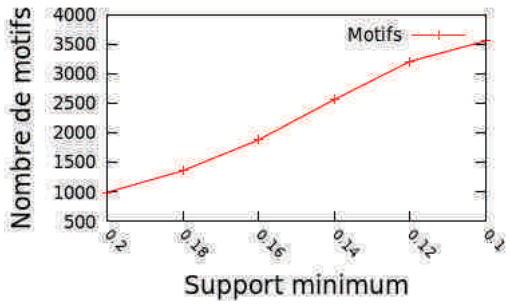
Avant de présenter les performances de notre approche et celle de l'approche par k-médoïdes, nous donnons dans les figures 4.4a, 4.4b et 4.4c le nombre de motifs partiellement ordonnés clos discriminants extraits dans chaque classe jaune des sous-jeux de données.



(a) Classe jaune *IBGN*



(b) Classe jaune *IBD*

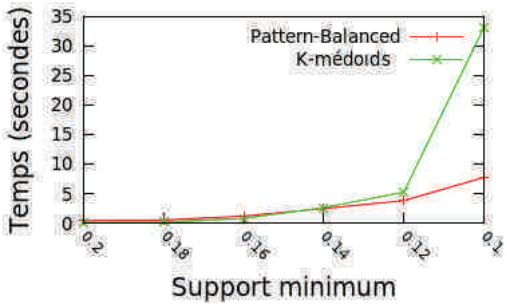


(c) Classe jaune *IPR*

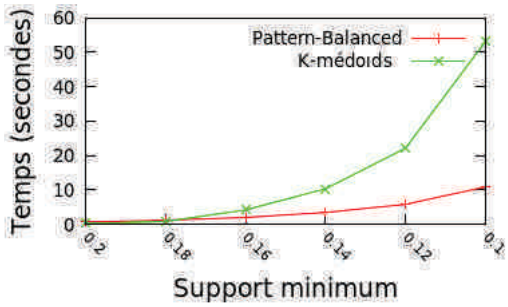
FIGURE 4.4 – Nombre de motifs discriminants extraits de la classe jaune des sous-jeux de données *IBGN*, *IBD* et *IPR*

La comparaison des temps de performance entre *Pattern-Balanced* et l'approche

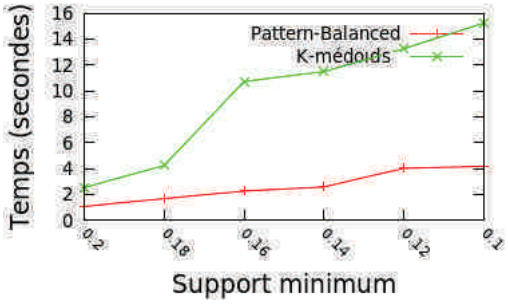
par k-médoïdes est donnée par les figures 4.5a, 4.5b et 4.5c.



(a) Classe jaune *IBGN*



(b) Classe jaune *IBD*



(c) Classe jaune *IPR*

FIGURE 4.5 – Comparaison entre *Pattern-Balanced* et l'approche par k-médoïdes sur la classe jaune des sous-jeux de données *IBGN*, *IBD* et *IPR*

Il est intéressant d'observer qu'une approche par k-médoïdes est plus rapide que la notre lorsque le nombre de motifs est de quelques centaines. Ensuite, plus ce nombre augmente, plus notre approche devient efficace. Par exemple prenons la classe jaune du sous-jeu de données *IBD* (figure 4.5b). Au seuil de fréquence minimum 0,2, l'approche par k-médoïdes s'exécute en 0,277 secondes et *Pattern-Balanced* en 0,65 secondes (304 motifs). Au seuil de fréquence minimum 0,12, l'approche par k-médoïdes s'exécute en 22,101 secondes et *Pattern-Balanced* en 5,789 secondes (2917 motifs). Maintenant, si nous prenons le jeu de données *IBGN* (figure 4.5a), nous observons que les deux approches ont un temps de calcul très proche au seuil de fréquence minimum de 0,14, qui correspond à 624 motifs extraits. Pour ces données, nous pouvons évaluer à 600/700 motifs la limite à partir de laquelle il est plus efficace d'utiliser *Pattern-Balanced* qu'une approche de clustering par k-médoïdes.

4.8 Synthèse

Dans ce chapitre, nous avons présenté *Pattern-Balanced*, un algorithme capable de sélectionner k motifs pour les utilisateurs. Cette méthode s'applique dans un contexte de bases de données partitionnées en classes. Les critères d'intérêt sont la fréquence, la discrimination et la non-redondance des motifs.

L'approche est un algorithme incrémental qui calcule, pour un ensemble de motifs, un score qui permet de sélectionner le motif représentant le plus d'intérêt. Ce score est une pondération entre les scores normalisés de chaque critère. L'exemple illustré montre d'une part que les motifs sélectionnés sont répartis dans l'espace des motifs extraits (non-redondance) et, d'autre part, qu'ils sont fréquents et/ou discriminants.

Les expérimentations montrent qu'utiliser l'approche *Pattern-Balanced* sur plusieurs milliers de motifs est plus efficace qu'une approche alternative basée sur le calcul de clusters par k-médoïdes.

Bien que la sélection de k motifs soit une manière efficace de filtrer l'information extraite pour en restituer un sous-ensemble d'intérêt, nous nous sommes également penchés sur une alternative à la fouille de motifs discriminants : l'extraction de consensus pour résumer un ensemble de motifs séquentiels discriminants. Cette méthode fait l'objet du prochain chapitre.

CHAPITRE 5

Extraction de consensus partiellement ordonnés discriminants

Contents

5.1 Introduction

Dans le chapitre précédent, nous avons étudié le problème de la sélection de k motifs d'intérêt. En effet, le nombre de motifs extraits peut être considérable et nécessite bien souvent la mise en place de méthodes de sélection qui permettent la restitution d'un sous-ensemble de motifs d'une taille assez petite pour être exploité par un humain.

Néanmoins, un utilisateur peut également être intéressé par un résumé approximatif des données, permettant la restitution d'un volume d'informations plus important qu'un seul motif. Un résumé restitue un point de vue global sur la base et ne nécessite pas l'analyse de chaque motif un par un. Dans ce chapitre, nous présentons une première approche qui extrait un résumé de l'information contenue dans une base de données de séquences. Nous appelons ce résumé un consensus partiellement ordonné. Comme pour la fouille de motifs partiellement ordonnés clos, nous l'avons adapté au cas des bases de données composées de classes pour l'extraction de consensus partiellement ordonnés discriminants.

Nous présentons également une seconde approche. Il existe un autre domaine où la recherche d'un résumé de l'information est pertinente, celui du *clustering* ou classification non supervisée. En effet, lorsque les classes d'un jeu de données ne sont pas connues, un *clustering* permet alors de générer un ensemble de classes selon la dissimilarité entre les éléments d'une base. Générer, ensuite, un consensus donne une idée du contenu de chaque *cluster* sans nécessiter l'analyse des séquences qui le composent. Cependant, les algorithmes de *clustering* requièrent l'utilisation d'une mesure de distance ou une dissimilarité entre les éléments de la base de données [KR09]. Pour calculer la dissimilarité entre deux séquences d'itemsets, en particulier dans le cadre de données environnementales, nous avons adapté l'approche *Dynamic Time Warping* [SC78] aux séquences d'itemsets et proposé la mesure de dissimilarité *Dynamic Sequence Warping*.

Les motivations de ce chapitre sont présentées dans la section 5.2. L'extraction d'un consensus partiellement ordonné dans une base de séquences est ensuite présentée dans la section 5.3 et son adaptation aux bases de données composées de classes dans section 5.4. Pour l'application de la fouille de consensus dans le cadre d'approches par *clustering*, une méthode de calcul de la dissimilarité entre séquences d'itemsets est introduite dans la section 5.5. Pour finir, nous faisons la synthèse de ce chapitre dans la section 5.6.

5.2 Motivations

À notre connaissance, dans la littérature, il existe peu de travaux sur la recherche de consensus dans les bases de données de séquences d'itemsets. La principale approche est *ApproxMap* [KPWD03], pour *APPROXimate Multiple Alignment Pattern Mining*. Cette méthode fonctionne en deux étapes : (1) les séquences de la base de données sont regroupées en clusters sur la base de leur similarité et (2) un motif consensus est extrait pour chaque *cluster* à partir d'un alignement multiple de ses séquences. L'alignement est résumé par une séquence dont les items sont pondérés. Ensuite, un consensus est généré à partir des items dont le poids est supérieur à un seuil de poids minimum. Même si cette approche permet d'extraire n (paramètre du *clustering*) consensus à partir d'une base de données de séquences, elle souffre de plusieurs inconvénients : (1) l'alignement multiple des séquences est une heuristique incrémentale qui est sensible à l'ordre dans lequel sont traitées les séquences et (2) le motif consensus extrait n'est pas exact, dans le sens où la séquence qui représente ce consensus n'est pas nécessairement inclue (sous-séquence) dans les données mais représente une approximation de l'alignement.

Comme nous l'avons vu, les motifs partiellement ordonnés clos permettent de capturer l'ordre partiel entre les éléments d'un ensemble de séquences. Ils donnent une information exacte sur cet ensemble car tous les chemins (motifs séquentiels) contenus dans un motif partiellement ordonné sont supportés par chaque séquence de l'ensemble. Nous nous sommes alors intéressés à la problématique suivante : *Peut-on utiliser la notion de séquence partiellement ordonnée (voir définition 14 du chapitre 2) comme consensus pour restituer un résumé de l'information ?*

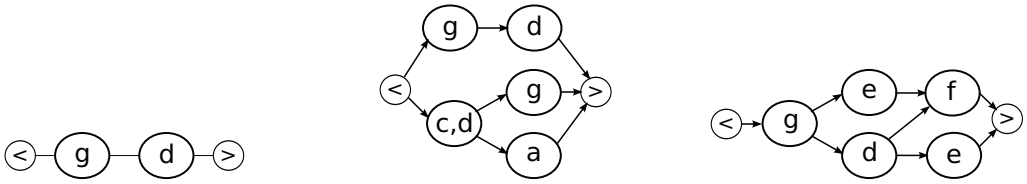
Contrairement à l'existant, notre approche doit respecter les contraintes suivantes : (1) chaque chemin dans le consensus représente au moins une séquence de la base de données (sous-séquence) et (2) le résultat n'est pas sensible à l'ordre dans lequel la base de données est traitée.

5.3 Consensus de motifs séquentiels

Avant de présenter notre approche et pour faciliter la compréhension de la méthode proposée, nous rappelons dans les figures 5.1, 5.1 et 5.2 la base de données exemple, les motifs partiellement ordonnés clos ainsi que l'arbre de recherche présentés dans les chapitres 2 et 3.

ID	Séquence
S_1	$\langle(c,d)(a)(g)(d)\rangle$
S_2	$\langle(g)(c,d,e)(f)(a,e,g)\rangle$
S_3	$\langle(g)(d)(e)(f)\rangle$

TABLE 5.1 – Exemple de base de données de séquences



(a) M_{G_1} sur $\{S_1, S_2, S_3\}$ (b) M_{G_2} sur $\{S_1, S_2\}$ (c) M_{G_3} sur $\{S_2, S_3\}$

FIGURE 5.1 – Motifs partiellement ordonnés clos extraits à partir de la base de données du tableau 5.1 avec un support minimum $\theta = 2$

5.3.1 Adaptation des motifs partiellement ordonnés

La méthode de consensus proposée se base sur les motifs partiellement ordonnés. Comme dit précédemment, un consensus est un résumé d'une base de données de séquences. Comme nous l'avons vu dans le chapitre 3, un motif partiellement ordonné clos peut être extrait sur un ensemble de séquences $\mathcal{S}$, où chaque chemin de ce motif est un motif séquentiel nécessairement inclus dans chaque séquence $S \in \mathcal{S}$.

Pour résumer une base de données de séquences, une approche naïve est d'extraire le motif partiellement ordonné clos qui couvre cette base, i.e. possible avec une fréquence minimale θ de 100% par exemple. Cependant, extraire un tel motif sur toute une base n'est pas pertinent car le motif représente une information trop générale. De plus, il est fort probable que ce motif soit vide s'il n'existe pas d'item commun à toutes les séquences. Par exemple, prenons le motif M_{G_1} de la figure 5.1a. Celui-ci couvre l'ensemble de séquences $\{S_1, S_2, S_3\}$. Il est seulement composé de deux items. Il ne donne pas d'information sur les sous-ensembles de séquences de la base.

À l'inverse, sélectionner une fréquence minimale inférieure à 100% permet d'extraire des motifs partiellement ordonnés clos sur des sous-ensembles de séquences. Cependant, on s'éloigne alors de la notion de consensus qui nécessite l'extraction d'une seule séquence partiellement ordonnée.

En effet, tout l'intérêt de la méthode qui est proposée dans ce chapitre est

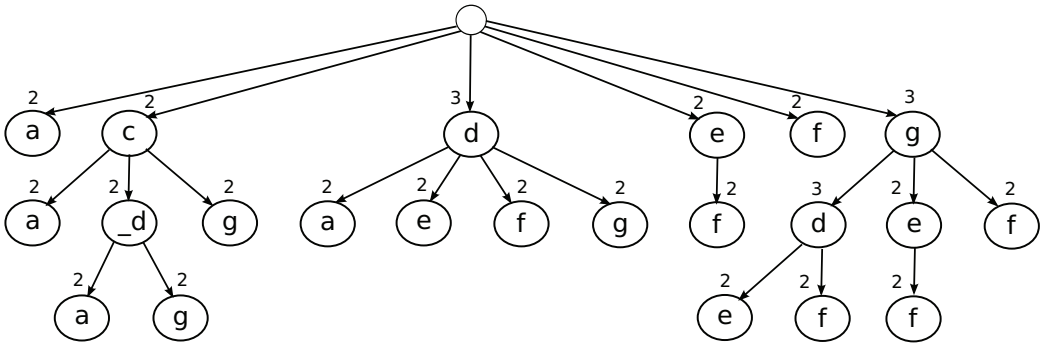


FIGURE 5.2 – Arbre de recherche des motifs séquentiels généré à partir de la base de données de séquences du tableau 5.1 avec un support minimum de $\theta = 2$

de pouvoir extraire une seule séquence partiellement ordonnée sur un ensemble de séquences $\mathcal{S}$, qui est aussi composée d'informations spécifiques à des sous-ensembles de séquences $\mathcal{S}' \subset \mathcal{S}$.

5.3.2 Extraire et fusionner l'arbre des motifs séquentiels fréquents

Lors de l'exploration de l'espace de recherche, une méthode de type *PrefixSpan*, ou bien *OrderSpan* (voir chapitre 3), génère de nouveaux motifs au fur et à mesure de la découverte de nouvelles extensions des motifs existants. Nous souhaitons adapter ce processus à la fouille de consensus. L'idée est d'explorer l'espace de recherche de la même manière que dans une approche classique de fouille de motifs. La différence est que nous ne générons pas de motif, mais seulement l'arbre représentant l'espace de recherche. Par exemple, extraire l'arbre des motifs séquentiels fréquents sur la base de données du tableau 5.1 avec un support minimum $\theta = 2$, génère le résultat présenté par la figure 5.2.

Comme présenté dans le chapitre 3, *OrderSpan* utilise la notion de sous-arbre de l'espace de recherche pour extraire les motifs partiellement ordonnés clos, où chaque arbre contient l'ensemble des motifs séquentiels qui couvrent un sous-ensemble de séquences de la base de données. Ici, l'idée est de fusionner non pas un sous-arbre de l'espace de recherche pour en faire un motif, mais de fusionner l'ensemble de l'espace de recherche selon une fréquence minimale θ pour générer un consensus partiellement ordonné.

Pour illustrer l'approche, reprenons l'arbre des motifs fréquents de la figure 5.2. Pour rappel celui-ci contient l'ensemble des motifs séquentiels supportés par les

séquences $\{S_1, S_2, S_3\}$ de la base de données du tableau 5.1 avec un support minimum $\theta = 2$. Si nous appliquons directement l'opération *MergingSuffixTree* (présentée dans le chapitre 3) sur l'arbre de recherche au complet, nous obtenons le consensus partiellement ordonné représenté par la figure 5.3.

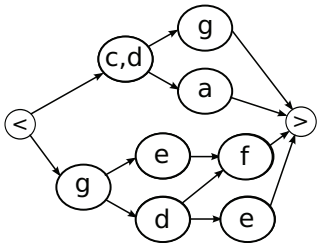


FIGURE 5.3 – Consensus partiellement ordonné généré à partir de l'arbre des motifs fréquents de la figure 5.2

Nous voyons que ce consensus partiellement ordonné contient l'ensemble des motifs séquentiels qui sont supportés par la base du tableau 5.1 avec $\theta = 2$. Plus encore, si nous prenons les motifs partiellement ordonnés clos M_{G_1}, M_{G_2} et M_{G_3}, nous voyons qu'ils sont des sous-séquences partiellement ordonnées (voir définition 17 du chapitre 2) de ce consensus. Par exemple, les trois motifs séquentiels $\langle(g)(e)(f)\rangle$, $\langle(g)(d)(f)\rangle$ et $\langle(g)(d)(e)\rangle$ qui composent le motif M_{G_3} sont inclus dans au moins un chemin du consensus de la figure 5.3. Ainsi, la définition 26 décrit la notion de consensus partiellement ordonné.

Définition 26 ***(Consensus partiellement ordonné)***
Soient G une séquence partiellement ordonnée, $\mathcal{P}_G$ l'ensemble des chemins dans G, $\mathcal{D}$ une base de données de séquences et θ une valeur entière tel que $0 < \theta \leq |\mathcal{D}|$. G est un ***consensus partiellement ordonné*** *si $\forall P_G \in \mathcal{P}_G$, $Support_{\mathcal{D}}(P_G) \geq \theta$ et il n'existe pas de séquence partiellement ordonnée G', avec $\mathcal{P}'_G$ l'ensemble des chemins dans G' et $\forall P'_G \in \mathcal{P}'_G, Support_{\mathcal{D}}(P'_G) \geq \theta$, tel que $G \preceq_g G'$. On appelle θ le support minimum.*

La contrepartie d'un consensus partiellement ordonné est la perte de l'information exacte sur la valeur du support de chaque chemin (motif séquentiel). Nous savons par contre que chaque chemin du consensus partiellement ordonné est supporté par un nombre de séquences de la base supérieur ou égal à θ.

Considérons maintenant le cas d'une base de données partitionnée en classes. Dans le chapitre précédent (chapitre 4), nous avons présenté *DiscriminantOrderSpan*, une adaptation d'*OrderSpan* à la fouille de motifs partiellement ordonnés clos

discriminants. Cependant il n'est pas possible d'appliquer un tel post-traitement pour l'extraction de consensus car nous n'extrayons qu'une seule structure de données. Nous allons voir, dans la prochaine section, comment adapter cette approche à la fouille de consensus discriminants, i.e. nous construisons un consensus où seuls les motifs séquentiels discriminants sont résumés.

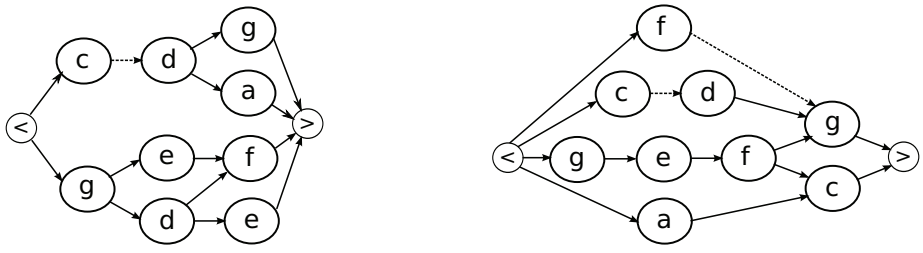
5.4 Consensus discriminant

Tout d'abord, pour illustrer l'importance d'adapter la notion de discrimination aux consensus partiellement ordonnés, prenons la base de données de la table 5.2 composée de deux classes, $\mathcal{C}_1$ et $\mathcal{C}_2$.

Classe $\mathcal{C}_1$		**Classe $\mathcal{C}_2$**	
ID	**Séquence**	**ID**	**Séquence**
S_{11}	$\langle(c,d)(a)(g)(d)\rangle$	S_{21}	$\langle(g)(e)(f)(a,g)(c)\rangle$
S_{12}	$\langle(g)(c,d,e)(f)(a,e,g)\rangle$	S_{22}	$\langle(a)(c)(a)(c,d)(a)(f,g)\rangle$
S_{13}	$\langle(g)(d)(e)(f)\rangle$	S_{23}	$\langle(f,g)(e)(f)(c,d)(g)\rangle$

TABLE 5.2 – Exemple de base de données de séquences composée de 2 classes

À partir de chaque classe de la base de données, il est possible d'extraire un consensus partiellement ordonné. Les figures 5.4a et 5.4b donnent les consensus pour les deux classes $\mathcal{C}_1$ et $\mathcal{C}_2$ avec un support minimum $\theta = 2$.



(a) (b)

FIGURE 5.4 – Consensus pour (a) la classe $\mathcal{C}_1$ et pour (b) la classe $\mathcal{C}_2$ avec $\theta = 2$

Si nous comparons ces deux classes, nous pouvons observer deux choses :

— certains motifs séquentiels sont autant fréquents dans les deux classes, comme les motifs $\langle(c,d)(g)\rangle$ et $\langle(g)(e)(f)\rangle$;

— d'autres motifs séquentiels sont au contraire plus fréquents dans une des deux classes, comme le motif $\langle (g)(d)(f) \rangle$ qui est plus fréquent dans $\mathcal{C}_1$ ou bien le motif $\langle (a)(c) \rangle$ qui est plus fréquent dans $\mathcal{C}_2$ (absent dans $\mathcal{C}_1$).

Ce sont ces derniers motifs séquentiels qui nous intéressent et que nous souhaitons résumer. Cependant, comme nous l'avons dit précédemment, il n'est pas possible d'effectuer un post-traitement. En effet, nous perdons le support des motifs séquentiels qui sont résumés dans un consensus, il est nécessaire d'effectuer la sélection des motifs séquentiels discriminants lors de la génération de l'arbre des motifs séquentiels. Prenons comme exemple l'arbre des motifs séquentiels de la classe $\mathcal{C}_1$ donné par la figure 5.5.

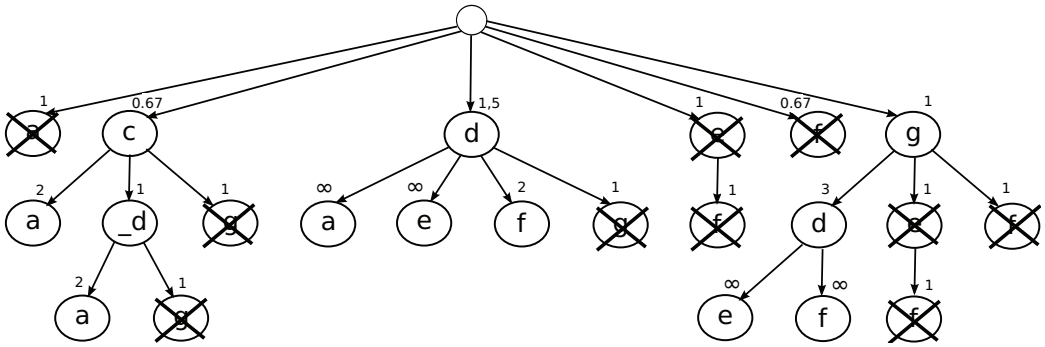


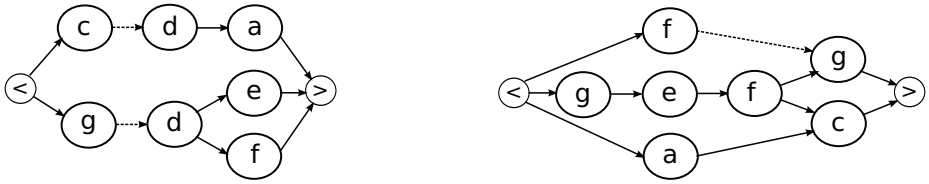
FIGURE 5.5 – Arbre de recherche avec élagage des motifs séquentiels qui ne sont pas discriminants

Contrairement à l'arbre de la figure 5.2, chaque motif séquentiel (sommet) est caractérisé par sa valeur de *growth rate* dans la classe $\mathcal{C}_1$. Pour rappel, le *growth rate* (voir définition 20 du chapitre 2) est un rapport de fréquence d'un motif entre deux classes, où une valeur supérieure à 1 dans une classe signifie que le motif est plus fréquent dans celle-ci que dans l'autre. Dans ce chapitre, nous utilisons le *growth rate* car l'exemple est composé de deux classes. Dans le cas d'une base de données de n classes avec $n > 2$, nous utilisons le *generalized growth rate* (voir définition 20 du chapitre 4 et l'application dans le chapitre 7).

Afin de générer un consensus discriminant depuis $\mathcal{C}_1$, l'idée consiste à élaguer l'arbre de tous les motifs séquentiels qui ne sont pas discriminants. Par exemple, les motifs séquentiels $\langle (g)(e)(f) \rangle$ et $\langle (g)(f) \rangle$ sont élagués dans l'arbre extrait de $\mathcal{C}_1$ car leur valeur de *growth rate* est égale à 1. Ils sont donc aussi fréquents dans $\mathcal{C}_1$ que dans $\mathcal{C}_2$. Par contre, nous observons que le motif séquentiel $\langle (g) \rangle$ n'est pas élagué, pourtant sa valeur de *growth rate* est également de 1. Il n'est pas élagué car il possède des sommets fils dont la valeur de *growth rate* est supérieure à 1. Par exemple, le

motif séquentiel $\langle (g)(d) \rangle$ a un *growth rate* de 3 et $\langle (g)(d)(e) \rangle$ a un *growth rate* infini.

Une fois tous les motifs séquentiels non-discriminants élagués, l'arbre des motifs séquentiels est transformé en séquence partiellement ordonnée et fusionné avec l'opération *MergingSuffixTree*. Les figures 5.6a et 5.6b donnent les consensus discriminants pour les classes $\mathcal{C}_1$ et $\mathcal{C}_2$. Nous observons dans les deux cas que seuls les motifs séquentiels discriminants sont résumés dans les consensus de chaque classe.



(a) (b)

FIGURE 5.6 – Consensus discriminants pour (a) la classe $\mathcal{C}_1$ et pour (b) la classe $\mathcal{C}_2$ avec $\theta = 2$

Cependant, afin d'extraire des consensus discriminants dont les motifs séquentiels résumés ont un *growth rate* plus élevé que 1, nous étendons alors la notion de consensus discriminant. Dans [DL99], l'auteur introduit un seuil de discrimination ρ qui permet d'extraire seulement les motifs dont la valeur de *growth rate* est supérieure ou égale à ρ. Nous adaptons alors ce seuil de discrimination ρ aux consensus discriminants. Nous donnons tout d'abord la définition de motif ρ-discriminant [DL99] (définition 27).

Définition 27 ***(Motif ρ-discriminant)***
*Soient M un motif, deux classes $\mathcal{C}$ et $\mathcal{C}'$ et ρ une valeur entière tel que $\rho > 1$. M est dit ρ-**discriminant** dans la classe $\mathcal{C}$ par rapport à la classe $\mathcal{C}'$ si $GR(M, \mathcal{C}, \mathcal{C}') \geq \rho$. On appelle ρ le seuil de **discrimination minimum**.*

Ainsi, à partir d'un seuil minimum de discrimination ρ, il est possible d'élaguer de l'arbre des motifs séquentiels fréquents tous ceux dont la valeur de *growth rate* est strictement inférieure à ρ. Ce qui nous permet de définir la notion de consensus ρ-discriminant (définition 28).

Définition 28 ***(Consensus ρ-discriminant)***
*Soient G_C un consensus, $\mathcal{P}_{G_C}$ l'ensemble des chemins dans G_C, deux classes $\mathcal{C}$ et $\mathcal{C}'$ et ρ une valeur entière tel que $\rho > 1$. G_C est dit ρ-**discriminant** dans la classe $\mathcal{C}$ par rapport à la classe $\mathcal{C}'$ si $\forall P_{G_C} \in \mathcal{P}_{G_C}$, $GR(P_{G_C}, \mathcal{C}, \mathcal{C}') \geq \rho$. On appelle ρ le seuil de **discrimination minimum**.*

À partir de cette définition, il est par exemple possible de générer les consensus partiellement ordonnés 3-discriminants pour chaque classe. Ils sont donnés par les figures 5.7a et 5.7b. Comme nous le voyons, d'autres motifs séquentiels sont élagués comme le motif séquentiel $\langle (c,d)(a) \rangle$ pour la classe $\mathcal{C}_1$ puisque celui-ci à une valeur de *growth rate* de 2 dans $\mathcal{C}_1$. Seuls les motifs séquentiels avec une valeur de *generalized growth rate* supérieure ou égale à $\rho = 3$ sont conservés dans l'arbre des motifs séquentiels.

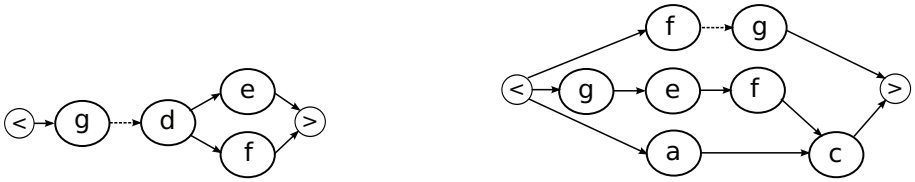


FIGURE 5.7 – Consensus 3-discriminants pour (a) la classe $\mathcal{C}_1$ et pour (b) la classe $\mathcal{C}_2$ avec $\theta = 2$

Dans cette section, nous avons présenté l'adaptation des séquences partiellement ordonnées pour la fouille de consensus discriminants. Dans la prochaine section nous présentons une méthode de calcul de dissimilarité entre deux séquences d'itemsets. Le but est de pouvoir adapter la fouille de consensus discriminant au cas de classes générées par un algorithme de classification non-supervisée, utilisant une mesure de dissimilarité.

5.5 Extraction de consensus pour résumer des clusters

Comme nous l'avons dit dans l'introduction de ce chapitre, l'extraction de consensus discriminants est pertinente dans le cadre de méthodes de *clustering*, lorsque les classes ne sont pas connues. En effet, si la base de données contient un nombre important de séquences, ou si le nombre de clusters généré est important, il n'est jamais simple de déterminer ce qui fait la particularité de chaque *cluster* : (1) *quelle est l'information que partagent en commun les séquences d'un même cluster ?* (2) *quelle est l'information commune à l'ensemble de séquences d'un cluster et qui n'est pas présente dans les autres clusters ?*

La notion de consensus partiellement ordonné discriminant est tout à fait adaptée

comme réponse à ces deux problématiques.

Prenons par exemple le cas illustratif (figure 5.8) d'une base de séquences où plusieurs clusters ont été identifiés.

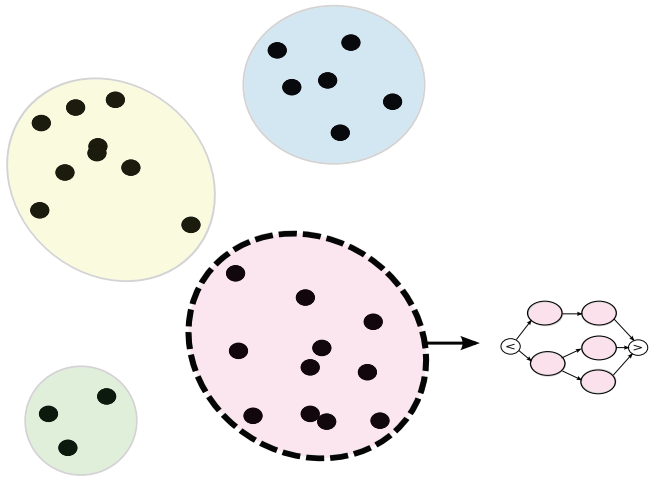


FIGURE 5.8 – Exemple illustratif du *clustering* d'une base de données de séquences

Dans ce schéma, quatre clusters ont été générés et un consensus discriminant a été extrait du *cluster* en bas à droite. Ce consensus représente les motifs séquentiels qui sont discriminants dans ce cluster, i.e. moins fréquents dans les autres clusters selon un seuil de support minimum θ et un seuil de discrimination minimum ρ.

Cependant, les méthodes de *clustering* sur les séquences nécessitent l'utilisation d'une distance ou mesure de dissimilarité entre ces séquences. Cette question a fait l'objet de nombreux travaux. Une des méthodes les plus connues est sûrement le calcul de la plus longue sous-séquence commune ($LCSS$) [BHR00] ou bien encore l'utilisation de la distance d'édition [Nav01]. Grâce à la définition de coûts liés à l'insertion, la suppression ou le remplacement, cette dernière est particulièrement efficace pour le calcul de distances entre des séquences de protéines ou sur les chaînes de caractères. Dans [ERC+13], les auteurs ont également proposé une mesure de similarité basée sur le nombre total de sous-séquences communes entre deux séquences.

Dans ce travail, nous nous sommes intéressés à une méthode de dissimilarité couramment utilisée dans le domaine des séries temporelles : *Dynamic Time Warping* [SC71]. Cette mesure s'appuie sur l'alignement optimal entre deux séries temporelles. Cette approche a démontré son efficacité en reconnaissance vocale [SC71, SC78] et de signatures [PP90]. Elle permet de prendre en compte la similarité entre deux séries temporelles dont la variabilité est différente, i.e. des similarités entre deux séries temporelles peuvent être détectées même si, dans une des séries temporelles,

le phénomène mesuré a varié plus rapidement que dans l'autre. Prenons l'exemple d'enregistrements de voix. Deux personnes n'ont pas la même vitesse de locution. *Dynamic Time Warping* peut déterminer que deux enregistrements sont similaires bien que les signaux aient une durée différente.

Dans le cas des séquences d'itemsets, il peut être intéressant de considérer cette différence de durée. Prenons par exemple $S_1 = \langle (b)(a)(c) \rangle$ et $S_2 = \langle (b)(b)(a)(a)(c) \rangle$ deux séquences. Elles sont toutes les deux de tailles différentes mais nous observons la même succession d'items : b est suivi de a qui est suivi de c. Même si dans S_2 les items a et c apparaissent chacun deux fois, successivement, nous observons le même enchaînement d'items que dans S_1. L'objectif de notre méthode, appelée *Dynamic Sequence Warping* (DSW) est alors d'obtenir une valeur de dissimilarité de 0 pour ces deux séquences, que nous considérons comme identiques. Concrètement, si nous considérons le cas de données temporelles environnementales, il est fréquent qu'un même phénomène ait lieu à deux endroits différents avec une durée différente, l'un instantané et l'autre sur une durée de plusieurs jours. Ces phénomènes peuvent avoir la même durée, mais l'hétérogénéité des mesures environnementales effectuées peut entraîner une absence de données. Nous souhaitons alors considérer ces séquences d'événements comme équivalentes, bien que différentes à première vue. Notre choix d'adapter *Dynamic Time Warping* aux séquences d'itemsets est également justifié par sa facilité d'implémentation et aux performances obtenues sur les données hydrobiologiques du projet Fresqueau.

Dynamic Time Warping calcule l'alignement optimal entre deux séries temporelles au travers d'une matrice de distance entre chaque paire de points des deux séries temporelles. Les éléments dans les séries temporelles sont des valeurs quantitatives. Une méthode simple pour calculer la distance entre deux valeurs numériques x et y est de calculer la valeur absolue de leur différence $|x - y|$.

Dans les séquences d'itemsets, les éléments sont des ensembles de variables discrètes. Ainsi, pour étendre *Dynamic Time Warping* aux séquences d'itemsets, nous introduisons tout d'abord une mesure de dissimilarité entre itemsets. Nous utilisons une dissimilarité calculée à partir de l'indice de similarité de *Jaccard* [Jac12], qui s'appuie sur le nombre d'éléments partagés ou distincts entre deux ensembles. La dissimilarité de *Jaccard* entre deux itemsets IS et IS', notée $Jaccard_{Dissim}(IS, IS')$, est donnée par la formule suivante.

Définition 29 ***(Dissimilarité de Jaccard)***
Soient IS et IS' deux itemsets, la dissimilarité de Jaccard est définie par :

$$\begin{aligned} Jaccard_{Dissim}(IS, IS') &= 1 - Jaccard_{Sim}(IS, IS') \\ &= 1 - \frac{|IS \cap IS'|}{|IS \cup IS'|} \end{aligned} \tag{5.1}$$

Par exemple, soit $IS_1 = \{a, d, e, f\}$ et $IS_2 = \{b, d, f\}$ deux itemsets. Le calcul de leur dissimilarité est donné par :

$$\begin{aligned} Jaccard_{Dissim}(IS_1, IS_2) &= 1 - Jaccard_{Sim}(IS_1, IS_2) \\ &= 1 - \frac{|\{a, d, e, f\} \cap \{b, d, f\}|}{|\{a, d, e, f\} \cup \{b, d, f\}|} \\ &= 1 - \frac{|\{d, f\}|}{|\{a, b, d, e, f\}|} = 1 - \frac{2}{5} = 0,6 \end{aligned} \tag{5.2}$$

Maintenant que nous avons défini une mesure de dissimilarité entre itemsets, nous définissons dans la définition 30 une mesure de dissimilarité entre séquences d'itemsets. Celle-ci se base sur l'approche *Dynamic Time Warping* qui utilise la programmation dynamique. La programmation dynamique consiste à calculer la solution finale d'un problème en le divisant en sous-problèmes résolus localement [GMS04].

Définition 30 ***(Dynamic Sequence Warping)***
Soient $S = \langle IS_1 IS_2 \dots IS_m \rangle$ et $S' = \langle IS'_1 IS'_2 \dots IS'_n \rangle$ deux séquences d'itemsets. Leur dissimilarité est définie par :

$$DSW(S,S') = \gamma(m, n) \tag{5.3}$$

où $\gamma(1, 1) = Jaccard_{Dissim}(IS_1, IS'_1)$ et où $\gamma(i, j)$, avec $i > 1$ et $j > 1$, est la valeur du chemin optimal récursivement défini comme :

$$\gamma(i, j) = Jaccard_{Dissim}(IS_i, IS'_j) + \min \begin{cases} \gamma(\ i-1\ ,\ j-1\ ), \\ \gamma(\quad i \quad ,\ j-1\ ), \\ \gamma(\ i-1\ ,\quad j \quad) \end{cases} \tag{5.4}$$

Nous illustrons ce calcul dans la figure 5.9 avec les séquences $S_1 = \langle (abce)(fg)(fgi)(bc)(gk) \rangle$ et $S_2 = \langle (bc)(abcd)(fghi)(fgk) \rangle$. Pour clarifier l'exemple, nous séparons le processus en deux matrices : (1) chaque cellule de la matrice de gauche

représente la valeur de la dissimilarité de *Jaccard* entre chaque paire d'itemsets des deux séquences (figure 5.9a) et (2) étant donné cette matrice de distance entre itemsets, la matrice de droite sert à calculer la valeur du chemin qui minimise les valeurs de distance entre ces itemsets (figure 5.9b) en utilisant la définition 30 et l'équation 5.4. Ainsi, la valeur de distance retournée par *Dynamic Sequence Warping* entre S_1 et S_2 est de 2,83.

	abce	fg	fgi	bc	gk
bc	0,5	1	1	0	1
abcd	0,25	1	1	0,5	1
fghi	1	0,5	0,25	1	0,75
fgk	1	0,33	0,5	1	0,33

(a)

	abce	fg	fgi	bc	gk
bc	0,5	1,5	2,5	2,5	3,5
abcd	0,75	1,5	2,5	3	3,5
fghi	1,75	1,25	1,5	2,5	3,25
fgk	2,75	1,58	1,75	2,5	2,83

(b)

FIGURE 5.9 – *Dynamic Sequence Warping* sur les séquences S_1 et S_2

Cette mesure de dissimilarité a la même complexité que *Dynamic Time Warping*. Soient $S = \langle IS_1 IS_2 \dots IS_m \rangle$ et $S' = \langle IS'_1 IS'_2 \dots IS'_n \rangle$ deux séquences d'itemsets, la matrice du calcul du chemin optimal peut être obtenue en $\Theta(m \cdot n)$ grâce à la programmation dynamique. Cette complexité en $\Theta(m \cdot n)$ correspond au nombre de calculs de la dissimilarité de *Jaccard* entre deux itemsets. Soient deux itemsets IS et IS', la complexité du calcul de la dissimilarité de *Jaccard* entre IS et IS' dépend de l'implémentation. Par exemple, l'utilisation de tables de hachage permet d'effectuer l'intersection des ensembles en $\Theta(min(|IS|, |IS'|))$ et l'union en $\Theta(|IS| + |IS'|)$.

Dans cette section, nous avons présenté *Dynamic Sequence Warping*, une adaptation de *Dynamic Time Warping* pour les séquences d'itemsets. Elle est particulièrement adaptée au problème de données environnementales, où l'hétérogénéité des données et la complexité des processus mis en jeu nécessitent de définir une méthode qui tolère la variabilité temporelle.

5.6 Synthèse

Dans ce chapitre, nous avons présenté une approche qui permet de résumer l'information contenue dans une base de données composée de classes. Ce résumé

d'information est représenté par un consensus partiellement ordonné qui se base sur la notion de séquence partiellement ordonnée. Il peut ainsi être vu comme une alternative viable à la fouille de motifs partiellement ordonnés discriminants.

Nous avons également présenté l'intérêt d'utiliser la fouille de consensus discriminants dans le cas de méthodes de *clustering* lorsque les classes ne sont pas connues. Pour pouvoir appliquer de telles méthodes, nous avons présenté *Dynamic Sequence Warping*, une méthode qui permet de calculer une mesure de dissimilarité entre séquences d'itemsets. Celle-ci se base sur l'approche *Dynamic Time Warping* utilisée dans le domaine des séries temporelles et qui fait partie de la famille des distances d'édition.

Pour montrer tout l'intérêt des consensus discriminants et de leur application au *clustering* de séquences, nous présentons dans le chapitre 7 un cas d'application concret sur les données hydrobiologiques du projet Fresqueau.

CHAPITRE 6

Contexte applicatif : hydrobiologie

Contents

6.1 Introduction

Cette thèse est financée par l'ANR Fresqueau. Ce projet original qui vise à fournir aux hydrobiologistes de nouveaux outils d'analyse basés sur la fouille de données, qui ont pour but d'être utilisés comme de nouvelles approches complémentaires aux méthodes statistiques, couramment utilisées dans ce domaine. Toutes les méthodologies présentées dans les chapitres précédent sont maintenant appliquées aux données du projet. Ce chapitre présente le contexte hydrobiologique ainsi que les données associées. Les résultats obtenus avec les approches méthodologiques de cette thèse sont présentés dans le chapitre suivant.

L'identification des sources de pollution dans les écosystèmes aquatiques est actuellement un domaine de recherche très actif et reste une tâche difficile. De nombreux paramètres sont impliqués dans la détermination de la qualité des écosystèmes aquatiques. Ces paramètres sont liés à différents aspects tels que la biologie, la physico-chimie et l'hydromorphologie des cours d'eau. La Directive Cadre Européenne [Eur00] a mis en évidence l'importance de rechercher de nouveaux outils opérationnels pour aider dans l'analyse et l'interprétation d'informations hydrobiologiques complexes. Ces informations peuvent concerner la qualité de l'eau des rivières et leur fonctionnement. Par conséquent, il est important de proposer de nouvelles méthodes qui prennent en compte toute la richesse et la complexité du problème.

La mesure de ces différentes dimensions environnementales est effectuée par de nombreuses organisations comme les agences de l'eau [1], les DREAL [2] ou bien l'ONEMA [3], qui ont chacune des objectifs précis. Au fil du temps, le volume de données collectées par ces organisations est devenu conséquent avec des gigaoctets de données hétérogènes. Il est donc important d'imaginer, de concevoir et de mettre en place une grande base de données homogène pour récolter et intégrer toutes ces données. Par exemple, le projet SEEE [4] a été créé par l'ONEMA pour collecter et traiter au niveau national les données des agences de l'eau. Dans un autre objectif, le projet ANR [5] Fresqueau [6] a commencé en 2011. Ce projet a pour but de traiter deux enjeux spécifiques : (1) mettre en évidence des liens entre différentes métriques permettant de caractériser la qualité des cours d'eau et (2) relier les sources de

1. http ://www.lesagencesdeleau.fr/
2. http ://www.alsace.developpement-durable.gouv.fr/
3. http ://www.onema.fr/
4. http ://seee.eaufrance.fr/SeeeEval/index
5. Agence nationale de la recherche
6. http ://engees-fresqueau.unistra.fr/

pressions sur le milieu à la qualité physico-chimique et biologique des cours d'eau. Pour cela, il a été nécessaire de constituer une base de données spécifique à partir d'un ensemble de données relatives à la qualité de l'eau, l'hydrologie, les stations de mesures, etc., mais également des données permettant de caractériser l'environnement des cours d'eau. Principalement basé sur la norme SANDRE[7], ce projet a rapatrié différentes sources de données. Le résultat obtenu est une base de données spatio-temporelle composée de nombreuses catégories de données qui décrivent des stations de prélèvements. À l'heure actuelle, cette base de données rassemble des informations relatives aux bassins versants de la partie Nord-Est et Sud-Est de la France. 11 329 stations de prélèvements réparties sur un total de 161 100 km^2 sont concernées, ce qui représente 29.45% de la France métropolitaine. Ces bassins versants sont groupés en deux zones hydrographiques majeures qui sont Rhin-Meuse (Nord-Est) notée RM et Rhone Méditéranée Corse (Sud-Est) notée RMC. La carte de France dans la figure 6.1 illustre la zone géographique couverte par ces deux zones hydrographiques. La zone en gris foncé correspond à RM et la zone en noir correspond à RMC. Les lignes blanches correspondent à la délimitation entre les différents bassins versants. Ces bassins versants sont traversés par un réseau hydrographique où des stations de prélèvements sont réparties. La partie droite de la figure 6.1 donne un exemple de réseau hydrographique (ici hypothétique, pour l'exemple) où se trouvent deux stations de prélèvements.

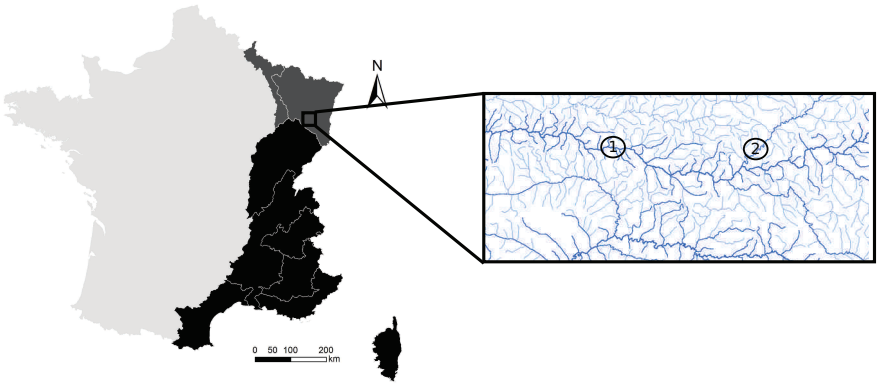


FIGURE 6.1 – Territoires concernés par notre étude et un exemple de deux stations de mesures sur le réseau hydrographique

Ces stations de prélèvements sont localisées dans l'espace par leurs coordonnées et leur rattachement à un cours d'eau, un bassin versant ou une commune. Diverses

7. http ://www.sandre.eaufrance.fr/

catégories de données sont associées aux stations qui sont relatives au climat, à la physico-chimie, à l'hydrobiologie, à l'occupation du sol, à l'hydromorphologie [8] et à l'hydrologie [9] (figure 6.2). Selon les catégories, les données peuvent avoir une dimension temporelle. Par exemple les données relatives à la physico-chimie et à l'hydrobiologie sont temporelles puisque pour chaque station, ces deux catégories sont renseignées par un ensemble de prélèvements effectués au cours du temps (généralement chaque année pour la biologie et tous les deux mois pour la physico-chimie). Ce sont ces deux dimensions temporelles qui vont nous intéresser plus particulièrement par la suite. À l'inverse, l'occupation du sol est par exemple une donnée statique que nous ne traitons pas dans le cadre de ce travail.

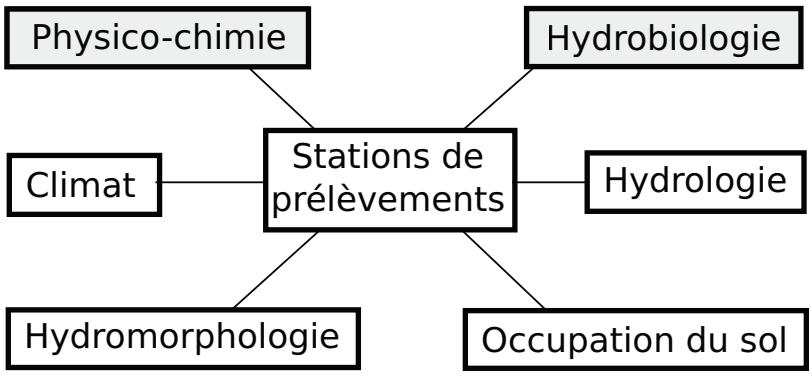


FIGURE 6.2 – Catégories de données stockées dans la base de données Fresqueau

Les méthodes développées dans cette thèse ont pour but de considérer l'aspect temporel des données. Dans le cadre du projet Fresqueau, nous pouvons les appliquer aux données de physico-chimie et d'hydrobiologie. Les hydrobiologistes sont intéressés par le lien qui existe entre ces deux catégories de données et qui reste mal connu bien que l'impact de certains éléments chimiques dans l'eau (pollution) sur la biodiversité des cours d'eau ait déjà été démontré [ZTZL96, TCM+02]. À notre connaissance, il existe peu de méthodes qui prennent en compte l'aspect temporel du problème. Une approche de fouille de données par motif temporel a l'avantage de prendre en compte l'ensemble des variables sans a priori et est adaptée à de telles problématiques. Ainsi, nos méthodes vont être utilisées pour répondre au problème suivant :

8. Science qui concerne l'étude de la géomorphologie due à l'hydrologie
9. Science qui concerne l'étude du cycle de l'eau

Peut-on trouver des liens dans le temps entre des ensembles de valeurs de paramètres physico-chimiques et les valeurs de paramètres biologiques ?

Afin d'appliquer toutes les méthodologies développées, il est nécessaire de préparer le jeu de données :

— Les données brutes sont numériques, or les méthodes par motifs ont besoin de données discrétisées ;

— Des séquences temporelles doivent être construites à partir des données brutes.

Dans ce chapitre, nous détaillons dans un premier temps les données physico-chimiques et biologiques (6.2.1). Ensuite, la section 6.3 présente un état de l'art sur les méthodes de fouille de données appliquées à l'hydrobiologie. Enfin, dans la section 6.4, nous décrivons les différentes étapes nécessaires à la transformation du jeu de données brut en un jeu de données utilisable par nos méthodes.

6.2 Description des données

6.2.1 Données biologiques

Ces données concernent la faune ainsi que la flore vivant dans les rivières. D'un point de hydrobiologique, le terme taxon est employé pour considérer l'ensemble des espèces vivantes. Ce terme peut par exemple faire référence à la famille, la sous-famille ou l'espèce d'un individu donné. Ces taxons sont regroupés en différents compartiments biologiques, par exemple les macrophytes et les diatomées sont représentatifs de la flore et les poissons, les macroinvertébrés et les oligochètes sont représentatifs de la faune. Ces cinq compartiments biologiques ont été sélectionnés et analysés dans le but de mettre en place des indices (notes) biologiques. Il est considéré que les indices biologiques sont des outils fiables pour déterminer la qualité du milieu aquatique. Voici une description de chaque compartiment :

Poissons Ce compartiment concerne l'ichtyofaune (peuplements de poissons). L'indice biologique associé est l'Indice Poisson Rivière (IPR) [AFN04b]. Calculé sur une station de prélèvements, cet indice vise à mesurer l'écart entre les peuplements de poissons actuels et ceux qui devraient être présents sans l'impact de l'activité humaine (situation de référence). Pour déterminer le score de l'indice, les hydrobiologistes prennent en considération des paramètres tels

que la température de l'air moyenne ou bien la profondeur et la largeur de la station. Un score de 0 signifie qu'il n'y a aucun écart entre la situation mesurée et celle de référence. Compris entre 0 et $+\infty$, un score élevé indique une mauvaise qualité du cours d'eau.

Macroinvertébrés Les macroinvertébrés benthiques d'eau douce vivent au fond des rivières et des lacs. Ils mesurent plus de 0,5mm et sont donc reconnaissables à l'œil nu. Le compartiment des macroinvertébrés rassemble par exemple des espèces de mollusques, de crustacés et de larves d'insectes. L'indice biologique associé est l'Indice biologique global normalisé (IBGN) [AFN04a]. Le calcul de cet indice est basé sur la présence ou l'absence de certains taxons de macroinvertébrés polluo-sensibles. Par exemple, la présence d'un taxon fortement sensible aux pollutions sera le signe d'une bonne qualité de l'eau. Dans le cadre de l'IBGN, une liste de 128 taxons est utilisée pour le calcul de la note. Cette note est un score entre 0 et 20 basé sur les taxons trouvés et leur variété taxonomique dans l'échantillon. Un score de 20 est représentatif d'une excellente qualité de la rivière, alors qu'un score de 0 est caractéristique d'une très mauvaise qualité.

Oligochètes Le taxon oligochète rassemble plusieurs espèces de vers comme le tubifex ou le lombric (ver de terre). L'indice biologique associé est l'Indice Oligochètes de Bioindication des Sédiments (IOBS) [AFN02]. En fonction des taxons d'oligochètes échantillonnés, cet indice permet de mettre en évidence des pollutions aux métaux lourds ou aux Polychlorobiphényles (PCB) et est utile pour l'identification des rejets polluants. Le score de cet indice est obtenu en comptant le nombre total de taxons contenus dans l'échantillon et calculant, parmi la famille des *tubificidae*, le pourcentage du groupe dominant. Compris entre 0 et 10, un score de 0 signifie qu'aucun oligochète n'a été trouvé dans l'échantillon. Un score supérieur à 6 est représentatif d'une bonne qualité du milieu aquatique.

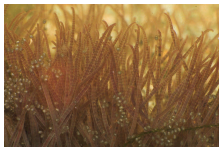
Macrophytes Le terme macrophyte est utilisé pour caractériser les espèces de plantes aquatiques qui sont visibles à l'œil nu. Ce terme rassemble différents types d'algues (espèces rouges, vertes et brunes) mais aussi les roseaux ou les nénuphars. L'indice biologique associé est l'Indice Biologique Macrophytique en Rivière (IBMR) [AFN03]. L'utilisation de cet indice vise à déterminer le statut trophique des rivières. Il est adapté à la mise en évidence des teneurs en ammonium, en orthophosphates ainsi que les pollutions organiques. Comme pour l'IPR, l'IBMR mesure un écart à la situation idéale (écart à la référence).

Le score est calculé à partir d'une liste de 208 taxons de macrophytes, de leur abondance, d'une cote spécifique et d'une valeur de sténo-euryécie [10]. Allant de 0 à 20, un score de 20 est typique d'un très bon état du milieu aquatique et un score de 0 d'un très mauvais état.

Diatomées Contrairement aux macrophytes qui rassemblent les espèces d'algues visibles à l'œil nu, le compartiment des diatomées regroupe les taxons de microalgues unicellulaires qui ont la particularité d'être entourées d'un squelette externe en silice. Il existe actuellement plus de 100 000 espèces à l'échelle mondiale et sont un composant majeur du phytobenthos. L'indice biologique associé est l'Indice biologique diatomées (IBD) [AFN07]. Cet indice est utilisé pour le suivi temporel ou spatial des milieux aquatiques ainsi que pour mettre en évidence les perturbations sur le milieu. Ce score est basé sur un sous-ensemble d'espèces sélectionnées pour leur polluo-sensibilité. Un score entre 0 et 20 est obtenu, où un score de 20 indique un très bon état du milieu aquatique.

Les cinq indices biologiques présentés ci-dessus, obtenus à partir de l'analyse de chaque groupe biologique, sont la base de l'analyse hydrobiologique des cours d'eau dans le cadre de la DCE. Pour chaque groupe, nous avons présenté le principal indice biologique qui lui correspond. Cependant, certains compartiments peuvent avoir d'autres indices, comme l'IBG-DCE [AFN09, AFN10] ou l'IBGA (en cours de normalisation) pour les macroinvertébrés. La figure 6.3 donne des exemples de taxons pour les groupes macroinvertébrés (6.3a), oligochètes (6.3b), macrophytes (6.3c) et diatomées (6.3d).

(a) *Odonata*



(b) *Naididae*

(c) *Callitriche*

(d) *Biddulphia*

FIGURE 6.3 – Exemple de taxons

10. Valeur traduisant la capacité d'un macrophyte à supporter d'importantes variations écologiques

6.2.2 Données physico-chimiques

Pour vivre, les espèces animales et végétales ont toutes besoin de nourriture. Les besoins de chaque espèce sont représentés par la chaîne alimentaire. Certaines espèces comme les poissons vont se nourrir d'autres poissons, de macroinvertébrés ou bien d'oligochètes. Tout en bas de la chaîne alimentaire se situe la flore qui elle se nourrit des éléments chimiques présents dans le milieu. Par exemple, les diatomées ont besoin de matière phosphorée pour se développer. Cependant, un excès de phosphore va au contraire entraîner une disparition de ces diatomées. De la même manière, certains éléments comme l'oxygène dissous sont nécessaires à toute forme vivante. Un manque d'oxygène entraîne une asphyxie du milieu et la mort de la faune et de la flore aquatique. Ainsi, la physico-chimie comprend la mesure de plusieurs paramètres physiques et chimiques dans les rivières. Ces éléments peuvent être à la fois nécessaires au bon développement de la biodiversité et une source de pollution. Le terme de physico-chimie rassemble principalement deux groupes différents, les macro-polluants et les micro-polluants.

Macro-polluants Ce groupe rassemble les éléments qui sont naturellement présents dans les cours d'eau et sont le plus souvent nécessaires au bon fonctionnement de l'écosystème. Par exemple, nous pouvons citer la concentration en oxygène dissous, la concentration en nitrates ou bien la teneur en matières phosphorées. Bien que nécessaires au bon fonctionnement du cours d'eau à des teneurs dites normales, un excès de macro-polluants peut entraîner une pollution néfaste sur la biodiversité. L'impact de l'excès de nitrates sur l'environnement, majoritairement causé par l'élevage intensif [TCM+02] mais aussi par l'abus de fertilisants [ZTZL96], est par exemple bien connu.

Micro-polluants À l'inverse des macro-polluants ils ne sont pas nécessaires à l'écosystème, bien au contraire. Ils sont appelés micro-polluants car une dose infime entraîne l'intoxication du milieu. Plus nombreux que les macro-polluants (des milliers), ils rassemblent par exemple beaucoup de molécules synthétiques comme les hydrocarbures aromatiques polycycliques (HAP), les pesticides et les polychlorobiphényles (PCB). Actuellement, les Antilles françaises sont par exemple touchées par une pollution à la chlordécone entraînée par l'utilisation de pesticides pour lutter contre le charançon du bananier. Bien que sujet à controverse, de nombreux travaux ont démontré ses effets néfastes sur le corps humain [Guz82, MCK+07]. Les micro-polluants rassemblent aussi des éléments présents naturellement dans le milieu comme les métaux lourds. Par

exemple, en conséquence de forages de puits trop profonds dans les années 1960, la pollution à l'arsenic des eaux souterraines au Bangladesh [SLR00] a entraîné une crise sanitaire majeure, où cette pollution est la cause d'un décès sur cinq dans ce pays.

Base de données biologiques et physico-chimiques

Le tableau 6.1 donne un exemple de prélèvements biologiques et physico-chimiques mesurés dans les deux stations de prélèvements présentées figure 6.1. Les échantillons de la station 1 correspondent à la période temporelle comprise entre février 2007 et juillet 2008. Les échantillons de la station 2 correspondent à la période temporelle comprise entre janvier 2004 et août 2005. Cinq paramètres physico-chimiques et un indice biologique (l'IBGN) ont été mesurés à différentes dates. L'ammonium (NH_4^+), l'azote Kjeldahl (NKJ) et les nitrates (NO_2^-) sont représentatifs des matières azotées. Les orthophosphates (PO_4^{3-}) et le phosphore total (P) correspondent au niveau de phosphore dans l'eau. Dans cet exemple illustratif, nous considérons l'indicateur biologique IBGN. Par exemple, une valeur d'orthophosphates de 0.132 a été mesurée en juillet 2004 dans la station 2. Une valeur d'IBGN de 8 sur cette même station a été mesurée en septembre 2004.

Ce jeu de données exemple sera utilisé tout au long de ce chapitre pour montrer les différents traitements appliqués à la base de données Fresqueau. Avant de présenter ces traitements, nous présentons un état de l'art sur les méthodes de fouille de données appliquées aux données hydrobiologiques.

6.3 État de l'art sur la fouille de données appliquée à l'hydrobiologie

Il existe de nombreux travaux qui traitent de l'application des méthodes de fouille aux données hydrobiologiques.

Un nombre important d'études se sont focalisées sur les communautés de macroinvertébrés [BLBBT09, GDG+07, DDD+07, DGGP04, DGP03]. Par exemple, [DDD+07] a étudié l'application de modèles basés sur des arbres de décision pour prédire la viabilité de l'habitat de certains taxons de macroinvertébrés dans la rivière Axios (Grèce). Les auteurs ont pris en compte la physico-chimie ainsi que les caractéristiques structurelles de la rivière. Dans le même objectif, l'efficacité des réseaux de neurones pour la prédiction de taxons de macroinvertébrés dans la rivière

Station	**Date**	$\mathbf{NH_4^+}$	**NKJ**	$\mathbf{NO_2^-}$	$\mathbf{PO_4^{3-}}$	**P**	**IBGN**
1	02/07	-	-	-	0.123	0.032	-
	06/07	-	0.672	0.026	-	-	-
	07/07	0.088	1.235	0.134		0.011	-
	09/07	-	-	-	-	-	17
	12/07	0.154	-	0.246	0.168	0.338	-
	02/08	0.062	0.040	0.091	0.025	0.003	-
	04/08	-	0.023	0.198	-	-	-
	05/08	-	-	-	-	-	12
	07/08	-	-	-	0.046	0.009	-
2	01/04	0.043	0.146	0.421	-	-	-
	04/04	-	-	-	1.325	0.093	-
	07/04	2.331	7.993	0.252	0.132	0.266	-
	08/04	-	1.414	-	-	-	-
	09/04	-	-	-	-	-	8
	11/04	0.117	0.0844	-	0.688	-	-
	12/04	-	-	-	0.067	0.278	-
	03/05	-	0.182	0.0310	0.137	-	-
	06/05	0.004	-	0.012	0.035	0.134	-
	08/05	-	-	-	-	-	10

TABLE 6.1 – Jeu de données exemple

Zalm (Belgique) a été montrée par [DGGP04].

Également, l'implication des altérations hydrologiques sur les communautés de poissons dans la rivière Illinois a été identifiée par [YCH08]. En se basant sur 32 indicateurs d'altération hydrologique, les auteurs ont mis en évidence les indicateurs les plus pertinents pour l'écologie.

D'autres auteurs se sont focalisés sur la flore plutôt que la faune. Récemment, l'impact de la physico-chimie sur les diatomées a été étudié dans [KNM+10] où les auteurs ont utilisé des arbres de régression multi-ciblés et ont identifié un impact significatif des ions métalliques et des nutriments sur les diatomés. [ROC+13] a étudié les populations de phytoplanctons dans le lac Kinneret (Israël). En appliquant un algorithme évolutionnaire hybride, les auteurs ont montré qu'utiliser en même temps les données physico-chimiques et biologiques dans les modèles améliore la prévision de la dynamique des populations de phytoplanctons. Par ailleurs, dans

[BLBBT09], les auteurs se sont appuyés sur l'analyse formelle de concepts pour étudier les traits des taxons de macrophytes dans le bassin versant Rhin-Meuse (France). L'objectif a été de lier les variables environnementales avec des granularités de traits biologiques pour identifier les groupes de taxons adaptés à un contexte environnemental particulier. En utilisant le même jeu de données que [BLBBT09], les auteurs de [DLBH13] ont cette fois-ci utilisé l'analyse relationnelle de concepts pour l'étude des traits des taxons de macrophytes.

Les méthodes de la littérature montrent l'importance de combiner les données biologiques et physico-chimiques pour identifier de la connaissance pertinente. Cependant, aucune étude n'a pris en compte l'aspect temporel en utilisant par exemple des méthodes de fouilles de motifs temporels, qui sont pertinentes pour l'analyse de la dynamique de la pollution aquatique et qui fournissent également une connaissance simple à analyser. Les méthodes de motifs proposées dans cette thèse ont pour objectif de prendre en compte cet aspect temporel des données.

6.4 Pré-traitements sur les données

Les données telles que présentées dans le tableau 6.1 ne sont cependant pas adaptées aux méthodes de fouille de données basées sur l'exploration de séquences temporelles. Nous présentons maintenant les trois étapes que nous avons appliquées au jeu de données pour pouvoir mettre en œuvre les méthodes proposées : (1) la discrétisation des données brutes ; (2) la transformation du jeu de données initial en un jeu de données de séquences et (3) le découpage du jeu de données de séquences en fonction des classes de qualité des indices biologiques.

6.4.1 Discrétisation des données

Cette première étape du pré-traitement consiste à effectuer la discrétisation des données. Le processus de discrétisation n'est pas une tâche évidente, il faut choisir un nombre d'intervalles et les plages de valeurs que vont prendre ces intervalles. Ceci se fait généralement d'une manière empirique en l'absence d'études préalables. Les données physico-chimiques et biologiques sont déjà étudiées depuis de nombreuses années. Ainsi, les hydrobiologistes ont déterminé des classes de qualité symbolisées par des couleurs différentes : de bleu à rouge, en passant par le vert, le jaune et l'orange. Ce dégradé de couleurs est relatif à la qualité estimée du milieu en fonction de la plage de valeurs dans laquelle est située la valeur d'un paramètre donné

(physico-chimique ou biologique). Il existe actuellement plusieurs standards. Pour discrétiser les données Fresqueau, nous utilisons les normes AFNOR de chaque indice biologique [AFN04a, AFN07, AFN04b, AFN02, AFN03] ainsi que le standard SEQ-eau [11] car ces deux standards ont plusieurs avantages :

- Le nombre de variables du jeu de données (pour la physico-chimie) est considérablement réduit en se basant sur les altérations proposées par le SEQ-eau. En effet, cela nous permet de passer de plusieurs centaines de paramètres à une quinzaine grâce à un regroupement par familles de polluants.
- La discrétisation est représentée par différentes couleurs qui caractérisent la viabilité du milieu aquatique. L'objectif est d'utiliser ces couleurs pour faciliter la visualisation et l'interprétation des résultats, puisque les couleurs de qualité parlent aux hydrobiologistes.
- Ces standards sont valables pour la France entière, ils sont donc simples à mettre en place dans le cas des données Fresqueau puisqu'elles couvrent plusieurs régions hydrographiques.

Nous présentons maintenant chacun de ces standards. Les normes AFNOR concernent les indices biologiques alors que le SEQ-eau s'applique aux paramètres physico-chimiques.

Normes AFNOR

Ces normes proposent des classes de qualité concernant les notes d'indices biologiques. Ainsi, pour chacun des cinq indices présentés dans la section 6.2.1 : IPR, IBGN, IOBS, IBMR et IBD, cinq classes de qualité représentées par des couleurs ont été mises en place. Le tableau 6.2 donne les intervalles de discrétisation pour ces cinq indices.

Cela veut par exemple dire qu'une valeur d'IBGN de 15 est associée à une classe de qualité verte et qu'une valeur d'IBGN de 6 est discrétisée en classe de qualité orange.

Norme SEQ-eau

Le SEQ-eau est un standard utilisé pour la discrétisation des paramètres physico-chimiques. Son fonctionnement est proche des normes AFNOR (même nombre d'in-

11. http ://sierm.eaurmc.fr/eaux-superficielles/fichiers-telechargeables/grilles-seq-eau-v2.pdf

Indices	**Bleu**	**Vert**	**Jaune**	**Orange**	**Rouge**
IPR	[0,7]	]7,16]	]16,25]	]25,36]	]36,∞]
IBGN	[20,17]	]17,13]	]13,9]	]9,5]	]5,0]
IOBS	[10,6]	]6,3]	]3,2]	]2,1]	]1,0]
IBMR	[20,17]	]17,13]	]13,9]	]9,5]	]5,0]
IBD	[20,17]	]17,13]	]13,9]	]9,5]	]5,0]

TABLE 6.2 – Classes de qualité des indices biologiques selon leur norme AFNOR

tervalles transcrits en couleurs) et permet également de réduire le nombre de paramètres physico-chimiques. Avec les indices biologiques, le problème ne se pose pas puisque ceux-ci sont déjà une synthèse de plusieurs mesures biologiques, mais la physico-chimie fait référence à plusieurs centaines de paramètres physico-chimiques. De plus, il y a de nombreuses données manquantes puisque lorsqu'un prélèvement est effectué, seule une partie des paramètres physico-chimiques est mesurée, en fonction du coût de l'intervention et de l'objectif du prélèvement. Par exemple, mesurer à chaque prélèvement la teneur pour chaque pesticide dans le milieu aquatique est coûteux. Le SEQ-eau permet ainsi de regrouper des ensembles de paramètres physico-chimiques par grandes familles, ce qui permet de diminuer le nombre de paramètres à une quinzaine environ. Dans le standard, une famille de paramètres est appelée altération. Cependant, nous n'utiliserons pas ce terme mais le terme macro-paramètre. En effet, les valeurs d'altérations telles que définies par le SEQ-eau sont calculées à partir de méthodes statistiques (agrégation de valeurs dans le temps). Ce n'est pas notre cas puisque les données ont pour objectif d'être transformées en séquences temporelles pour ainsi éviter cette agrégation. Nous utilisons donc les classes de qualité des altérations pour créer des macro-paramètres. Les macro-paramètres regroupent les paramètres physico-chimiques en fonction de leur nature et de leur fonction. Par exemple, il existe les macro-paramètres pour les pesticides (PEST), pour les hydrocarbures (HAP), pour les minéraux (MINE) ou encore comme dans le tableau 6.3, les macro-paramètres pour les matières azotées hors nitrates (AZOT) et pour les matières phosphorées (PHOS). Le tableau 6.3 donne les intervalles de discrétisation de deux macro-paramètres : AZOT et PHOS. AZOT étant calculé à partir de trois paramètres physico-chimiques et PHOS à partir de deux.

Le mode de calcul de la valeur d'un macro-paramètre est le suivant : pour chaque paramètre physico-chimique appartenant à un macro-paramètre donné, sa classe de

Groupe	Paramètre	Bleu	Vert	Jaune	Orange	Rouge
AZOT	NH_4^+ (mg/l)	[0,0.1[	[0.1,0.5[	[0.5,2[	[2,5[	[5,∞]
	NKJ (mg/l)	[0,1[	[1,2[	[2,4[	[4,10[	[10,∞]
	NO_2^- (mg/l)	[0,0.03[	[0.03,0.3[	[0.3,0.5[	[0.5,1[	[1,∞]
PHOS	PO_4^{3-} (mg/l)	[0,0.1]	[0.1,0.5[	[0.5,1[	[1,2[	[2,∞]
	P (mg/l)	[0,0.05]	[0.05,0.2[	[0.2,0.5[	[0.5,1[	[1,∞]

TABLE 6.3 – Classes de qualité des macro-paramètres physico-chimiques selon les altérations de la norme SEQ-eau

qualité (couleur) est calculée. Ensuite, la pire valeur de qualité est assignée au macro-paramètre. Prenons le macro-paramètre PHOS donné dans le tableau 6.3 avec une valeur d'orthophosphate (PO_4^{3-}) de 0,026 et une valeur de phosphore total (P) de 0,67. PO_4^{3-} est discrétisé en classe de qualité bleue et P est discrétisé en classe de qualité jaune. Le macro-paramètre PHOS est alors discrétisé en classe de qualité jaune qui est la pire classe de qualité calculée (dans notre exemple le paramètre P). De plus, il est important de noter qu'un macro-paramètre peut être calculé même si certains des paramètres physico-chimiques qu'il contient n'ont pas été mesurés, la mesure d'au moins un paramètre suffit. La raison est que de nombreux macro-paramètres contiennent un important nombre de paramètres et que certains de ces paramètres ne sont pas toujours mesurés par les hydrobiologistes. Si un tel choix entraîne une évidente généralisation des données, il permet de réduire le nombre de dimensions et d'augmenter la densité du jeu de données. Prenons par exemple les prélèvements effectués en Avril 2008 dans la station 1 (tableau 6.1), les paramètres NKJ et NO_2^- ont été mesurés mais NH_4^+ ne l'a pas été. En se basant sur le tableau 6.3, le paramètre NKJ est discrétisé en classe de qualité bleue et NO_2^- en classe de qualité verte, le macro-paramètre AZOT prend alors la valeur de classe de qualité verte.

Application sur le jeu de données

À partir du jeu de données hydrobiologique donné par le tableau 6.1, nous donnons dans le tableau 6.4 le jeu de données discrétisé obtenu en utilisant les classes de qualité fournies par les normes SEQ-eau et AFNOR. Nous observons que le nombre de variables a été réduit. En effet, à partir du tableau 6.3, les paramètres physico-chimiques NH_4^+, NKJ et NO_2^- ont été réduits en un seul macro-paramètre AZOT et les paramètres physico-chimiques PO_4^{3-} et P ont été réduits

en un seul macro-paramètre PHOS. Grâce au calcul des macro-paramètres, même lorsqu'il y a des valeurs manquantes, la complétude du jeu de données a augmenté de 42,1% à 45,6% (nous avons divisé le nombre de champs ayant une valeur par le nombre total de champs). Le gain est encore plus important dans le jeu de données réel puisque certains macro-paramètres possèdent plusieurs dizaines de paramètres physico-chimiques.

Station	**Date**	**AZOT**	**PHOS**	**IBGN**
1	02/07	-	Vert	-
	06/07	Bleu	-	-
	07/07	Vert	Bleu	-
	09/07	-	-	Bleu
	12/07	Vert	Jaune	-
	02/08	Vert	Bleu	-
	04/08	Vert	-	-
	05/08	-	-	Jaune
	07/08	-	Bleu	-
2	01/04	Jaune	-	-
	04/04	-	Orange	-
	07/04	Orange	Jaune	-
	08/04	Vert	-	-
	09/04	-	-	Orange
	11/04	Vert	Jaune	-
	12/04	-	Jaune	-
	03/05	Vert	Vert	-
	06/05	Bleu	Vert	-
	08/05	-	-	Jaune

TABLE 6.4 – Jeu de données hydrobiologique discrétisé

Après avoir discrétisé le jeu de données initial, nous passons maintenant à la transformation de celui-ci en un ensemble de séquences.

6.4.2 Génération des séquences

Dans cette étape, le but est de construire une séquence pour chacune des stations de prélèvements en ordonnant l'ensemble des échantillons en fonction de leur date de

prélèvement. Les séquences sont des séquences d'itemsets telles que définies dans le chapitre 2, où les items qui composent les itemsets des séquences sont représentés par des variables physico-chimiques et biologiques. Le tableau 6.5 donne les séquences obtenues à partir de la base de données hydrobiologique discrétisée du tableau 6.4. Par exemple, si nous considérons la station 1, nous observons une valeur de PHOS en vert à la date de prélèvement 02/07, une valeur d'AZOT en bleu à la date de prélèvement 06/07 et une valeur de PHOS en bleu ainsi qu'une valeur d'AZOT en vert à la date de prélèvement 07/07. La séquence de la station de prélèvements 2 commence donc par la sous-séquence $\langle$(PHOS^{Vert}) (AZOT^{Bleu}) (AZOT^{Vert}, PHOS^{Bleu})$\rangle$. Ceci signifie que l'item PHOS^{Vert} est temporellement suivi par l'item AZOT^{Bleu}, lui même suivi par les deux items AZOT^{Vert} et PHOS^{Bleu}.

Station	**Séquence**
1	$\langle$(PHOS^{Vert})(AZOT^{Bleu})(AZOT^{Vert},PHOS^{Bleu})(IBGN^{Bleu}) (AZOT^{Vert},PHOS^{Jaune})(AZOT^{Vert},PHOS^{Bleu})(AZOT^{Vert}) (IBGN^{Jaune})(PHOS^{Bleu})$\rangle$
2	$\langle$(AZOT^{Jaune})(PHOS^{Orange})(AZOT^{Orange}, PHOS^{Jaune}) (AZOT^{Vert})(IBGN^{Orange})(AZOT^{Vert},PHOS^{Jaune})(PHOS^{Jaune}) (AZOT^{Vert},PHOS^{Vert})(AZOT^{Bleu},PHOS^{Vert}) (IBGN^{Jaune})$\rangle$

TABLE 6.5 – Transformation du tableau 6.4 en un jeu de données de séquences

6.4.3 Base de données de séquences par classe de qualité biologique

Cependant, la transformation de la base de données brute en séquences n'est pas suffisant pour relier la physico-chimie avec la biologie. En effet, les hydrobiologistes sont par exemple intéressés par les valeurs de physico-chimie que l'on retrouve plus fréquemment dans un cours d'eau où un mauvais état biologique a été mesuré, que dans un cours d'eau de très bonne qualité biologique, et inversement. Intuitivement, l'idée est de découper le jeu de données de séquences obtenu en plusieurs jeux de données, où chaque nouveau jeu de données est relatif à un état de qualité biologique du cours d'eau. L'ensemble de ces nouveaux jeux de données de qualité biologique forme ainsi un jeu de données composé de plusieurs classes/catégories. Cela permet ainsi d'appliquer des méthodes de fouille de données de type discriminantes sur les

classes/catégories de qualité biologique.

Dans la base de données du tableau 6.5, la biologie et la physico-chimie sont toutes deux incluses dans les séquences. Ainsi, utiliser la biologie pour fabriquer différentes catégories (en fonction de la qualité de l'indice) nécessite de scanner les séquences pour identifier les variables biologiques. Pour chaque variable biologique, nous souhaitons récupérer les mesures de paramètres physico-chimiques qui la précèdent selon un intervalle de temps donné. Nous avons ainsi mis en place un processus de découpage des séquences brutes. Considérons la variable biologique IBGN^{Bleu}. Il s'agit de récupérer toutes les sous-séquences de la base de données qui précèdent les variables IBGN^{Bleu} en fonction d'un certain intervalle de temps représenté par une fenêtre temporelle.

Station	**Séquence**
1	$\langle$(PHOS^{Vert})(AZOT^{Bleu})(AZOT^{Vert},PHOS^{Bleu})$\boxed{\text{(IBGN}^{Bleu}\text{)}}$ (AZOT^{Vert},PHOS^{Jaune})(AZOT^{Vert}, PHOS^{Bleu})(AZOT^{Vert}) $\boxed{\text{(IBGN}^{Jaune}\text{)}}$(PHOS^{Bleu})$\rangle$
2	$\langle$(AZOT^{Jaune})(PHOS^{Orange})(AZOT^{Orange}, PHOS^{Jaune}) (AZOT^{Vert})$\boxed{\text{(IBGN}^{Orange}\text{)}}$ (AZOT^{Vert},PHOS^{Jaune})(PHOS^{Jaune}) (AZOT^{Vert},PHOS^{Vert})(AZOT^{Blue},PHOS^{Vert}) $\boxed{\text{(IBGN}^{Jaune}\text{)}}\rangle$

TABLE 6.6 – Processus de découpage du jeu de données du tableau 6.5

Nous illustrons cette étape en découpant les séquences de la base de données du tableau 6.5. Dans celle-ci, l'indice biologique IBGN est représenté par trois valeurs de classes de qualité : bleu, jaune et orange, répartis en quatre prélèvements. Grâce à l'opération présentée précédemment, nous récupérons les sous-séquences de variables physico-chimiques qui précèdent chacune des mesures de l'IBGN et nous les regroupons en trois bases de données distinctes, chacune représentative d'une classe de qualité de l'IBGN. Nous obtenons ainsi un jeu de données décomposé en classes : IBGN^{Bleu}, IBGN^{Jaune} et IBGN^{Orange}. Pour cet exemple, nous avons sélectionné une fenêtre temporelle de six mois. Ce processus est illustré par le tableau 6.6 où les variables d'indices biologiques sont encadrées, et où les variables physico-chimiques non-sélectionnées par la fenêtre glissante sont grisées. Par exemple, la première variable PHOS^{vert} de la séquence de la station 1 n'est pas collectée puisqu'elle a été mesurée sept mois avant la première mesure de l'IBGN. Le tableau 6.7 représente

ainsi le résultat du processus avec pour chaque classe de qualité de l'IBGN, l'ensemble des séquences collectées.

Classe de qualité	**Séquences**
IBGN^{Bleu}	$\langle(\text{AZOT}^{Bleu})(\text{AZOT}^{Vert},\text{PHOS}^{Bleu})\rangle$
IBGN^{Jaune}	$\langle(\text{AZOT}^{Vert},\text{PHOS}^{Vert})(\text{AZOT}^{Bleu},\text{PHOS}^{Vert})\rangle$ $\langle(\text{AZOT}^{Vert},\text{PHOS}^{Jaune})(\text{AZOT}^{Vert},\text{PHOS}^{Bleu})$ $(\text{AZOT}^{Vert})\rangle$
IBGN^{Orange}	$\langle(\text{PHOS}^{Orange})(\text{AZOT}^{Orange},\text{PHOS}^{Jaune})$ $(\text{AZOT}^{Vert})\rangle$

TABLE 6.7 – Ensembles de séquences associés aux classes de qualité de l'IBGN

Les méthodes de fouille de données temporelles peuvent maintenant être appliquées. En effet, les différents pré-traitements appliqués ont permis de transformer une base de données classique composée de variables numériques, en une base de données composée de séquences de variables discrètes et divisée en classes de qualité biologique.

6.5 Synthèse

Nous avons présenté dans ce chapitre le contexte applicatif de cette thèse qui est relatif à l'étude de données hydrobiologiques dans le cadre du projet ANR Fresqueau.

Le pré-traitement effectué sur la base de données permet ainsi d'appliquer les différentes méthodologies présentées dans les chapitres 3, 4 et 5. Ainsi, dans le chapitre suivant, chaque méthodologie est appliquée pour l'étude du lien entre les données biologiques et les données physico-chimiques dans les cours d'eau.

Dans un tel contexte, l'intérêt et l'originalité de cette application est d'être, à notre connaissance, la première à mettre en œuvre un processus d'extraction de connaissances basé sur les motifs temporels.

CHAPITRE 7

Exploration des données hydrobiologiques

Contents

7.1 Introduction

Dans le chapitre précédent, nous avons présenté le contexte applicatif de cette thèse : les données hydrobiologiques, ainsi qu'un processus pour générer des jeux de données de séquences temporelles associées à des classes. Les différents jeux de données générés à partir de ce processus ont été utilisés en entrée des algorithmes présentés dans les chapitres précédents. Nous présentons dans ce chapitre les résultats obtenus.

Afin de permettre aux hydrobiologistes de tester facilement nos méthodes, nous les avons implémentées dans un logiciel de visualisation et de navigation qui permet de filtrer les données, d'appliquer les méthodes et de visualiser les résultats.

Pour finir, nous donnons l'interprétation des informations extraites par les experts. Cette interprétation nous a permis d'évaluer l'intérêt d'appliquer des méthodes par motifs temporels dans le contexte des données hydrobiologiques.

Tout d'abord, dans la section 7.2, nous présentons les résultats obtenus avec chacune des méthodes : la fouille de motifs partiellement ordonnés clos discriminants, la sélection des motifs d'intérêt et pour finir l'extraction de consensus discriminants. Ensuite, nous présentons dans la section 7.3 l'intégration des différentes méthodes dans un logiciel de visualisation. Pour finir, nous discutons de la connaissance extraite dans la section 7.4 et nous faisons une synthèse de ce chapitre dans la section 7.5.

7.2 Application des différentes méthodes

Avant de présenter les résultats obtenus, nous donnons quelques statistiques importantes concernant les jeux de données générés. Nous avons généré trois sous-jeux de données pour les indices biologiques *IBGN*, *IBD* et *IPR*, qui correspondent respectivement aux compartiments biologiques des macroinvertébrés, des diatomées et des poissons.

L'application de la procédure présentée dans le chapitre 6 sur chaque indice nous a permis de générer un jeu de données associé. Nous avons choisi différents intervalles de temps pour capturer les séquences qui précèdent une mesure biologique. Pour les indices biologiques *IBGN* et *IPR*, nous avons sélectionné un intervalle de quatre mois avant une mesure. Pour l'*IBD*, cet intervalle a été réduit à deux mois avant une mesure. La raison est que les populations de diatomées se renouvellent plus rapidement que les populations de macroinvertébrés et de poissons. Il est donc

nécessaire de mettre un intervalle temporel plus petit pour l'*IBD*.

Le tableau 7.1 synthétise le nombre de séquences que nous avons généré pour chaque classe de qualité, pour chaque indice biologique.

	IBGN	**IBD**	**IPR**
Bleu	1056	1076	52
Vert	2405	1375	162
Jaune	1282	532	126
Orange	556	108	76
Rouge	89	17	62

TABLE 7.1 – Jeux de données générés pour l'*IBGN*, l'*IBD* et l'*IPR*

Par exemple, nous avons généré 1375 séquences pour la classe de qualité verte de l'*IBD*. Nous observons que pour chaque indice, nous obtenons des classes de séquences très déséquilibrées. Par exemple, la classe verte de l'*IBGN* contient 2405 séquences alors que la classe rouge pour ce même indice en contient 89.

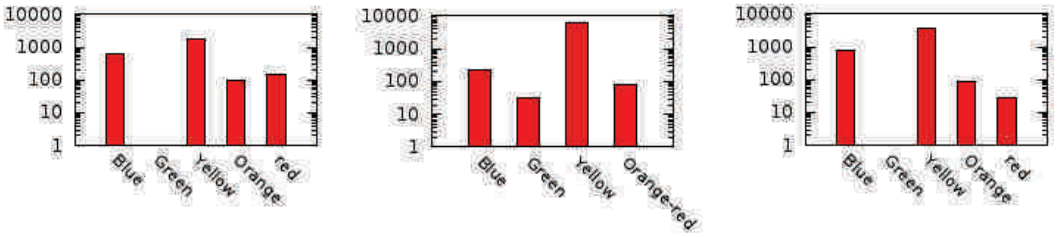
Certaines classes, comme la classe de qualité rouge pour l'*IBD*, sont peu représentées (17 séquences). Extraire des motifs sur cette classe avec une fréquence minimale de 10% revient à extraire des motifs qui sont seulement supportés par deux séquences de cette classe. Statistiquement, extraire de l'information sur seulement deux instances d'un problème (séquences) n'est pas pertinent. Ainsi, nous avons fusionné les classes rouge et orange de l'*IBD* en une nouvelle classe orange-rouge composée de 125 séquences.

Nous présentons maintenant les résultats obtenus avec chacune des méthodes.

7.2.1 Extraction de motifs partiellement ordonnés clos discriminants

Nous présentons tout d'abord l'extraction des motifs partiellement ordonnés clos discriminants en utilisant *DiscriminantOrderSpan* (voir chapitre 4). Pour ces expérimentations, nous avons sélectionné un seuil de fréquence minimale de 10% dans chaque classe de qualité. Nous avons choisi cette valeur de seuil car elle permet d'extraire un volume intéressant de motifs discriminants dans les données. Nous n'avons pas choisi de seuil inférieur à cette valeur pour rester statistiquement significatif suite à des discussions avec les experts.

Les figures 7.1a, 7.1b et 7.1c nous montrent le nombre de motifs discriminants extraits dans chaque classe de qualité pour chaque indice biologique.

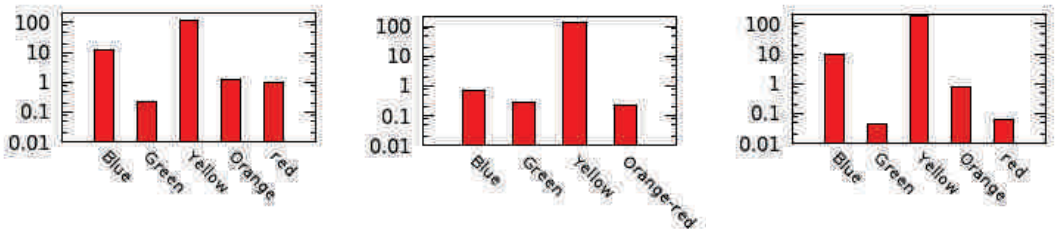


(a) IBGN (b) IBD (c) IPR

FIGURE 7.1 – Nombre de motifs partiellement ordonnés clos dans chaque classe de qualité de chaque indice

Nous observons que le nombre de motifs discriminants dans chaque classe est très déséquilibré. Par exemple, 6202 motifs sont extraits de la classe jaune de l'*IBD* alors que seulement 31 sont extraits de la classe verte pour ce même indice. Une hypothèse à ce déséquilibre est d'une part la différence du nombre de séquences dans chaque classe, mais aussi l'hétérogénéité des classes de qualité d'un point de vue hydrobiologique. Pour les trois indices biologiques, la classe jaune est celle qui contient le volume le plus important de motifs extraits, au moins deux fois plus que dans les autre classes. De plus, aucun motif n'a été trouvé dans la classe verte de l'*IPR* et seulement un a été extrait de la classe verte pour l'*IBGN*.

Le temps de calcul dans chaque classe de qualité pour chaque indice biologique est donné par les figures 7.2a, 7.2b et 7.2c.



(a) IBGN (b) IBD (c) IPR

FIGURE 7.2 – Temps de calcul (en secondes) dans chaque classe de qualité de chaque indice

Nous observons que le nombre de motifs discriminants extraits est corrélé avec le temps d'extraction. En effet, le temps que met un algorithme d'extraction de motifs est très souvent lié au volume de motifs en sortie de cet algorithme. Nous observons que le temps de calcul maximum est de 177,11 secondes pour la classe jaune de l'*IPR* (3553 motifs), tandis que le temps de calcul minimum est de 0,046 secondes pour la classe verte de l'*IPR* (aucun motif extrait). Dans tous les cas, le temps de calcul est acceptable puisque cette application aux données hydrobiologiques ne requiert pas une exécution en temps réel et n'excède pas trois minutes dans le pire des cas.

7.2.2 Sélection de motifs d'intérêt

Bien que nous ayons extrait un volume intéressant de motifs dans les données, il n'est pas possible de tous les restituer aux experts. L'analyse et l'interprétation seraient bien trop difficile. Nous avons alors appliqué la méthode *Pattern-Balanced* (voir chapitre 4) pour sélectionner des motifs d'intérêt dans ces ensembles de milliers de motifs. Pour rappel, cette méthode permet de considérer à la fois les motifs les plus fréquents et les plus discriminants, tout en minimisant le fait que deux motifs soient supportés par des ensembles de séquences proches (non-redondance).

L'algorithme *Pattern-Balanced* prend en paramètre un entier k qui correspond au nombre de motifs que nous souhaitons restituer. Nous avons choisi empiriquement le nombre de 15 motifs puisqu'après de nombreux tests et discussions avec les experts, nous avons observé que cela permettait de restituer un ensemble acceptable de motifs, tout en capturant une connaissance diversifiée sur les données (non-redondance).

Pour chaque indice biologique (*IBGN*, *IBD* et *IPR*), nous présentons deux exemples de motifs partiellement ordonnés clos discriminants qui ont été extraits et filtrés depuis la classe bleu et rouge de chaque indice (figures 7.3, 7.4, 7.5, 7.6, 7.7 et 7.8). Ces deux classes sont les plus extrêmes en terme de qualité biologique (très bonne qualité et très mauvaise qualité). Toutefois, une exception existe pour l'indice biologique *IBD* puisque, comme dit précédemment, les classes orange et rouge ont été fusionnées en une classe orange-rouge. Pour chaque motif, nous donnons également leur fréquence dans les autres classes de l'indice considéré.

Par exemple, les figures 7.3 et 7.4 nous donnent deux motifs discriminants extraits des classes bleu et rouge de l'*IPR*. Le motif de la figure 7.3 a une fréquence égale à 39.62% dans la classe bleue de l'*IPR* et une fréquence de 16.12% dans la classe rouge de ce même indice.

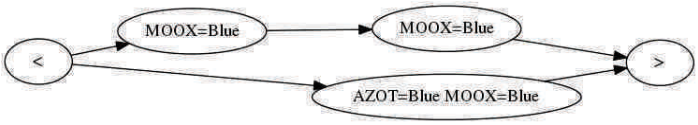


FIGURE 7.3 – Motif discriminant pour la classe bleue de l'*IPR*, avec les fréquences : Bleu=39.62%, Vert=25.34%, Jaune=17.07%, Orange=20.28%, Rouge=16.12%

FIGURE 7.4 – Motif discriminant pour la classe rouge de l'*IPR*, avec les fréquences : Bleu=5.66%, Vert=7.53%, Jaune=5.69%, Orange=4.34%, Rouge=9.67%

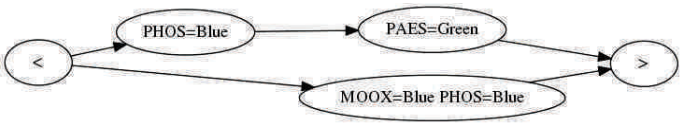


FIGURE 7.5 – Motif discriminant pour la classe bleue de l'*IBGN*, avec les fréquences : Bleu=16.84%, Vert=10.12%, Jaune=5.41%, Orange=1.77%, Rouge=0%

FIGURE 7.6 – Motif discriminant pour la classe rouge de l'*IBGN*, avec les fréquences : Bleu=0%, Vert=0.16%, Jaune=1.23%, Orange=7.8%, Rouge=19.1%

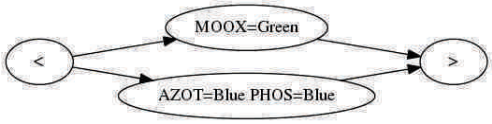


FIGURE 7.7 – Motif discriminant pour la classe bleue de l'*IBD*, avec les fréquences : Bleu=14.95%, Vert=4.58%, Jaune=1.21%, Orange-Rouge=0.74%

FIGURE 7.8 – Motif discriminant pour la classe orange-rouge de l'*IBD*, avec les fréquences : Bleu=0%, Vert=0.12%, Jaune=1.03%, Orange-Rouge=12.68%

Les ensembles de 15 motifs filtrés par classe de chaque indice sont accessibles à cette adresse internet [1] (en couleur). Pour la compréhension, les motifs présentés ici contiennent un maximum de quatre items, mais des motifs plus complexes sont

1. http ://engees-fresqueau.unistra.fr/patterns/patterns.html

fournis sur cette page internet.

7.2.3 Extraction de consensus discriminants

L'extraction et la sélection des motifs d'intérêt ont été présentés dans les deux sections précédentes. Cependant, nous nous sommes également intéressés à l'extraction de consensus discriminants (voir chapitre 5).

Comme pour l'extraction de motifs discriminants, nous avons choisi un seuil de fréquence minimum θ de 10% en faisant varier le seuil de discrimination minimum ρ. La figure 7.9 donne par exemple le consensus discriminant extrait depuis la classe rouge de l'*IBGN* avec $\theta = 10\%$ et $\rho = 2$.

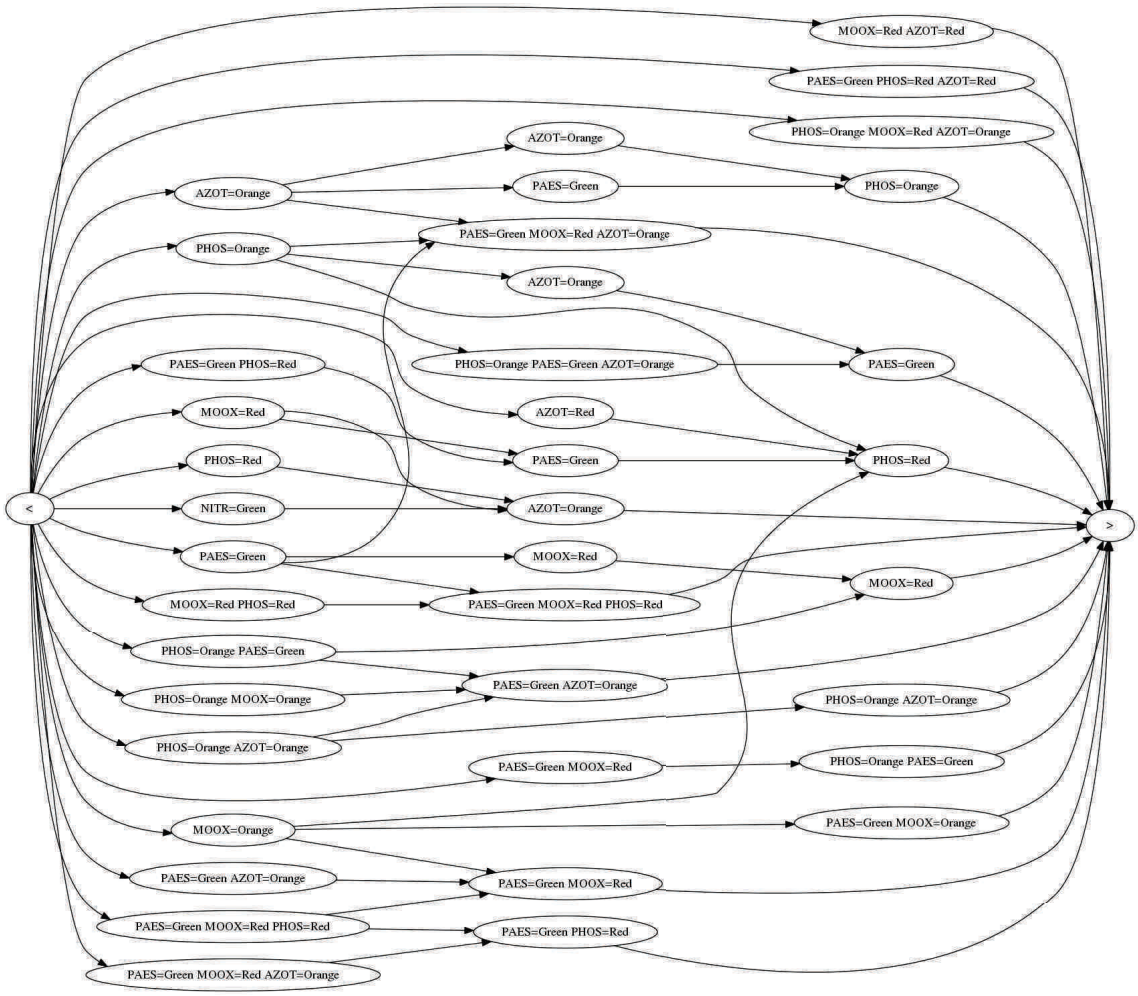


FIGURE 7.9 – Consensus discriminant pour la classe rouge de l'*IBGN* avec $\theta = 10\%$ et $\rho = 2$

L'interprétation de ce consensus se fait de la manière suivante : (1) chaque chemin (motif séquentiels) dans le consensus est au moins supporté par 10% des séquences de la classe rouge de l'*IBGN* et (2) chaque motif séquentiel est également au moins deux fois plus fréquent dans la classe rouge de l'*IBGN* que dans les autres classes de ce même indice biologique.

Regardons maintenant le résultat obtenu si nous augmentons le seuil de discrimination ρ à la valeur de 4 (figure 7.10).

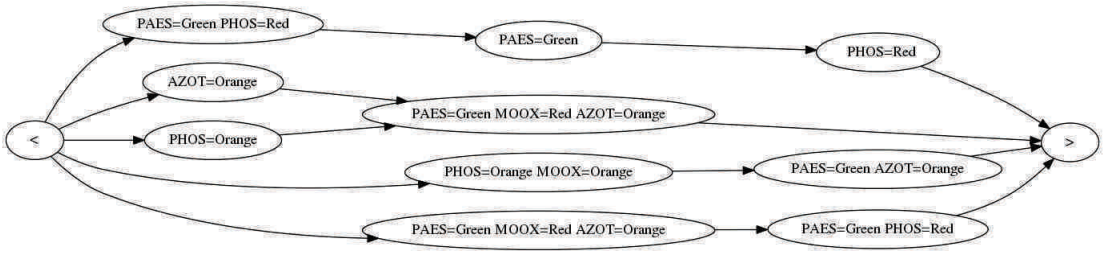


FIGURE 7.10 – Consensus discriminant pour la classe rouge de l'*IBGN* avec $\theta = 10\%$ et $\rho = 4$

Nous observons que ce nouveau consensus est de taille bien plus petite que celui extrait avec $\rho = 2$. En effet, augmenter le seuil de discrimination revient à synthétiser un volume de motifs séquentiels discriminants plus réduit. Ici, chaque motif séquentiel est au moins quatre fois plus supporté dans la classe rouge de l'*IBGN* que dans les autres classes de ce même indice biologique.

Cependant, les discussions avec les hydrobiologistes ont mis en évidence que certaines classes de qualité sont très hétérogènes, en particulier la classe jaune. En effet, celle-ci est représentative d'une qualité modérée de la biologie et est la classe médiane par rapport à toutes les autres classes de qualité. Pour ces classes hétérogènes, il peut alors être intéressant d'identifier les différents groupes de séquences qui contiennent une information proche, ainsi que les différences entre ces groupes.

Nous avons donc appliqué à la classe jaune de l'*IBGN* le processus présenté dans le chapitre 5. Avec la mesure de dissimilarité *Dynamic Sequence Warping*, nous avons réalisé un *clustering* sur les séquences d'une classe de qualité. Ensuite, pour chaque *cluster*, nous extrayons un consensus discriminant. Nous prenons comme exemple la classe jaune de l'*IBGN*. Comme le montre le tableau 7.1, cette classe contient 1282 séquences. Pour l'extraction des *clusters* et après de nombreux tests, nous avons utilisé l'algorithme des k-médoids [KR09] avec le paramètre k (nombre de *clusters*) empiriquement fixé à deux. Le tableau 7.2 nous donne le nombre de séquences regroupées dans chaque *cluster*.

ID	Nombre de séquences
Cluster 1	320
Cluster 2	962

TABLE 7.2 – Nombre de séquences dans chaque *cluster*

Comme nous le voyons, le nombre de séquences dans chaque *cluster* est très déséquilibré. Par exemple, le second *cluster* contient 962 séquences alors que le premier en contient 320. Pour extraire des consensus discriminants facilement interprétables, nous avons fait empiriquement varier le seuil de support minimum θ et le seuil de discrimination minimum ρ. Les figures 7.11 et 7.12 donnent les consensus discriminants pour chaque *cluster*.

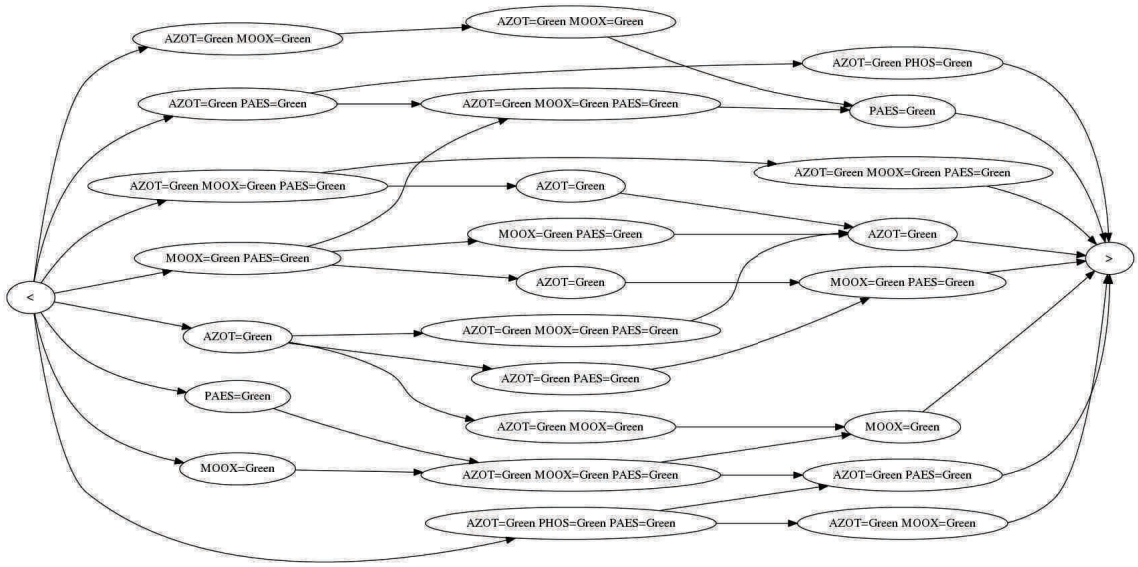


FIGURE 7.11 – Consensus discriminant pour le *cluster* 1 avec $\theta = 40\%$ et $\rho = 4$

En analysant ces deux consensus, nous observons que celui du *cluster* 1 est composé de l'ensemble d'items $\{AZOT^{Vert}, MOOX^{Vert}, PHOS^{Vert}, PAES^{Vert}\}$, alors que celui du *cluster* 2 est composé de l'ensemble d'items $\{AZOT^{Bleu}$, $MOOX^{Bleu}, PHOS^{Vert}, PAES^{Vert}\}$. La différence entre ces deux *clusters* est relative aux valeurs des paramètres AZOT et MOOX.

Dans cette section, nous avons présenté les résultats obtenus sur le jeu de données Fresqueau avec chaque méthode. Ces résultats ne sont pas exhaustifs et un volume plus important d'expérimentations a été effectué. Dans cette optique, la prochaine section présente un logiciel de visualisation destiné aux experts pour leur permettre

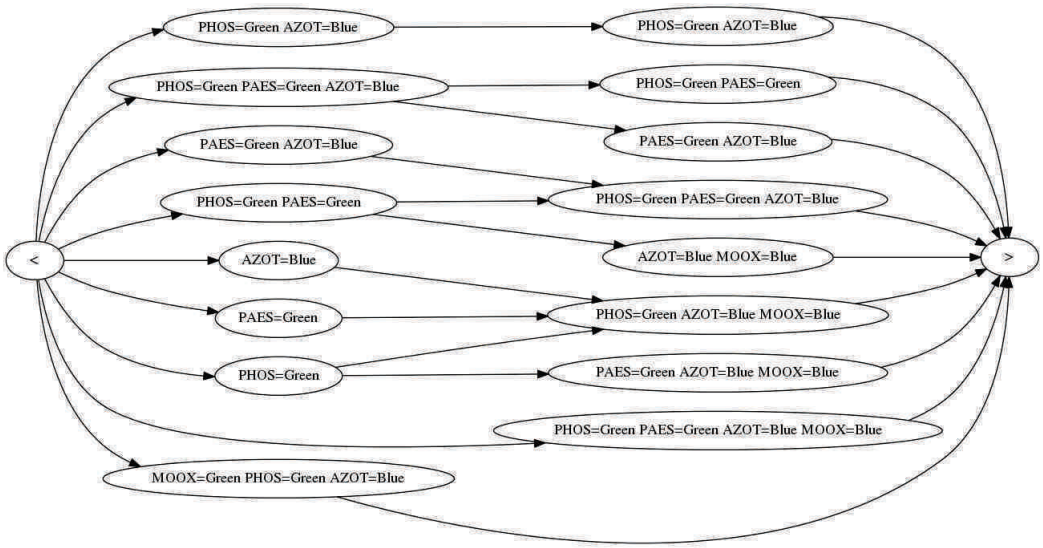


FIGURE 7.12 – Consensus discriminant pour le *cluster* 2 avec $\theta = 10\%$ et $\rho = 1,3$

de sélectionner un sous-jeu de données et de choisir les différents paramètres en entrée des méthodes.

7.3 Intégration dans un logiciel de visualisation

Cette section présente l'intégration des différentes méthodes dans un logiciel de visualisation. Ce logiciel a été conçu dans l'objectif de combiner nos techniques de fouille de motifs partiellement ordonnés avec des techniques visuelles et interactives. Cet outil a été conçu dans le cadre du projet Fresqueau, en collaboration avec l'équipe Tatoo du LIRMM et est destiné aux hydrobiologistes. Il est composé de plusieurs vues qui vont de l'utilisation de cartes géographiques, à la projection de *clusters* de stations de prélèvements et à la visualisation de motifs partiellement ordonnés.

L'analyse visuelle est une science de raisonnement analytique facilitée par des interfaces visuelles interactives [TC05]. Plus précisément, « l'analyse visuelle combine des techniques d'analyses automatiques avec des visualisations interactives pour une compréhension efficace, un raisonnement et une prise de décision sur la base de très larges jeux de données complexes » [KAF+08]. Les techniques d'analyses automatiques incluent les statistiques, les mathématiques, la représentation des connaissances, leur gestion et leur découverte.

Actuellement, de nombreux outils d'analyse basés sur les systèmes d'informa-

tion géographiques (SIG) et qui incluent des statistiques et des graphiques ont été proposés (comme ArcGIS Geostatistical Analyst [Arc]). Cependant, ces outils sont principalement basés sur des techniques statistiques et n'utilisent pas de méthodes de fouille de données temporelles. L'objectif est donc de faciliter l'utilisation des méthodes proposées dans cette thèse sur des données hydrobiologiques, pour fournir des connaissances nouvelles et différentes de celles produites par des outils statistiques.

Questions générales des experts

Dans cette section, nous listons les questions intéressant les hydrobiologistes et discutées lors de réunions. Elles ont permis de définir un ensemble de besoins applicatifs.

Nous avons identifié cinq types de questions exprimées en termes généraux : (1) dans un objectif précis, quel jeu de données utiliser ? ; (2) pour une zone d'étude précise, quelles sont les stations de prélèvements qui ont un comportement similaire, ou proche ? ; (3) à l'inverse, pour un sous ensemble de stations de prélèvements qui ont un comportement similaire, où sont-elles localisées ? ; (4) pour une station de prélèvements et un indice biologique, quelle est la plus mauvaise classe de qualité mesurée pour cet indice parmi tous les prélèvements de cette station ? et (5) pour un sous ensemble de stations, quelles sont les principaux liens entre les valeurs de classe de qualité des indices biologiques et des paramètres physico-chimiques ?

Besoins de l'outil

À partir de ces différentes questions, nous avons identifié sept besoins auquel l'outil doit répondre :

R1 - Sélectionner un sous jeu de données (années, mois, indices biologiques et paramètres physico-chimiques), pour répondre à la question 1.

R2 - Visualiser et naviguer à travers les stations de prélèvements à partir de leur localisation dans l'espace, pour répondre aux questions 3 et 4.

R3 - Visualiser et naviguer à travers les stations de prélèvements à partir de leur comportement similaire, pour répondre aux questions 2 et 4.

R4 - Sélectionner un groupe de stations de prélèvements à partir de leur localisation dans l'espace, pour répondre aux questions 2 et 5.

R5 - Sélectionner un groupe de stations de prélèvements à partir de leur comportement similaire, pour répondre aux questions 3 et 5.

R6 - Visualiser les classes de qualité des indices biologiques, pour répondre à la question 4.

R7 - Extraire et visualiser les motifs temporels mettant en lien les valeurs de classe de qualité des indices biologiques et des paramètres physico-chimiques, pour répondre à la question 5.

7.3.1 Les différentes vues du système

Nous présentons tout d'abord les trois vues qui sont utilisées dans notre outil. Les différentes vues sont : (1) la vue géographique avec navigation sur une carte ; (2) la vue *clustering* avec affichage des stations de prélèvements en fonction de leur comportement, et (3) la vue des motifs temporels pour analyser les motifs extraits à partir des deux vues précédentes.

Vue géographique

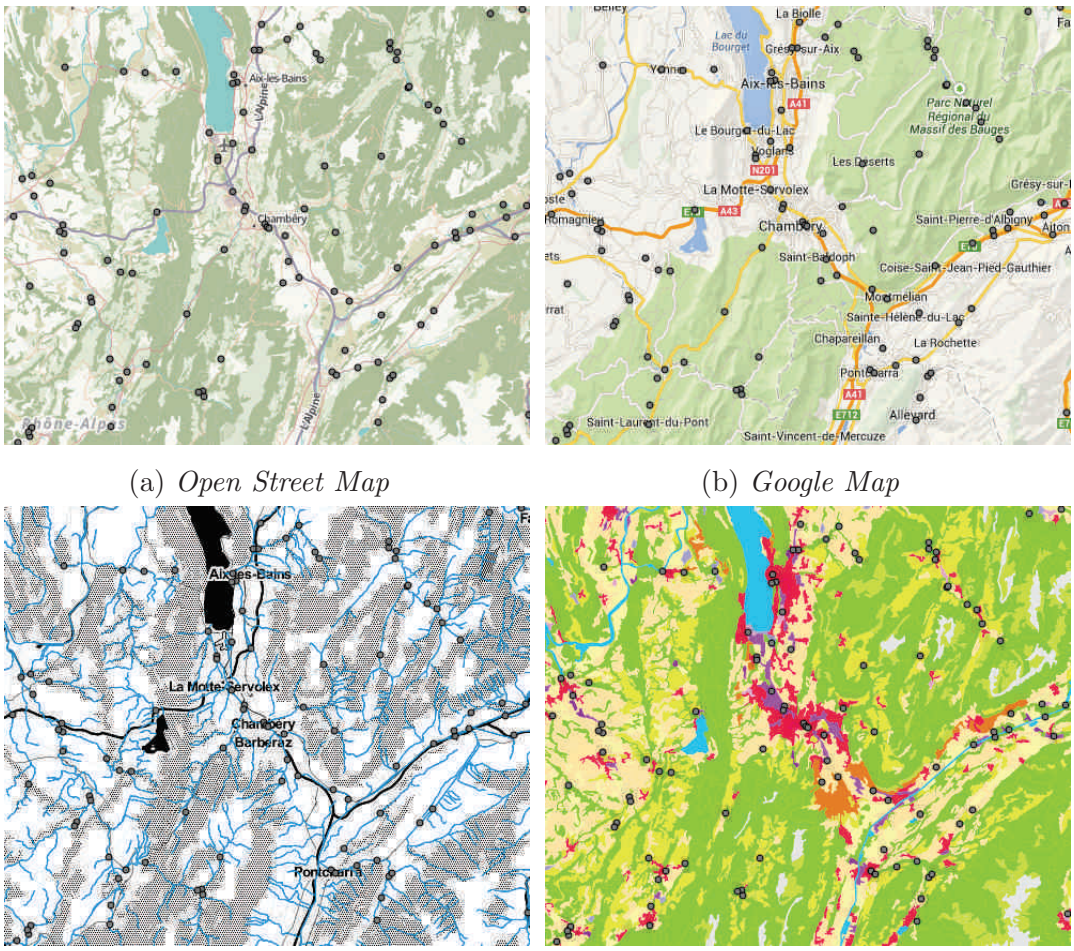
Dans cette sous-section, nous décrivons la vue qui montre la localisation géographique des stations de prélèvements (besoin R2).

Nous avons choisi le service web google maps comme solution technique pour afficher les stations de prélèvements sur un contexte géographique. La projection utilisée dans la base de données pour représenter les coordonnées spatiales est la projection conique conforme de Lambert, qui est utilisée par l'institut national géographique (IGN). Pour être utilisées sur des cartes telles que les Google Maps, les coordonnées des stations de prélèvements ont été converties en coordonnées conformes au système géodésique standard mondial (WGS84). Plusieurs autres fonds de cartes, extérieurs à Google, sont également fournis à l'utilisateur comme le relief ou bien l'occupation du sol.

La figure 7.13 donne quatre exemples de cartes disponibles dans l'application : une vue Open Street Map (figure 7.13a), une vue Google Map (figure 7.13b), une vue du réseau hydrographique (figure 7.13c) ou bien encore une vue de l'occupation du sol utilisant Corine Land Cover (figure 7.13d). D'autres type de cartes sont disponibles.

Vue *clustering*

Pour répondre au besoin R3, nous avons besoin de regrouper les stations de prélèvements qui ont un comportement similaire ou proche, i.e. les items en commun



(a) *Open Street Map* (b) *Google Map*

(c) *Réseau hydrographique* (d) *Corine Land Cover*

FIGURE 7.13 – Différentes vues de la région de Chambéry

dans leur séquences. Nous utilisons une approche par *clustering*, qui est parfaitement adaptée à une telle tâche.

Nous avons utilisé la mesure de dissimilarité *Dynamic Sequence Warping* (voir chapitre 5) dans un algorithme de *clustering* hiérarchique ascendant [KR09] pour regrouper les stations de prélèvements. En sortie de l'algorithme, nous obtenons un ensemble de *clusters* qui sont alors projetés sur la vue *clustering*. Cette projection est faite en quatre étapes qui ont pour objectif d'améliorer la visualisation en fonction des besoins des hydrobiologistes :

1. Avec *Dynamic Sequence Warping*, une matrice de dissimilarité entre les séquences des stations de prélèvements est calculée. Elle est utilisée pour positionner les stations dans un espace à deux dimensions en utilisant un

algorithme de Multidimensional Scaling.

2. Les *clusters* sont représentés par des cercles positionnés au barycentre des stations qu'ils contiennent.
3. Des méthodes de détection de collisions sont utilisées pour séparer le chevauchement entre plusieurs *clusters*.
4. Dans chaque *cluster*, les stations sont réparties de manière à mieux occuper l'espace du cercle correspondant au *cluster*.

Cette approche de visualisation de *clusters* est le travail de Pierre Accorsi et est décrite en détail dans [AFS+14].

La vue finale inclut également un zoom pour faciliter l'exploration et la navigation. La figure 7.14 montre le résultat obtenu sur un exemple réel.

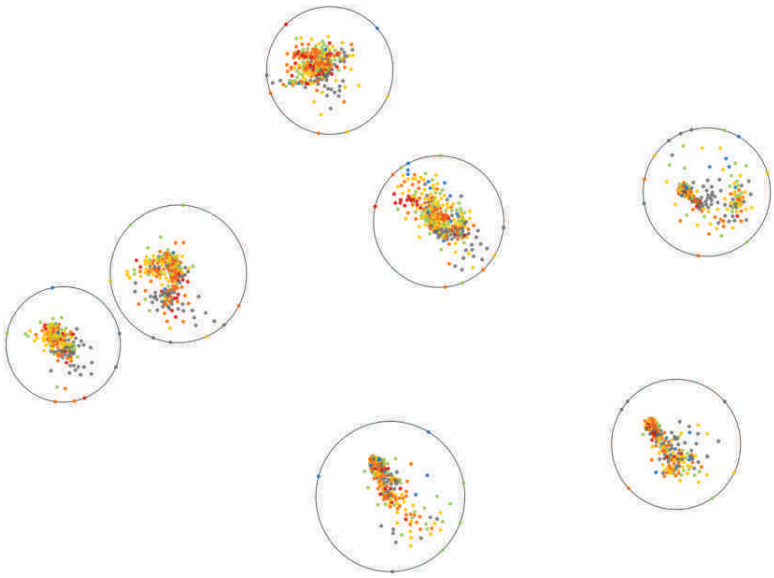


FIGURE 7.14 – Vue des stations de prélèvements groupées en fonction de la distance entre leurs séquences

Vue analyse des motifs

Ici, nous décrivons la vue qui montre les motifs partiellement ordonnés clos extraits, et qui répond au besoin R7. Cette vue est utilisée après la sélection du jeu de données et l'application du processus d'extraction. L'objectif est de pouvoir extraire des motifs discriminants par classe de qualité biologique en suivant le protocole présenté dans le chapitre 6, c'est-à-dire en sélectionnant un indice biologique et un

intervalle de temps, qui va servir à générer des séquences de qualité biologique à partir des séquences des stations.

Cette vue est divisée en deux panneaux. Un premier panneau montre la liste des motifs partiellement ordonnés extraits. Chaque élément de la liste montre un identifiant associé à un motif, sa taille (en nombre de sommets), la classe de l'indice biologique dans lequel le motif est discriminant et la valeur de discrimination pour évaluer à quel point le motif est fréquent dans cette classe de qualité comparé aux autres classes. La sélection d'un élément de la liste affiche sur le second panneau le motif correspondant.

Comme nous l'avons vu, un motif partiellement ordonné est un graphe dirigé acyclique. Nous avons donc choisi d'utiliser les techniques utilisées en visualisation de graphes. Chaque motif partiellement ordonné est dessiné par couches successives en utilisant l'algorithme de Sugiyama [STT81]. Cet algorithme a l'avantage de mettre en évidence l'aspect séquentiel des motifs partiellement ordonnés. Pour cela, nous avons utilisé l'implémentation de l'algorithme fournie par la librairie graphviz [2] [EGK+03].

Appliqués aux données du projet Fresqueau, les motifs partiellement ordonnés contiennent comme items des macro-paramètres physico-chimiques avec une couleur de classe de qualité associée. Pour les besoins des experts, les items sont visualisés avec le nom du macro-paramètre physico-chimique et avec la couleur qui correspond à sa classe de qualité. Les motifs fournis sur cette page internet [3] ont été générés selon ce procédé.

7.3.2 Interaction

Cette section discute de l'interaction entre l'utilisateur et le système de visualisation.

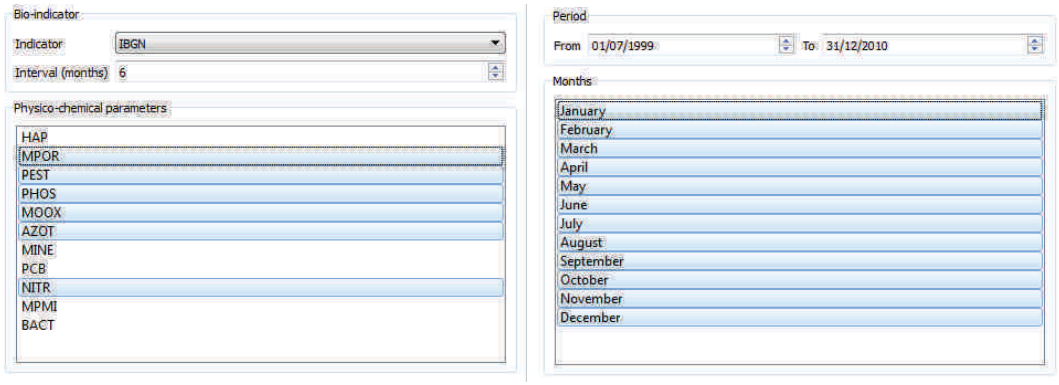
Filtres sur le jeu de données

Premièrement, l'utilisateur doit sélectionner un jeu de données. Ce jeu de données peut ensuite être filtré. Les différents paramètres du filtres sont : un intervalle d'années à choisir, un intervalle de mois, un ensemble d'indices biologiques et de macro-paramètres physico-chimiques à analyser (voir besoin R1). Les intervalles de temps permettent de considérer différentes granularités temporelles. Par exemple, il est possible de sélectionner les prélèvements qui vont de 2004 à 2007 et qui concernent

2. www.graphviz.com
3. http ://engees-fresqueau.unistra.fr/patterns/patterns.html

spécifiquement la période qui va de Janvier à Avril. Les paramètres sélectionnés sont ensuite affichés dans un panneau spécifique.

La figure 7.15 montre les diverses possibilités et combinaisons de filtres à appliquer sur le jeu de données.



(a) (b)

FIGURE 7.15 – Sélection (a) des paramètres biologiques et physico-chimiques et (b) des intervalles de temps

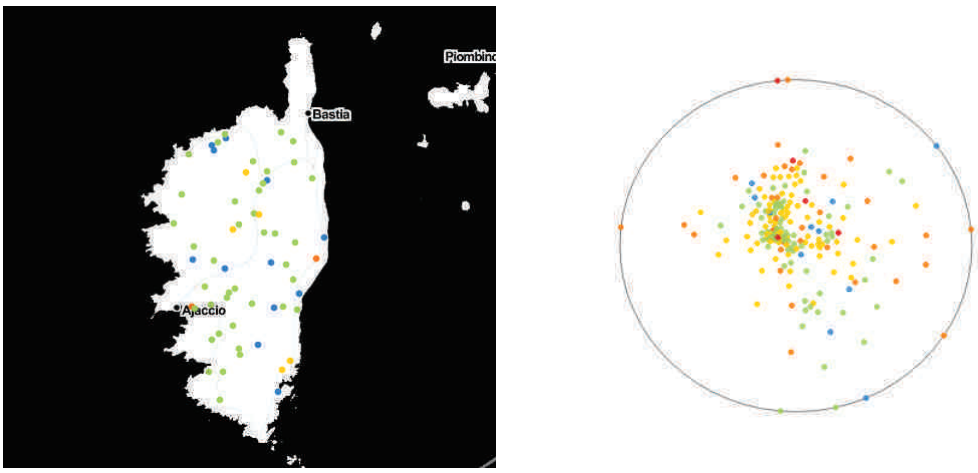
Coloration des stations de prélèvements

Une couleur peut être affectée à chaque cercle représentant une station. Celle-ci correspond à la pire valeur de classe de qualité (discussion avec les experts) d'un indice biologique sélectionné au préalable. Par exemple, si l'indice biologique sélectionné est l'IBGN, un point coloré en jaune signifie que sur l'intervalle de temps sélectionné (filtre), la station associée a eu au pire une mesure correspondant à une qualité d'IBGN jaune parmi tous les prélèvements. Cette coloration des points de stations de prélèvements s'effectue à la fois sur la vue *clustering* ainsi que sur la vue géographique. L'utilisateur peut changer l'indice biologique sélectionné grâce à un bouton dédié (voir besoin R6).

La figure 7.16 montre la coloration des stations de prélèvements sur les différentes vues de l'outil.

Sélection des stations à analyser

Ensuite, avant d'extraire les motifs partiellement ordonnés clos, l'utilisateur doit sélectionner un ensemble de stations de prélèvements à analyser. Cette sélection peut



(a) (b)

FIGURE 7.16 – Coloration des points des stations (a) sur la vue géographique et (b) sur la vue *cluster*

se faire en passant par la vue *clustering* et par la vue géographique (voir besoins R4 et R5).

Sélectionner les stations peut se faire de manière individuelle en cliquant dessus ou bien plusieurs stations peuvent être sélectionnées en une seule fois en utilisant un outil de type lasso. Les stations sélectionnées sont alors mises en évidence en changeant la couleur de leur bordure en violet (à la place de noir).

La figure 7.18 montre la sélection des stations de prélèvements sur les différentes vues de l'outil.

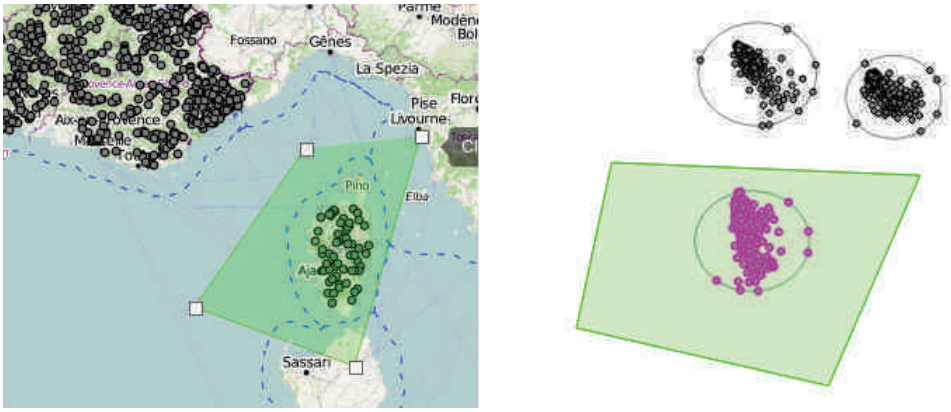
De plus, lorsque la sélection est effectuée sur une des vues, l'autre est automatiquement mise à jour pour correspondre exactement à la même sélection.

La figure 7.18 montre cette fonctionnalité où les stations sélectionnées sur la vue *clustering* (sélection du *cluster* du bas) sont également sélectionnées sur la vue géographique.

Comme demandé par les experts, nous conservons un historique de l'ensemble des motifs extraits durant une session.

Extraction motifs et leur analyse

Pour finir, les stations sélectionnées sont utilisées en paramètre de l'algorithme d'extraction des motifs partiellement ordonnés. Un bouton permet l'accès à un formulaire pour la sélection des paramètres de l'algorithme, comme la fréquence minimale ou bien les classes de qualité qui seront utilisées pour extraire les motifs



(a) (b)

FIGURE 7.17 – Sélection de type lasso (a) sur la vue géographique et (b) sur la vue *cluster*

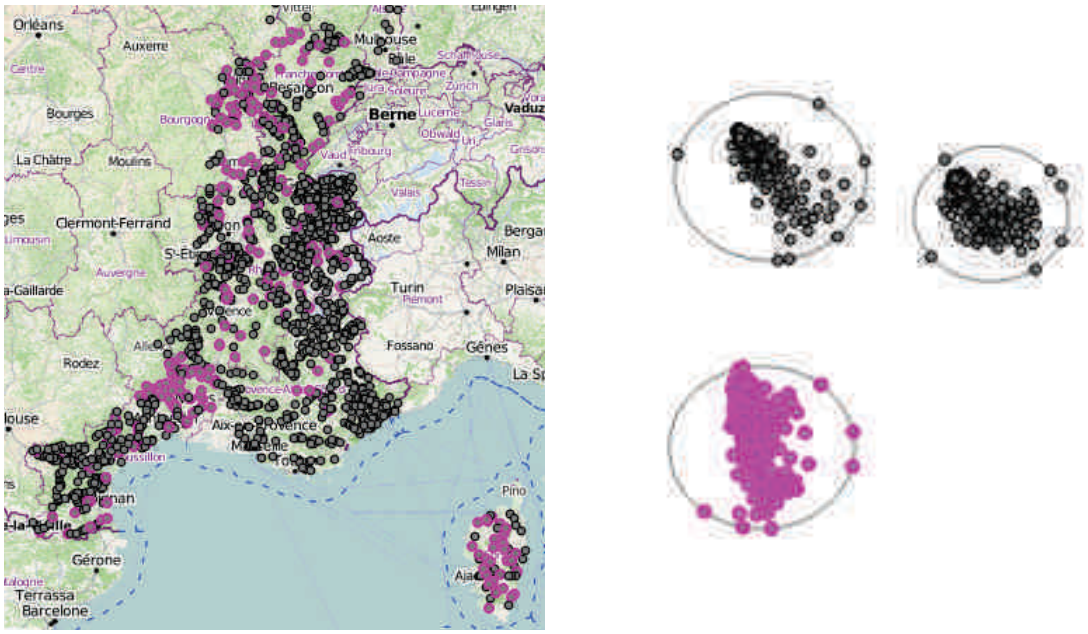
partiellement ordonnés discriminants. Par exemple, il est possible d'extraire avec une fréquence minimale de 30% sur l'IBGN bleu et de 20% sur l'IBGN orange et rouge, sans considérer la classe verte et la classe jaune. L'extraction est ainsi totalement personnalisable.

7.4 Discussion

Les résultats présentés ci-dessus ainsi que le logiciel de visualisation ont été fournis aux hydrobiologistes. Pour rappel, ces derniers sont intéressés par le lien qui peut exister entre les paramètres physico-chimiques et les paramètres biologiques. Dans cette section, nous discutons : (1) de l'analyse et de l'interprétation des résultats par les hydrobiologistes partenaires dans le projet Fresqueau et (2) de l'intérêt des méthodes proposées dans cette thèse pour le traitement des données hydrobiologiques. Pour une meilleure compréhension de cette section, nous rappelons dans le tableau 7.3 la relation entre les couleurs des paramètres discrétisés et les catégories de qualité.

Tout d'abord, l'analyse des motifs a permis de retrouver la connaissance du domaine. Avant de chercher à identifier de nouvelles connaissances, cette première étape est importante pour valider l'utilisation des motifs partiellement ordonnés dans ce contexte.

Par exemple, l'indice biologique *IBGN* a été développé pour considérer la problématique de la pollution par les matières organiques (paramètre physico-



(a) (b)

FIGURE 7.18 – Synchronisation (a) de la vue géographique et (b) de la vue *cluster*

Couleur	**Qualité**
Bleu	Très bon
Vert	Bon
Jaune	Moyen
Orange	Mauvais
Rouge	Très mauvais

TABLE 7.3 – Correspondance entre les couleurs et les catégories de qualité

chimique MOOX). L'analyse des motifs extraits a en effet permis de retrouver cette connaissance : le MOOX en qualité bleue apparaît fréquemment dans les motifs discriminants pour l'*IBGN* bleu et un MOOX en qualité rouge apparaît fréquemment dans les motifs discriminants pour l'*IBGN* rouge.

Concernant les diatomées (indice biologique *IBD*), l'impact du paramètre physico-chimique PHOS sur les diatomées a déjà été étudié [KPW95, Cor99] et est connu. La qualité biologique pour les diatomées est souvent mauvaise lorsque une forte concentration de phosphates est mesurée. Cette connaissance se retrouve dans les motifs puisque comme pour l'*IBGN* et le MOOX, un PHOS bleu se retrouve

fréquemment dans les motifs discriminants de l'*IBD* bleu et un PHOS rouge se retrouve fréquemment dans les motifs discriminants de l'*IBD* rouge.

Pour l'indice biologique *IPR*, les motifs montrent qu'une très bonne qualité biologique (communautés de poissons idéales) a besoin d'une très bonne qualité physico-chimique de l'eau. En effet, il est connu que les communautés poissons sont sensibles à la pollution due aux matières organiques ainsi que les matières azotées. D'autre part, l'analyse de certains motifs montre qu'une mauvaise qualité pour les communautés de poissons peut exister même si la qualité des paramètres physico-chimiques est bonne. Une raison possible est la présence de conditions hydromorphologiques mauvaises (canal bétonné, barrage, etc.). En effet, les conditions hydromorphologiques ont un impact important sur l'habitat des poissons.

Concordance pour les classes de qualité intermédiaires

Cependant, bien que les classes extrêmes entre biologie et physico-chimie soient concordantes (qualité bleue et rouge), l'analyse a fait apparaître que les classes intermédiaires ne le sont pas forcement. Nous illustrons cet aspect par l'analyse de la concordance entre classes de qualité de l'*IBD* et du PHOS. L'analyse des motifs a permis d'observer que pour chaque classe de qualité de l'*IBD*, une classe de qualité du macro-paramètre PHOS est généralement liée. Le tableau 7.4 résume cette observation où nous voyons que chaque valeur de classe de qualité biologique ne correspond pas exactement avec la valeur des classes de qualité physico-chimiques. En effet, la valeur des classes de qualité de l'*IBD* décroit plus rapidement que la valeur des classes de qualité du PHOS. Par exemple, un *IBD* jaune est lié à un PHOS vert et un *IBD* orange est lié à un PHOS jaune. Un IBD rouge est en général lié à un PHOS orange ou rouge, mais il est également fortement lié à la combinaison d'un PHOS rouge avec un AZOT rouge et un MOOX rouge dans un même motif. Dans le tableau, la dernière ligne signifie que la combinaison de ces trois valeurs dans un même motif ne se retrouve que dans le cas de l'*IBD* rouge.

IBD	**Physico-chimie**
Jaune	PHOS^{Vert}
Orange	PHOS^{Jaune}
Rouge	PHOS^{Orange} ou PHOS^{Rouge}
Fortement rouge	PHOS^{Rouge} et AZOT^{Rouge} et MOOX^{Rouge}

TABLE 7.4 – Correspondance entre les classes de qualité de l'IBD et celle du PHOS

Pour l'indice *IBD*, nous observons donc que : (1) le statut biologique est très sensible à la perturbation chimique de l'eau et (2) la combinaison de mauvaises valeurs (rouge) de plusieurs macro-paramètres physico-chimiques produit une très forte dégradation de cette dimension biologique. Cela nous amène à l'importance de : (1) déterminer de nouveaux intervalles pour les classes de qualité, pour faire correspondre avec plus d'exactitude la biologie avec la physico-chimie et (2) évaluer avec plus de précision l'impact généré par la combinaison de plusieurs paramètres physico-chimiques.

Ce même phénomène se retrouve entre les classes de qualité de l'*IBGN* et du MOOX.

De nouveaux liens entre biologie et physico-chimie

Également, les motifs ont permis de mettre en évidence des liens entre biologie et physico-chimie qui sont actuellement supposés et moins connus.

Par exemple, l'impact des matières phosphorées (paramètre physico-chimique PHOS) sur l'indice biologique *IBGN* est un fait moins établi. Pourtant, les motifs montrent un lien très fort entre les classes de qualité du PHOS et de l'*IBGN* (voir page internet [4]). Une telle correspondance est intéressante puisqu'elle pourrait signifier que les sources de pollution aux phosphores ont un impact significatif sur les macro-invertébrés (*IBGN*).

Utilité de la connaissance experte pour la discrétisation des données

Les méthodes de motifs et de consensus partiellement ordonnés sont généralement sensibles au processus de discrétisation. Le choix des intervalles détermine fortement la connaissance extraite. Mais un tel processus est également très bien adapté pour l'utilisation de la connaissance du domaine comme remplacement des variables numériques. La discrétisation des données à l'aide de la connaissance experte (standards SEQ-eau et AFNOR) s'est avérée utile.

En effet, les valeurs continues ne sont pas toujours faciles à analyser. Des valeurs discrétisées grâce à la connaissance experte permettent, en amont, d'améliorer la robustesse et la facilité d'interprétation des résultats. La raison est que les hydrobiologistes sont habitués depuis plusieurs années à manipuler ce type de connaissance. La restitution et la compréhension des approches basées sur les motifs temporels en sont ainsi simplifiées.

4. http ://engees-fresqueau.unistra.fr/patterns/patterns.html

Intérêt des motifs et des consensus partiellement ordonnés pour les données environnementales

Dans ce chapitre, nous avons présenté l'application des motifs et des consensus partiellement ordonnés pour relier la physico-chimie avec la biologie. Cependant, une telle approche est certainement utile pour explorer des données concernant d'autres problèmes hydrobiologiques ou bien environnementaux.

Sur ce type de données temporelles, il est également possible d'appliquer des méthodes basées sur les motifs séquentiels comme alternative aux motifs partiellement ordonnés. Comparé à ces derniers, les motifs séquentiels sont représentés par une liste plate de valeurs symboliques. Les motifs partiellement ordonnés ont l'avantage de prendre en compte la coexistence d'éléments qui ne sont pas temporellement ordonnés. Par exemple, prenons le motif donné par la figure 7.7, AZOT^{Bleu} et PHOS^{Bleu} sont fréquemment observés en même temps et ils coexistent avec MOOX^{Vert}. Une approche par motifs séquentiels extrait deux motifs séquentiels différents : $\langle(\text{AZOT}^{Bleu}, \text{PHOS}^{Bleu})\rangle$ et $\langle(\text{MOOX}^{Vert})\rangle$. Cette division de la connaissance est un inconvénient car la coexistence des éléments permet aux hydrobiologistes de mieux identifier les paramètres physcio-chimiques, qui associés avec d'autres paramètres, renforcent potentiellement l'impact sur la biologie.

Les méthodes appliquées sont particulièrement adaptées à l'identification d'observations temporelles composées de plusieurs variables. En effet, elles sont capables de capturer des groupes de variables qui sont observées en même temps, qui se suivent temporellement ou bien sans ordre temporel entre ces variables. Ainsi, les motifs et consensus fournissent une information sur la fréquence d'apparition d'une ou plusieurs variables. Elles sont aussi adaptées dans le cas de jeux de données composés de plusieurs centaines ou milliers d'instances. Il est alors possible d'utiliser en premier lieu ces méthodes pour obtenir une vision globale sur les informations présentes dans les données, et dans un second temps d'utiliser par exemple des approches statistiques multivariées [LL12]. L'idée est d'affiner la connaissance en analysant par exemple seulement les paramètres fréquemment observés dans les motifs. Les méthodes par extraction de motifs peuvent donc être parfaitement complémentaires aux méthodes statistiques.

7.5 Synthèse

Ce chapitre présente l'application des méthodes proposées dans les chapitres 3, 4 et 5 sur les données hydrobiologiques du projet Fresqueau.

Les données traitées sont relatives à trois jeux de données correspondant aux indices biologiques *IBGN*, *IBD* et *IPR*. Les jeux de données ont été générés en suivant le processus présenté dans le chapitre 6. Les différentes méthodes ont permis d'extraire un volume intéressant d'informations concernant les différentes classes de qualité biologiques de chaque indice.

Afin de rendre les méthodes accessibles aux experts et pour faciliter leur utilisation, elles ont été implantées dans un logiciel de visualisation qui permet :

- de filtrer un jeu de données à partir : (1) d'une sélection de stations de mesures sur différentes vues (vue géographique ou bien vue *clustering*) et (2) de sélectionner les paramètres à analyser au travers d'un formulaire.
- d'appliquer les méthodes sur le jeu de données filtré et de visualiser les différents motifs obtenus, par classe de qualité.

L'interprétation des résultats par les hydrobiologistes a mis en évidence l'intérêt d'utiliser de nouvelles approches telles que les motifs partiellement ordonnés pour le traitement de données environnementales temporelles. En effet, les motifs ont permis de retrouver les connaissances du domaine mais également d'amener de nouvelles perspectives, en particulier l'importance d'évaluer avec plus de précision l'impact de la combinaison de plusieurs paramètres physico-chimiques sur la biologie, ainsi que l'intérêt de définir de nouveaux intervalles de qualité plus adaptés à cette problématique.

CHAPITRE 8

Conclusion et perspectives

Les données massives désignent des ensembles de données tellement volumineux qu'ils sont difficiles à exploiter avec des outils classiques. De nombreux challenges sont associés à ces données massives tels que leur capture, leur stockage, leur exploration, leur analyse et leur visualisation. Dans le cadre de cette thèse, nous nous sommes intéressés à l'exploration d'un type de données particulières, les données séquentielles et à un domaine d'application particulier, l'hydrobiologie. Bien qu'appliquées à ce domaine, nos méthodes sont bien sur généralisables et applicables à tout domaine où les données sont séquentielles, comme les données génomiques, temporelles ou encore textuelles.

L'exploration de bases de données séquentielles reste une problématique difficile. Bien qu'il existe de nombreux travaux dans la littérature, des verrous scientifiques demeurent :

— Certains types de motifs temporels complexes, comme les motifs partiellement ordonnés étudiés dans cette thèse, nécessitent de naviguer dans un espace de recherche important et restent à ce jour difficiles à extraire ;
— Les informations extraites sont souvent nombreuses. La navigation dans les résultats et leur restitution à l'utilisateur restent difficiles ;
— Il existe peu d'approches qui tentent de résumer et condenser les données pour les rendre interprétables par un utilisateur, quitte à perdre de l'information, notamment dans les bases de données de séquences d'itemsets.

Les propositions réalisées dans cette thèse apportent des solutions pour ces trois verrous.

Les travaux existants se focalisent principalement sur la fouille de motifs séquentiels. De nombreux algorithmes ont été développés pour résoudre les difficultés d'extraction des motifs séquentiels, alors qu'il existe d'autres manières d'explorer ces bases de données. C'est ainsi que nous nous sommes intéressés à un autre type de motifs, les motifs partiellement ordonnés, qui permettent de capturer une information

partielle sur l'ordre entre les éléments d'un ensemble de séquences, contrairement aux motifs séquentiels qui représentent un ordre total. Ces motifs ont l'avantage de résumer des ensembles de motifs séquentiels ce qui permet de condenser l'information, qui reste interprétable et visualisable pour les humains. Cependant, l'extraction de tels motifs reste peu étudiée dans les bases de données de séquences. Les méthodes existantes ne permettent pas à la fois d'extraire l'ensemble complet des motifs partiellement ordonnés clos et d'être applicables sur n'importe quel type de bases de données de séquences.

L'étude des motifs partiellement ordonnés a donc été le cœur de cette thèse et nous avons développé différentes approches pour résumer efficacement les bases de données de séquences :

— Un algorithme pour l'extraction de motifs partiellement ordonnés clos. Ces derniers permettent de résumer des ensembles de motifs séquentiels clos.
— Une méthode pour filtrer k motifs partiellement ordonnés clos selon plusieurs critères d'intérêt pour l'utilisateur.
— Une approche pour extraire un résumé d'une base de données, appelé consensus partiellement ordonné, comme alternative aux motifs partiellement ordonnés clos.

8.1 Contributions

OrderSpan pour l'extraction de motifs partiellement ordonnés clos

Comme souligné précédemment, l'extraction des motifs partiellement ordonnés clos reste peu étudiée [CG05, PWL+06] et les approches existantes souffrent de certaines limites. Pour corriger ces limites, nous avons proposé l'algorithme *OrderSpan*, qui permet d'extraire l'ensemble complet des motifs partiellement ordonnés clos dans toutes les bases de données de séquences. *OrderSpan* fonctionne en deux phases :

1. À partir d'un support minimum θ, l'algorithme extrait tout d'abord, en utilisant les propriétés des préfixes fréquents [PHMA+04], des motifs partiellement ordonnés qui correspondent à des sous-arbres de l'espace de recherche des motifs séquentiels. Chaque sous-arbre correspond à un sous-ensemble de séquences $\mathcal{S}$ de la base de données tel que $|\mathcal{S}| \geq \theta$.
2. Ensuite, chaque motif partiellement ordonné clos est transformé en utilisant les propriétés des suffixes fréquents. L'opération permet de réduire le nombre de sommets et de supprimer la redondance dans les chemins.

Le processus a été optimisé en adaptant la propriété sur l'équivalence des bases projetées [YHA03]. En effet, comme notre approche se base sur des méthodes qui ont fait leurs preuves en fouille de motifs séquentiels (paradigme *Pattern-Growth*), il est possible d'adapter à notre cas les optimisations existantes.

Les expérimentations mettent en évidence l'efficacité de la version optimisée d'*OrderSpan* par rapport à la version non optimisée. Dans un but de comparaison, nous avons également adapté notre algorithme pour extraire le même ensemble de motifs que l'approche décrite dans [CG05], qui effectue un post-traitement sur un ensemble de motifs séquentiels clos. Les résultats montrent qu'extraire directement les motifs partiellement ordonnés est plus efficace.

Filtrer k motifs avec *Pattern-Balanced*

Nous nous sommes également intéressés à la sélection de quelques motifs d'intérêt dans un ensemble volumineux de motifs, en particulier dans un contexte d'extraction de motifs dans des bases de données compartimentées en classes. Nous avons proposé l'algorithme *Pattern-Balanced*. Cet algorithme permet d'extraire un sous-ensemble de k motifs selon une combinaison de plusieurs critères d'intérêt pour les utilisateurs, où chaque critère est d'importance équivalente. Dans notre cas, nous avons identifié trois critères qui sont la fréquence, la discrimination ainsi que la non-redondance des motifs.

Les différents tests effectués sur la base de données *Fresqueau* et les discussions avec les experts hydrobiologistes, ont montré que la sélection d'une quinzaine de motifs avec l'algorithme *Pattern-Balanced* permet de résumer efficacement plusieurs milliers de motifs. Le nombre de motifs extraits n'est plus une limite puisque, comme montré dans le chapitre 4, l'approche permet de filtrer directement quelques motifs d'intérêt pour l'utilisateur.

Extraction de consensus comme alternative à la fouille de motifs

L'extraction de motifs partiellement ordonnés clos est utile pour identifier l'information commune à des sous-ensembles de séquences d'une base de données. Cependant, en filtrant quelques motifs d'intérêt parmi des milliers, il est possible d'oublier des informations intéressantes. Pour pallier cet écueil, nous avons alors proposé une approche qui vise à extraire un consensus pour résumer, selon un support minimum θ, l'ensemble des motifs séquentiels qui sont supportés par des sous-ensembles de séquences d'une base de données. Dans ce cadre, un consensus est une structure

partiellement ordonnée où chaque chemin est un motif séquentiel.

De plus, nous avons étendu l'approche au cas de bases de données composées de classes. Pour cela, nous avons introduit un seuil de discrimination minimum ρ tel que chaque motif séquentiel résumé dans le consensus de la classe courante doit avoir une valeur de discrimination supérieure ou égale à ρ vis-à-vis des autres classes.

Outre le fait d'être une alternative à la fouille de motifs, l'extraction de consensus est pertinente dans le contexte du *clustering* de séquences. Lorsque les classes d'une base de données ne sont pas connues, une approche par *clustering* permet alors de les générer d'une manière non supervisée. Un consensus permet alors d'identifier rapidement l'information qui caractérise chaque classe ou *cluster*. Cependant, ces méthodes nécessitent l'utilisation d'une mesure de dissimilarité sur les séquences. Nous avons alors proposé la mesure de dissimilarité *Dynamic Sequence Warping* adaptée aux séquences temporelles d'itemets.

L'application et la validation des méthodes énoncées ci-dessus ont été effectuées sur des données temporelles relatives aux écosystèmes aquatiques, en collaboration avec des hydrobiologistes dans le cadre du projet ANR Fresqueau. Les experts ont souligné l'intérêt de ces approches nouvelles en complément des méthodes statistiques qu'ils ont pour habitude d'utiliser.

8.2 Perspectives

Optimiser *OrderSpan*

Nous avons présenté une optimisation pour *OrderSpan* basée sur les propriétés de l'algorithme *CloSpan* [YHA03]. Il serait intéressant d'étudier d'autres algorithmes qui sont également efficaces dans l'extraction des motifs séquentiels clos. Par exemple, l'algorithme *BIDE* [WH04] est basé sur une approche alternative. L'idée consiste à élaguer l'espace de recherche d'une manière différente. Là où *CloSpan* permet de fusionner certaines branches de l'espace de recherche, *BIDE* permet lui de revenir en arrière et de supprimer une branche en partie explorée et redondante, qu'il n'est pas possible de détecter avec la méthode de *CloSpan*.

Bien que les approches soient en général présentées comme concurrentes pour la fouille de motifs séquentiels clos, nous pensons qu'il est tout à fait possible de combiner les avantages de ces deux méthodes pour être encore plus efficace lors de l'extraction des motifs partiellement ordonnés clos.

Ajouts de contraintes temporelles

Les algorithmes d'extraction de motifs partiellement ordonnés clos et d'extraction de consensus ont été présentés pour considérer tous les types de bases de données séquentielles. Cependant, dans le cadre de l'extraction de motifs séquentiels dans les bases de données temporelles, de nombreux auteurs ont travaillé sur l'utilisation de contraintes temporelles ou autres [PHW02, MPT04]. Par exemple, ces contraintes sont la longueur maximale des chemins d'un motif, l'écart temporel entre deux itemsets consécutifs, la présence obligatoire de certains items ou encore le fait qu'un motif respecte un certain modèle numérique [FSP$^+$14]. Ainsi, il est possible d'adapter certaines contraintes utilisées dans l'extraction de motifs séquentiels pour l'extraction de motifs ou de consensus partiellement ordonnés.

Ces techniques ont plusieurs avantages : (1) extraire seulement les motifs ou consensus qui respectent les contraintes entrées par l'utilisateur et (2) améliorer dans certains cas les performances des algorithmes en limitant l'exploration de l'espace de recherche.

k motifs partiellement ordonnés clos pour la classification

Dans le cadre de données temporelles, il est souvent difficile d'appliquer directement des algorithmes de classification classiques tels que les arbres de décisions, les réseaux de neurones, ou bien les machines à vecteurs de support [HK06]. Une solution est alors de passer par l'extraction de motifs séquentiels. Ainsi, de nombreux travaux ont porté sur l'utilisation des motifs séquentiels comme descripteurs dans un contexte de classification de données séquentielles [TL09, FBP$^+$11, BVCH11]. Dans une moindre mesure, les motifs partiellement ordonnés ont également été utilisés pour la classification [MF10].

Cependant, lorsque le nombre de motifs extraits est volumineux, les algorithmes de classification ont du mal à passer à l'échelle et il est alors nécessaire de filtrer un sous-ensemble de motifs. Il serait alors intéressant d'évaluer l'efficacité de notre algorithme de sélection de k motifs d'intérêt, en utilisant ensuite les motifs filtrés comme descripteurs des données pour des problématiques de classification. L'idée serait de chercher l'ensemble minimal de k motifs qui donne les meilleurs résultats de classification.

Aide à la navigation dans les motifs

Nous avons présenté des méthodes pour filtrer les motifs et pour résumer les données. Cependant, une approche alternative pourrait être de permettre à l'utilisateur de naviguer lui même dans l'ensemble des motifs extraits.

Une telle navigation pourrait être basée sur différentes méthodes : (1) pouvoir, à partir d'un motif, restituer l'ensemble des k motifs qui sont les plus proches en utilisant une mesure de dissimilarité telle que celle proposée dans le chapitre 4, et (2) pouvoir naviguer dans les motifs en partant des plus généraux aux plus spécifiques, ou inversement. C'est-à-dire, à partir d'un motif partiellement ordonné clos sur un ensemble $\mathcal{S}$, restituer les motifs clos sur des ensembles $\mathcal{S}'$ tel que $\mathcal{S}' \subset \mathcal{S}$ ou $\mathcal{S} \subset \mathcal{S}'$. Avec une simple modification sur *OrderSpan*, nous pensons qu'il est possible de restituer le treillis des motifs partiellement ordonnés clos sous la forme d'une structure de treillis de Galois [GK01]. Au travers de ce treillis, il serait alors possible de naviguer dans les motifs, des plus généraux aux plus spécifiques (et inversement).

Approfondir l'application aux données hydrobiologiques

Dans le chapitre 7, nous avons appliqué nos méthodes sur les données du projet Fresqueau et en particulier sur les trois indices biologiques *IBGN*, *IBD* et *IPR*. Ces indices ont l'avantage d'avoir été mesurés et étudiés depuis de très nombreuses années dans les écosystèmes aquatiques. Cela nous a permis d'obtenir un volume de données important à analyser et une interprétation des experts. Cependant, il existe d'autres indices biologiques comme l'*IBMR* et l'*IOBS* (présentés dans le chapitre 6) qui n'ont pas encore été analysés avec nos méthodes. Ainsi, nous souhaitons également les étudier pour potentiellement découvrir de nouvelles connaissances utiles sur les rivières. En effet, explorer le plus de dimensions biologiques possibles permet d'avoir une meilleure idée de la qualité biologique globale d'une rivière.

Pour finir, les expérimentations ont montré que les motifs extraits sont capables de capturer des informations discriminantes sur la qualité biologique des rivières. Il serait intéressant d'étendre notre processus à la classification et la prédiction de l'état biologique. Mesurer une note d'indice biologique sur le terrain entraîne un coût important et un certain délai d'analyse. Prédire l'état biologique des rivières, selon un compartiment biologique, en analysant seulement les mesures récentes de physico-chimie prendrait alors tout son intérêt.

Publications dans le cadre de cette thèse

Revues internationales avec comité de lecture

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Corinne Grac, Florence Le Ber, Danielle Levet, Maguelonne Teisseire :** *Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment*, Ecological Informatics (accepté pour publication)

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Maguelonne Teisseire :** *Mining Closed Partially Ordered Patterns, a new optimized algorithm*, Knowledge Based System (accepté pour publication)

Conférences internationales avec comité de lecture

— **Pierre Accorsi, Mickaël Fabrègue, Arnaud Sallaberry, Flavie Cernesson, Nathalie Lalande, Agnès Braud, Sandra Bringay, Florence Le Ber, Pascal Poncelet, Maguelonne Teisseire :** *HydroQual : Visual Analysis for Water Quality of Rivers*, VAST 2014, Paris, France

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Maguelonne Teisseire :** *OrderSpan : mining closed partially ordered patterns*, IDA 2013, London, England

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Maguelonne Teisseire :** *Including spatial relations and scales within sequential pattern extraction*, Discovery Science 2012, Lyon, France

Conférences et ateliers nationaux avec comité de lecture

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Maguelonne Teisseire :** *Recherche de motifs partiellement ordonnés clos discriminants pour caractériser l'état des milieux aquatiques*, AnaENV 2014. Atelier associé à la conférence RFIA, Rouen, France

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Charles Lecellier, Pascal Poncelet, Maguelonne Teisseire :** *OrderGeneMiner : Logiciel pour l'extraction et la visualisation de motifs partiellement ordonnés à partir de puces à ADN*, Démonstration EGC 2013, Toulouse, France

— **Mickaël Fabrègue, Agnès Braud, Sandra Bringay, Florence Le Ber, Maguelonne Teisseire :** *Extraction de motifs spatio-temporels à différentes échelles avec gestion de relations spatiales qualitatives*, Inforsid 2012, Montpellier, France

Bibliographie

[AFGY02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 429–435. ACM, 2002.

[AFN02] AFNOR (Association Française de NORmalisation). Qualité de l'eau : détermination de l'Indice Oligochètes de Bioindication des Sédiments (IOBS). Norne Française NF T90-390., 2002.

[AFN03] AFNOR (Association Française de NORmalisation). Qualité de l'eau : détermination de l'Indice Biologique Macrophytique en Rivière (IBMR). Norne Française NF T90-395., 2003.

[AFN04a] AFNOR (Association Française de NORmalisation). Qualité de l'eau : détermination de l'Indice Biologique Global Normalisé (IBGN). Norne Française NF T90-350., 1992, révision 2004.

[AFN04b] AFNOR (Association Française de NORmalisation). Qualité de l'eau : détermination de l'Indice poissons rivière (IPR). Norne Française NF T90-344., 2004.

[AFN07] AFNOR (Association Française de NORmalisation). Qualité de l'eau : détermination de l'Indice Biologique Diatomées (IBD). Norne Française NF T90-354., 2000, révision 2007.

[AFN09] AFNOR (Association Française de NORmalisation). Qualité de l'eau : Prélèvement des macro-invertébrés aquatiques en rivières peu profondes. Norne Française XP T90-333., 2009.

[AFN10] AFNOR (Association Française de NORmalisation). Qualité de l'eau : Traitement au laboratoire d'échantillons contenant des macro-invertébrés de cours d'eau. Norne Française XP T90-388., 2010.

[AFS+14] P. Accorsi, M. Fabregue, A. Sallaberry, F. Cernesson, N. Lalande, A. Braud, S. Bringay, F. Le Ber, P. Poncelet, and M. Teisseire. Hydroqual : Visual analysis of river water quality. In Visual Analytics Science and Technology, 2014. VAST 2014. IEEE, 2014.

[Arc] ArcGIS Geostatistical Analyst. `http://www.esri.com/software/arcgis/extensions/geostatistical`. [Online ; accessed 31-March-2014].

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, pages 487–499, 1994.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In International Conference on Data Engineering, ICDE, pages 3–14. IEEE, 1995.

[ASBF+12] Hugo Alatrista Salas, Sandra Bringay, Frédéric Flouvat, Nazha Selmaoui-Folcher, and Maguelonne Teisseire. The Pattern Next Door : Towards Spatio-sequential Pattern Discovery. In Advances in Knowledge Discovery and Data Mining, volume 7302, pages 157–168. Springer, 2012.

[BG97] Ingwer Brog and Patrick J. F. Groenen. Modern multidimensional scaling : Theory and applications. Springer-Verlag, 1997.

[BHR00] Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on, pages 39–48. IEEE, 2000.

[BLBBT09] Aurélie Bertaux, Florence Le Ber, Agnès Braud, and Michèle Trémolières. Identifying Ecological Traits : A Concrete FCA-Based Approach. In Formal Concept Analysis, volume 5548, pages 224–236. Springer, 2009.

[BM70] Marc Barbut and Bernard Monjardet. Ordre et Classification : Algèbre et Combinatoire, volume 2. Hachette, 1970.

[BMZ11] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. Journal of Computational Science, pages 1–8, 2011.

[BVCH11] Iyad Batal, Hamed Valizadegan, Gregory F Cooper, and Milos Hauskrecht. A pattern mining approach for classifying multivariate temporal data. In Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on, pages 358–365. IEEE, 2011.

[CF06] Deepayan Chakrabarti and Christos Faloutsos. Graph mining : Laws, generators, and algorithms. ACM Comput. Surv., 38(1), June 2006.

[CG05] Gemma Casas-Garriga. Summarizing sequential data with closed partial orders. In SIAM International Conference on Data Mining, SDM, 2005.

[Con06] Shengnan Cong. A Sampling-Based Framework for Parallel Mining Frequent Patterns. PhD thesis, University of Illinois, 2006.

[Cor99] E Coring. Situation and developments of algal (diatom)-based techniques for monitoring rivers in Germany. Use of Algae for Monitoring Rivers III, pages 122–127, 1999.

[DDD+07] Eleni Dakou, Tom D'Heygere, Andy P. Dedecker, Peter L.M. Goethals, Maria Lazaridou-Dimitriadou, and Niels Pauw. Decision Tree Models for Prediction of Macroinvertebrate Taxa in the River Axios (Northern Greece). Aquatic Ecology, 41 :399–411, 2007.

[Den14] Zhi-Hong Deng. Fast mining Top-Rank-k frequent patterns by using Node-lists . Expert Systems with Applications, 41 :1763 – 1768, 2014.

[DF07] Zhi-Hong Deng and Guo-Dong Fang. Mining top-rank-K frequent patterns. In Machine Learning and Cybernetics, 2007 International Conference on, volume 2, pages 851–856. IEEE, 2007.

[DGGP04] Andy P Dedecker, Peter L.M Goethals, Wim Gabriels, and Niels De Pauw. Optimization of artificial neural network (ann) model design for prediction of macroinvertebrates in the zwalm river basin (flanders, belgium). Ecological Modelling, 174 :161 – 173, 2004.

[DGP03] Tom D'Heygere, Peter L.M. Goethals, and Niels De Pauw. Use of genetic algorithms to select input variables in decision tree models for the prediction of benthic macroinvertebrates. Ecological Modelling, 160 :291 – 300, 2003.

[dJBC10] Aí da Jiménez, Fernando Berzal, and Juan-Carlos Cubero. Frequent tree pattern mining : A survey. Intell. Data Anal., 14(6) :603–622, 2010.

[DL99] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns : Discovering trends and differences. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, pages 43–52. ACM, 1999.

[DLBH13] Xavier Dolques, Florence Le Ber, and Marianne Huchard. AOC-posets : a scalable alternative to Concept Lattices for Relational Concept Analysis. In CLA 2013 : 10th International Conference on Concept Lattices and Their Applications, pages 129–140, 2013.

[EGK+03] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and dynagraph ? static and dynamic graph drawing tools. In Graph Drawing Software, pages 127–148. Springer-Verlag, 2003.

[ERC+13] Elias Egho, Chedy Raïssi, Toon Calders, Nicolas Jay, Amedeo Napoli, et al. Vers une mesure de similarité pour les séquences complexes. Extraction et gestion des connaissances (EGC), 2013.

[ET05] David Enke and Suraphan Thawornwong. The use of data mining and neural networks for forecasting stock market returns. Expert Systems with applications, pages 927–940, 2005.

[Eur00] European Union. Directive 2000/60/ec of the European parliament and of the council of 23 october 2000 establishing a framework for community action in the field of water policy. Official Journal, OJ L 327 :1–73, 2000.

[FBP+11] Mickael Fabrègue, Sandra Bringay, Pascal Poncelet, Maguelonne Teisseire, and Béatrice Orsetti. Mining microarray data to predict the histological grade of a breast cancer. Journal of Biomedical Informatics, 2011.

[FPSSU96] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. Advances in knowledge discovery and data mining. 1996.

[FSP+14] Frédéric Flouvat, Jérémy Sanhes, Claude Pasquier, Nazha Selmaoui-Folcher, and Jean-François Boulicaut. Improving pattern discovery relevancy by deriving constraints from expert models. In Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014), pages 327–332, 2014.

[FVWT13] Philippe Fournier-Viger, Cheng-Wei Wu, and Vincent S Tseng. Mining maximal sequential patterns without candidate maintenance. In Advanced Data Mining and Applications, pages 169–180. Springer, 2013.

[GB12] Aloysius George and D. Binu. Drl-prefixspan : A novel pattern growth algorithm for discovering downturn, revision and launch (drl) sequential patterns. Central European Journal of Computer Science, 2 :426–439, 2012.

[GDG+07] Peter L.M. Goethals, AndyP. Dedecker, Wim Gabriels, Sovan Lek, and Niels Pauw. Applications of artificial neural networks predicting macroinvertebrates in freshwaters. Aquatic Ecology, 41 :491–508, 2007.

[GH06] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining : A survey. ACM Computing Survey, 38, 2006.

[GK01] Bernhard Ganter and Sergei O Kuznetsov. Pattern structures and their projections. In Conceptual Structures : Broadening the Base, pages 129–142. Springer, 2001.

[GMS04] Robert Giegerich, Carsten Meyer, and Peter Steffen. A discipline of dynamic programming over sequence data. Science of Computer Programming, 51(3) :215–263, 2004.

[GRS99] Minos N. Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Spirit : Sequential pattern mining with regular expression constraints. In Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99, pages 223–234, 1999.

[Guz82] Philip S Guzelian. Comparative toxicology of chlordecone (kepone) in humans and experimental animals. Annual review of pharmacology and toxicology, 22(1) :89–113, 1982.

[GW99] Bernhard Ganter and Rudolf Wille. Formal concept analysis, volume 284. Springer Berlin, 1999.

[GZ01] Karam Gouda and Mohammed Javeed Zaki. Efficiently Mining Maximal Frequent Itemsets. In Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM '01, pages 163–170. IEEE, 2001.

[GZ04] Bart Goethals and Mohammed J. Zaki. Advances in frequent itemset mining implementations : report on FIMI'03. SIGKDD Explor. Newsl., 6(1) :109–117, 2004.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. Bell System Technical Journal, 29 :147–160, 1950.

[HK06] Jiawei Han and Micheline Kamber. Data Mining, Southeast Asia Edition : Concepts and Techniques. Morgan kaufmann, 2006.

[HWLT02] Jiawei Han, Jianyong Wang, Ying Lu, and Petre Tzvetkov. Mining top-k frequent closed patterns without minimum support. In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, volume 3453, pages 211–218. IEEE, 2002.

[HZZ08] Yan Huang, Liqin Zhang, and Pusheng Zhang. A Framework for Mining Sequential Patterns from Spatio-Temporal Event Data Sets. IEEE Transactions on Knowledge and Data Engineering, 20(4) :433–448, 2008.

[Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. New phytologist, 11(2) :37–50, 1912.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melancon. Visual analytics : Definition, process, and challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, Information Visualization : Human-Centered Issues and Perspectives, volume 4950 of Lecture Notes in Computer Science, pages 154–175. Springer Berlin Heidelberg, 2008.

[KNM+10] Dragi Kocev, Andreja Naumoski, Kosta Mitreski, Svetislav Krstić, and Sašo Džeroski. Learning habitat models for the diatom community in lake prespa. Ecological Modelling, 221(2) :330–337, 2010.

[KPW95] M G Kelly, C J Penny, and B A Whitton. Comparative performance of bentic diatom indices to asses river water quality. Hydrobiologia, 302 :179–188, 1995.

[KPWD03] Hye-Chung Kum, Jian Pei, Wei Wang, and Dean Duncan. Approxmap : Approximate mining of consensus sequential patterns. In SDM, pages 311–315. SIAM, 2003.

[KR09] Leonard Kaufman and Peter J Rousseeuw. Finding groups in data : an introduction to cluster analysis, volume 344. John Wiley & Sons, 2009.

[LEK+03] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In SDM, pages 25–36, 2003.

[LHK+07] Anthony J.T Lee, Ruey-Wen Hong, Wei-Min Ko, Wen-Kwang Tsao, and Hsiu-Hui Lin. Mining spatial association rules in image databases. Information Sciences, 177(7) :1593–1608, 2007.

[LL12] Pierre Legendre and Loic FJ Legendre. Numerical ecology, volume 20. Elsevier, 2012.

[LMP03] Antti Leino, Heikki Mannila, and Ritva Liisa Pitkänen. Rule discovery and probabilistic modeling for onomastic data. In PKDD, pages 22–26. Springer, 2003.

[MCK+07] Luc Multigner, Sylvaine Cordier, Philippe Kadhel, Farida Huc-Terki, Pascal Blanchet, Henri Bataille, and Eustase Janky. Pollution par le chlordécone aux Antilles Quel impact sur la santé de la population ? Environnement, Risques & Santé, 6 :405–407, 2007.

[MF10] Fabian Moerchen and Dmitriy Fradkin. Robust mining of time intervals with semi-interval partial order patterns. In SDM, pages 315–326. SIAM, 2010.

[MPT04] Florent Masseglia, Pascal Poncelet, and Maguelonne Teisseire. Pre-processing time constraints for efficiently mining generalized sequential patterns. In Temporal Representation and Reasoning, 2004. TIME 2004. Proceedings. 11th International Symposium on, pages 87–95. IEEE, 2004.

[MTP05] Florent Masseglia, Maguelonne Teisseire, and Pascal Poncelet. Sequential pattern mining : A survey on issues and approaches. In Encyclopedia of Data Warehousing and Mining, nformation Science Publishing, pages 3–29, 2005.

[MTV97] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1 :259–289, 1997.

[Nav01] Gonzalo Navarro. A guided tour to approximate string matching. ACM computing surveys (CSUR), 33(1) :31–88, 2001.

[OMHO94] Gary J Olsen, Hideo Matsuda, Ray Hagstrom, and Ross Overbeek. fastdnaml : a tool for construction of phylogenetic trees of dna sequences using maximum likelihood. Computer applications in the biosciences : CABIOS, pages 41–48, 1994.

[PBTL99] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In Proceedings of the 7th International Conference on Database Theory, ICDT '99, pages 398–416. Springer, 1999.

[PC09] Marc Plantevit and Bruno Crémilleux. Condensed representation of sequential patterns according to frequency-based measures. In Advances in Intelligent Data Analysis VIII, volume 5772 of Lecture Notes in Computer Science, pages 155–166. Springer, 2009.

[PCKR09] Marc Plantevit, Thierry Charnois, Jiri Klema, and Christophe Rigotti. Combining sequence and itemset mining to discover named entities in biomedical texts : a new type of pattern. International Journal of Data Mining, Modelling and Management, 1 :119–148, 2009.

[PHMA+04] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth : The prefixspan approach. IEEE Transactions on Knowledge and Data Engineering, 16 :1424–1440, 2004.

[PHP+01] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. Multi-dimensional sequential pattern mining. In Proceedings of the tenth international conference on Information and knowledge management, CIKM '01, pages 81–88. ACM, 2001.

[PHW02] Jian Pei, Jiawei Han, and Wei Wang. Mining sequential patterns with constraints in large databases. In Proceedings of the eleventh international conference on Information and Knowledge Management, pages 18–25. ACM, 2002.

[PHW07] Jian Pei, Jiawei Han, and Wei Wang. Constraint-based sequential pattern mining : the pattern-growth methods. Journal of Intelligent Information Systems, 28 :133–160, 2007.

[PKSG05] Panagiotis Papapetrou, George Kollios, Stan Sclaroff, and Dimitrios Gunopulos. Discovering frequent arrangements of temporal intervals. In Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05), pages 354–361. IEEE, 2005.

[PLL+10] Marc Plantevit, Anne Laurent, Dominique Laurent, Maguelonne Teisseire, and Yeow WEI Choong. Mining multidimensional and multilevel sequential patterns. ACM Trans. Knowl. Discov. Data, 4 :1–37, 2010.

[PP90] Marc Parizeau and Réjean. Plamondon. A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 :710–717, 1990.

[PWL+06] Jian Pei, Haixun Wang, Jian Liu, Ke Wang, Jianyong Wang, and Philip S. Yu. Discovering frequent closed partial orders from strings. IEEE Transactions on Knowledge and Data Engineering, 18, 2006.

[Rab11] Julien Rabatel. Extraction de motifs contextuels : Enjeux et applications dans les données séquentielles. PhD thesis, Université de Montpellier II, 2011.

[ROC+13] Friedrich Recknagel, Ilia Ostrovsky, Hongqing Cao, Tamar Zohary, and Xiaoqing Zhang. Ecological relationships, thresholds and time-lags determining phytoplankton community dynamics of lake kinneret, israel elucidated by evolutionary computation and wavelets. Ecological Modelling, 255 :70–86, 2013.

[RPT06] Chedy Raissi, Pascal Poncelet, and Maguelonne Teisseire. Speed : mining maximal sequential patterns over data strearns. In Intelligent Systems, 2006 3rd International IEEE Conference on, pages 546–552. IEEE, 2006.

[RWD+09] Jiadong Ren, Libo Wang, Jun Dong, Changzhen Hu, and Kunsheng Wang. A Novel Sequential Pattern Mining Algorithm for the Feature Discovery of Software Fault. In International Conference on Computational Intelligence and Software Engineering, volume 5854 of CiSE, pages 439–447. IEEE, 2009.

[SBF+12] Hugo Alatrista Salas, Sandra Bringay, Frédéric Flouvat, Nazha Selmaoui-Folcher, and Maguelonne Teisseire. The pattern next door : Towards spatio-sequential pattern discovery. In Advances in Knowledge Discovery and Data Mining, pages 157–168. Springer, 2012.

[SC71] Hiroaki Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. In Proceedings of the 7th International Congress on Acoustics, volume 3, pages 65–69, 1971.

[SC78] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, 26(1) :43–49, 1978.

[SCA+11] Daniela Stojanova, Michelangelo Ceci, Annalisa Appice, Donato Malerba, and Sašo Džeroski. Global and local spatial autocorrelation in predictive clustering trees. In Proceedings of the 14th international conference on Discovery science, DS'11, pages 307–322. Springer, 2011.

[SLR00] Allan H Smith, Elena O Lingas, and Mahfuzar Rahman. Contamination of drinking-water by arsenic in bangladesh : a public health emergency. Bulletin of the World Health Organization, 78(9) :1093–1103, 2000.

[SPB+11] Arnaud Sallaberry, Nicolas Pecheur, Sandra Bringay, Mathieu Roche, and Maguelonne Teisseire. Sequential patterns mining and gene sequence visualization to discover novelty from microarray data. Journal of Biomedical Informatics, 44 :760–774, 2011.

[STT81] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics, 11(2) :109–125, 1981.

[TC05] James J. Thomas and Kristen A. Cook. Illuminating the Path. IEEE, 2005.

[TC12] Nikolaj Tatti and Boris Cule. Mining closed strict episodes. Data Mining and Knowledge Discovery, 25 :34–66, 2012.

[TCM+02] David Tilman, Kenneth G Cassman, Pamela A Matson, Rosamond Naylor, and Stephen Polasky. Agricultural sustainability and intensive production practices. Nature, 418 :671–677, 2002.

[TG01] Ilias Tsoukatos and Dimitrios Gunopulos. Efficient Mining of Spatiotemporal Patterns. In Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, SSTD '01, pages 425–442. Springer, 2001.

[TL09] Vincent S. Tseng and Chao-Hui Lee. Effective temporal data classification by integrating sequential pattern mining and probabilistic induction. Expert Systems with Applications, 36(5) :9524 – 9532, 2009.

[TW08] Pawel Terlecki and Krzysztof Walczak. Efficient discovery of top-k minimal jumping emerging patterns. In Rough Sets and Current Trends in Computing, pages 438–447. Springer, 2008.

[TYH05] Petre Tzvetkov, Xifeng Yan, and Jiawei Han. Tsp : Mining top-k closed sequential patterns. Knowledge and Information Systems, 7(4) :438–457, 2005.

[WH04] Jianyong Wang and Jiawei Han. BIDE : efficient mining of frequent closed sequences. In International Conference on Data Engineering (ICDE'04), pages 79–90. IEEE, 2004.

[WM03] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. SIGKDD Explor. Newsl., 5(1) :59–68, 2003.

[YCH08] Yi-Chen E. Yang, Ximing Cai, and Edwin E. Herricks. Identification of hydrologic indicators related to fish diversity and abundance : A data mining approach for fish community analysis. Water Resources Research, 44, 2008.

[YHA03] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan : Mining closed sequential patterns in large datasets. In SIAM International Conference on Data Mining (SDM'03), pages 166–177. SIAM, 2003.

[YHN06] Ben Yahia, Tarek Hamrouni, and Engelbert Mephu Nguifo. Frequent closed itemset based algorithms : a thorough structural and analytical survey. SIGKDD Explor. Newsl., 8(1) :93–104, 2006.

[Zak01] Mohammed J Zaki. Spade : An efficient algorithm for mining frequent sequences. Machine learning, 42(1-2) :31–60, 2001.

[ZLC10] Wenzhi Zhou, Hongyan Liu, and Hong Cheng. Mining closed episodes from event sequences efficiently. In Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, volume 6118 of PAKDD, pages 310–318. Springer, 2010.

[ZTZL96] WL Zhang, ZX Tian, N Zhang, and XQ Li. Nitrate pollution of groundwater in northern China. Agriculture, Ecosystems & Environment, 59(3) :223–231, 1996.

Résumé

L'exploration des bases de données temporelles à l'aide de méthodes de fouille de données adaptées a fait l'objet de nombreux travaux de recherche. Cependant le volume d'informations extraites est souvent trop important et la tâche d'analyse reste alors difficile. Dans cette thèse, nous présentons des méthodes pour synthétiser et filtrer l'information extraite. L'objectif est de restituer des résultats qui soient interprétables par un humain. Pour cela, nous avons exploité la notion de séquence partiellement ordonnée et nous proposons (1) un algorithme qui extrait l'ensemble des motifs partiellement ordonnés clos ; (2) un post-traitement pour filtrer un ensemble de motifs d'intérêt pour l'utilisateur et (3) une approche qui extrait un consensus partiellement ordonné comme alternative à l'extraction de motifs. Les méthodes proposées ont été testées pour validation sur des données hydrobiologiques issues du projet ANR Fresqueau. De plus, elles ont été implantées dans un logiciel de visualisation destiné aux hydrobiologistes pour l'analyse de la qualité des cours d'eau.

Mots clefs : Fouille de données séquentielles, Motifs séquentiels, Motifs partiellement ordonnés, Résumé de l'information, Hydrobiologie

---

Abstract

Exploring temporal databases with suitable data mining methods have been the subject of several studies. However, it often leads to an excessive volume of extracted information and the analysis is difficult for the user. We addressed this issue and we specifically focused on methods that synthesize and filter extracted information. The objective is to provide interpretable results for humans. Thus, we relied on the notion of partially ordered sequence and we proposed (1) an algorithm that extracts the set of closed partially ordered patterns ; (2) a post-processing to filter some interesting patterns for the user and (3) an approach that extracts a partially ordered consensus as an alternative to pattern extraction. The proposed methods were applied for validation on hydrobiological data from the Fresqueau ANR project. In addition, they have been implemented in a visualization tool designed for hydrobiologists for water course quality analysis.

Keywords : Sequential data mining, Sequential patterns, Partially ordered patterns, Summarized information, Hydrobiology