THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITE DE MONTPELLIER

En Informatique

École doctorale Information, Structures, Systèmes

Unité de recherche LIRMM

Visualisation de données dynamiques et complexes : des séries temporelles hiérarchiques aux graphes multicouches

Présentée par Erick CUENCA PAUTA

Le 12 Novembre 2018

Sous la direction de Pascal PONCELET

Devant le jury composé de

Guy MELANÇON, Professeur, Université de Bordeaux	Rapporteur
Christophe HURTER, Professeur, École Nationale de l'Aviation Civile	Rapporteur
Marianne HUCHARD, Professeur, Université de Montpellier	Présidente
Romain BOURQUI, Maître de Conférences, Université de Bordeaux	Examinateur
Pascal PONCELET, Professeur, Université de Montpellier	Directeur
Arnaud SALLABERRY, Maître de Conférences, Université Paul-Valéry Montpellier 3	Co-encadrant

UNIVERSITÉ DE MONTPELLIER

Résumé

L'analyse de données de plus en plus complexes, volumineuses et issues de différentes sources (*e.g.* internet, médias sociaux, etc.) est une tâche difficile. Elle reste cependant cruciale dans de très nombreux domaines d'application. Elle implique, pour pouvoir en extraire des connaissances, de mieux comprendre la nature des données, leur évolution ou les nombreuses relations complexes qu'elles peuvent contenir. La visualisation d'informations s'intéresse aux méthodes de représentations visuelles et interactives permettant d'aider un utilisateur à extraire des connaissances. C'est dans ce contexte que se situe le travail présenté dans ce mémoire.

Dans un premier temps, nous nous intéressons à la visualisation de longues séries temporelles hiérarchiques. Après avoir analysé les différentes approches existantes, nous présentons le système MultiStream permettant de visualiser, explorer et comparer l'évolution de séries organisées dans une structure hiérarchique. Nous illustrons son utilisation par deux exemples d'utilisation : émotions exprimées dans des médias sociaux et évolution des genres musicaux. Dans un second temps nous abordons la problématique de données complexes modélisées sous la forme de graphes multicouches (différents types d'arêtes peuvent relier les nœuds). Plus particulièrement nous nous intéressons au requêtage visuel de graphes volumineux en présentant VERTIGo un système qui permet de construire des requêtes, d'interroger un moteur spécifique, de visualiser/explorer les résultats à différents niveaux de détail et de suggérer de nouvelles extensions de requêtes. Nous illustrons son utilisation à l'aide d'un graphe d'auteurs provenant de différentes communautés.

Mots-clés : Visualisation d'information, focus+contexte, requêtage visuel, séries temporelles hiérarchiques, graphes multicouches.

Abstract

The analysis of data that is increasingly complex, large and from different sources (*e.g.* internet, social medias, etc.) is a difficult task. However, it remains crucial for many fields of application. It implies, in order to extract knowledge, to better understand the nature of the data, its evolution or the many complex relationships it may contain. Information visualization is about visual and interactive representation methods to help a user to extract knowledge. The work presented in this document takes place in this context.

At first, we are interested in the visualization of large hierarchical time series. After analyzing the different existing approaches, we present the MultiStream system for visualizing, exploring and comparing the evolution of the series organized into a hierarchical structure. We illustrate its use by two examples: emotions expressed in social media and the evolution of musical genres. In a second time, we tackle the problem of complex data modeled in the form of multilayer graphs (different types of edges can connect the nodes). More specifically, we are interested in the visual querying of large graphs and we present Vertigo, a system which makes it possible to build queries, to launch them on a specific engine, to visualize/explore the results at different levels of details and to suggest new query extensions. We illustrate its use with a graph of co-authors from different communities.

Keywords: Information visualization, focus+context, visual querying, hierarchical time series, multilayer graphs.

Remerciements

Tout d'abord, je remercie et dédie cette thèse à mon épouse Karol pour tout son soutien inconditionnel pendant le déroulement de cette thèse. A ma famille aussi, ma mère Maria, mon père Victor et mes deux frères Christian et Alex car même si nous étions loin, j'ai toujours senti vos gestes d'encouragement et de soutien. Je vous embrasse à la distance.

Je voudrais remercier profondément mes deux encadrants, Pascal Poncelet et Arnaud Sallaberry. C'était grâce à un module enseigné par Arnaud que j'ai pu rencontrer Pascal et faire mon stage de Master pour ensuite continuer cette thèse. Je les remercie pour m'avoir guidé dans ce monde de la recherche qui était (un peu) inconnu pour moi. Je les remercie pour les réunions et discussions qui non seulement m'ont aidé à renforcer mes connaissances mais ont également permis de vrais échanges. Un grand merci à eux.

Je souhaite remercier les membres de mon jury de thèse. Guy Melançon et Christophe Hurter pour avoir accepté de relire mon manuscrit de thèse et pour toutes les remarques pertinentes, et Marianne Huchard et Romain Bourqui pour avoir accepté de participer à mon jury et d'évaluer mon travail.

Je tiens aussi à remercier la Secrétaire d'Éducation du Gouvernement de l'Équateur (SENESCYT) pour avoir financé les travaux présentés dans cette thèse. Ainsi que l'Université de Montpellier (UM), le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) et l'équipe ADVANSE qui m'ont accueilli et permis de réaliser cette thèse.

Je voudrais remercier toutes les personnes qui ont contribué aux travaux de cette thèse. Je pense à mes encadrants, Florence Y. Wang du CSIRO et Dino Ienco d'IRSTEA pour les travaux présents dans les chapitres 2 et 3 de ce manuscrit. C'est clair que ces contributions nous ont permis d'atteindre les buts voulus. Merci beaucoup.

Un grand merci aux membres (et anciens membres que j'ai pu connaître) de l'équipe ADVANSE. Pascal, Arnaud, Sandra, Jérôme, Pierre, Nancy, Maximilien, ma colocataire de bureau Samiha, Jessica, Mike, Antonio, Vijay, Amine, Sarah, Bilel, Fati et Walleed pour leur disponibilité aux pauses cafés.

Pour finir, je voudrais aussi remercier tous mes amis qui m'ont supporté dans cette période, Lucie, Yonathan, Imene, Aziz. Également Eric et Dominique Cadier avec qui j'ai débuté dans le domaine de l'informatique en Équateur et pour lesquels j'ai beaucoup d'estime.

Sommaire

1 Introduction **1**

1.1 Contexte 2

1.2 Contributions 3

1.3 Organisation du mémoire 5

2 Visualisation de séries temporelles hiérarchiques **7**

2.1 Introduction 8

2.2 État de l'art 12

2.2.1 Visualisations de type stacked graphs et streamgraphs 13

2.2.2 Structure hiérarchique dans les séries temporelles 15

2.2.3 Techniques d'interaction 16

2.3 Critères 21

2.4 Encodages visuels et méthodes d'interaction 22

2.4.1 Conception visuelle 23

2.4.2 Vue globale 23

2.4.3 Vue multirésolution 24

2.4.4 Le contrôleur 26

2.4.5 Gestionnaire de hiérarchie 29

2.5 Contraintes techniques pour la vue multirésolution 35

2.5.1 Longueurs de pas de temps 35

2.5.2 Les couleurs 38
2.6 Discussion 41
2.6.1 Une approche flexible 41
2.6.2 Comparaison avec des techniques alternatives 43
2.7 Exemples 44
2.7.1 L'élection présidentielle des États-Unis en 2016 45
2.7.2 L'évolution des genres musicaux 49
2.8 Conclusion 54

3 Visualisation de graphes multicouches 55
3.1 Introduction 56
3.2 État de l'art 57
3.2.1 Systèmes visuels de requêtage de graphes 57
3.2.2 Suggestion visuelle de requêtes 60
3.2.3 Comparaison des approches 61
3.3 Critères 62
3.4 Encodages visuels et modes d'interaction 64
3.4.1 Conception visuelle 64
3.4.2 Vue de la requête 64
3.4.3 Vue du graphe 69
3.4.4 Vue des résultats 70
3.5 Interactions 72
3.5.1 Visualiser les résultats 74
3.5.2 Explorer les résultats 77
3.6 Une approche basée sur les diagrammes Kelp 79
3.6.1 Positionner les sommets et les résultats 79
3.6.2 Génération d'une visualisation imbriquée 80

3.6.3 Mise en œuvre et équipement 83
3.7 Exemple 83
3.8 Conclusion 88

4 Conclusions générales et perspectives **89**
4.1 Résumé des contributions 90
4.1.1 Visualisation de séries temporelles hiérarchiques 90
4.1.2 Visualisation de graphes multicouches 90
4.2 Perspectives 91
4.2.1 Agrégation temporelle des séries 91
4.2.2 Fouille visuelle de graphes 92
4.2.3 Traitement de graphes dynamiques 92

Bibliographie **94**

Liste des figures

1.1 A gauche, des membres sont liés via une relation d'amitié sur Facebook. A droite, les membres sont liés par plusieurs arêtes représentant les relations issues de trois médias sociaux. 2

1.2 L'approche MultiStream est utilisée pour visualiser l'évolution de séries temporelles hiérarchiques. (a) Une vue globale représente les séries à un haut niveau d'abstraction. (b) Une vue multirésolution représente les séries à différents niveaux d'abstraction. (c) Un contrôleur permet de sélectionner un intervalle de temps et relie la vue globale et la vue multirésolution. (d) Un gestionnaire de hiérarchie permet de naviguer dans la structure hiérarchique des séries. 4

1.3 L'approche VERTIGo est utilisée pour explorer un réseau de co-auteurs. (a) Une vue de la requête permet de construire visuellement une requête et de l'exécuter sur moteur de recherche pour extraire des résultats. (b) Une vue du graphe montre les emplacements des résultats grâce à une carte de chaleur. (c) Un histogramme permet de filtrer les résultats en fonction de la taille de leur boîte englobante. (d) Une vue de résultats montre les occurrences pour un ensemble d'entités sélectionnées, *e.g.* les résultats contenant l'auteur N. Ramakrishnan. (e) A un niveau de zoom élevé, la vue du graphe montre les relations d'un sous-ensemble de résultats sélectionnés grâce à un diagramme Kelp. 5

2.1 Un diagramme linéaire montrant l'évolution du nombre de chansons écoutées du genre musical *alternative metal* par un utilisateur. 9

2.2 Un diagramme linéaire montrant l'évolution du nombre de chansons écoutées par un utilisateur selon leur genre musical : *alternative metal*, *black metal*, *doom metal*, *classic jazz*, etc. 9

2.3 Un stacked graph (a) et un streamgraph (b) montrant l'évolution du nombre de chansons par genre musical écoutés par un utilisateur. . . 10

2.4 Problème de surcharge d'un streamgraph avec une centaine de couches (séries temporelles) qu'il est difficile d'analyser. 11

2.5 Agrégation de séries temporelles. (a) Le streamgraph montre l'évolution des genres: *alternative metal*, *power metal*, *doom metal* et *black metal*. (b) Le streamgraph montre l'agrégation des genres de la Figure (a) pour le genre *metal*. 12

2.6 TIARA [47] utilise une visualisation de type stacked graph pour représenter l'évolution du contenu thématique d'une collection de documents. 13

2.7 NameVoyager [71] adopte une visualisation de type stacked graph pour explorer les tendances sur le choix des prénoms d'enfants au cours du temps. 14

2.8 Les ThemeRivers [34,35] utilisent une ligne de base centrale parallèle à l'axe temporel et les couches sont reparties le long de cet axe simulant ainsi le flux d'une rivière. 14

2.9 Les Streamgraphs [10] permettent d'obtenir une représentation plus « lisse » par rapport à la représentation de type ThemeRiver. 14

2.10 BookVoyager [72] utilise un stacked graph pour visualiser l'évolution d'une hiérarchie de catégories et de sous-catégories de livres. 15

2.11 ManyEyes [68] affiche sur un stacked graph la répartition du budget fédéral historique des États-Unis. 16

2.12 TouchWave [3] utilise un streamgraph pour représenter un historique musical. 16

2.13 NewsLab [29] permet de visualiser les mots-clés d'une collection de vidéos d'actualités organisées hiérarchiquement par thèmes. 17

2.14 HierarchicalTopics [24] permet de visualiser les sujets organisés en catégories et sous-catégories issus d'une collection de textes. 17

2.15 RoseRiver [22] visualise les sujets hiérarchisés dans une collection de texte. 18

2.16 La technique de zoom. (a) Le contexte est affiché mais les détails ne sont pas disponibles. (b) En zoom avant, les détails sont affichés mais le contexte n'est plus disponible. 18

2.17 (a) Technique dite focus+contexte. (b) Distorsion de type *fisheye*. . . 19

2.18 La technique de vue d'ensemble+détail composée par deux vues. La vue de gauche montre les détails et la vue de droite montre en même temps le contexte. 20

2.19 Modèle émotionnel circumplex de Russell. 21

2.20 La vue globale montre le niveau supérieur de la hiérarchie; elle permet à l'utilisateur d'observer les principaux motifs temporels. 23

2.21 La vue multirésolution. (a) Les zones de contexte représentent un haut niveau de la hiérarchie, (c) la zone détaillée représente un bas niveau de la hiérarchie et (b) les zones de transition représentent la transition entre les zones de contexte et la zone détaillée. 25

2.22 Le contrôleur. les zones de contexte (a), les zones de transition (b) et la zone détaillée (c) sont gérées par des lignes colorées qui peuvent être étendues et réduites de manière interactive. La zone détaillée (c) reste colorée en gris pour rappeler à l'utilisateur le point de focalisation. 27

2.23 Utilisation du contrôleur (a) La zone détaillée est déplacée vers la gauche. (b) La zone détaillée est agrandie vers la droite. (c) La zone de transition droite est agrandie vers la droite. (d) La position des zones de contexte est verrouillée en cliquant sur l'icône du cadenas. . 28

2.25 Une infobulle affiche la valeur à chaque pas de temps dans le streamgraph. 28

2.24 En bas, le contrôleur montre les zones affichées dans la vue multirésolution. En haut, la vue multirésolution, résultat de la projection des zones définies par le contrôleur. Les lignes de projection (en pointillés) aident l'utilisateur à suivre ces projections. 29

2.26 Le gestionnaire de hiérarchie. Une hiérarchie de séries temporelles est représentée sous la forme d'un arbre, où les lignes verticales en pointillés indiquent les niveaux d'abstraction affichés sur la vue multirésolution. La ligne verticale en pointillés bleue indique les couches représentées dans les zones de contexte (*i.e.* haut niveau d'abstraction), la ligne verticale en pointillés verte indique les couches représentées dans la zone détaillée (*i.e.* bas niveau d'abstraction). 30

2.27 Interactions permettant d'explorer différents niveaux d'abstraction dans la hiérarchie. (a) Désagrégation: diviser le nœud sélectionné en nœuds enfants. (b) Agrégation: regrouper les nœuds enfants dans leur nœud parent. 32

2.28 Mise en évidence des séries. (a) Comme le nœud sélectionné *reggae* est représenté dans la zone détaillée (*i.e.* traversé par la ligne verte), seule la couche correspondante est mise en évidence. (b) Comme le nœud sélectionné *electronica* est représenté dans les zones de contexte (*i.e.* traversées par la ligne bleue), tous les descendants qui sont affichés dans la zone détaillée sont mis en surbrillance. 33

2.29 Filtrage des nœuds. (a) Comme le nœud de genre *electronica* est représenté dans la zone détaillée (*i.e.* traversé par la ligne verte) alors seule cette couche est filtrée. (b) Comme la couche de genre *electronica* est représentée dans les zones de contexte (*i.e.* traversé par la ligne bleue), alors tous leurs descendants sont également filtrés. 34

2.30 Conception de la projection des zones du contrôleur (en bas) sur la vue multirésolution (en haut). 35

2.31 La répartition horizontale des pas de temps dans la vue multirésolution. 37

2.32 Dans la première image, une vue multirésolution représente correctement la transition (b) entre une zone de contexte (a) et une zone détaillée (c). Dans l'image du milieu, les zones de transition (b) ne respectent pas l'exigence */E2/*. La troisième image montre l'importance de l'utilisation de la zone de transition (b) où la zone de transition à gauche de la zone détaillée (c) n'est pas affichée. 39

2.33 L'importance d'une interpolation des couleurs dans les zones de transition (b) dans la vue multirésolution. 40

2.34 Techniques d'interaction pour les séries temporelles hiérarchiques sur un streamgraph. (a) Modification de la ligne de base dans un streamgraph (stacked graphs ou streamgraph). (b) Vue d'ensemble+détail. (c) Distorsion fisheye. (d) Combinaison entre une vue d'ensembre+détail et une distorsion fisheye. 42

2.35 Un streamgraph de longues séries temporelles limité par la taille de l'écran. 43

2.36 Évolution des émotions exprimées dans les tweets le jour de l'élection présidentielle des États-Unis en 2016: pic de tristesse à 19h00. 47

2.37 Évolution des émotions exprimées dans les tweets le jour de l'élection présidentielle des États-Unis en 2016: les résultats. 48

2.38 Évolution des genres musicaux de 1960 à 2016: focus sur la période 1975-1995. 51

2.39 Évolution des genres musicaux de 1960 à 2016: navigation à travers les sous-genres du *rock* à différents niveaux de détail. 52

2.40 Évolution des genres musicaux de 1960 à 2016: focus sur la période 2000-2010. 53

3.1 L'interface de GRAPHITE [13]. A gauche, la requête est construite visuellement. A droite, un panneau montre les résultats un par un. . . 58

3.2 L'interface de VISAGE [58]. A gauche, la requête est construite. A droite, les résultats sont affichés sous la forme d'une liste. 58

3.3 L'interface de VIGOR [57]. En haut, la requête est exprimée en langage Cypher. A gauche, la requête est transformée en une requête visuelle. A droite, les résultats sont affichés sous la forme des clusters. 59

3.4 GraphVista [55] permet de visualiser un à un les résultats dès qu'ils sont disponibles. 59

3.5 VOGUE [6] permet de guider la construction de requêtes dans une base de données comportant plusieurs graphes. 60

3.6 VIIQ [40]. Le panneau de suggestion est présenté à droite de la requête. 61

3.7 La *vue de la requête* affichant un exemple de requête d'un graphe multicouches. (a) La requête est représentée dans l'espace de dessin. (b) La barre de conception fournit des interactions pour construire la requête. (c) La barre standard fournit des fonctionnalités classiques et contient également le bouton *Search* qui lance la requête sur le moteur de recherche. (d) La barre d'état affiche des informations textuelles sur la vue. (e) L'utilisateur peut spécifier un attribut sur un sommet. 65

3.8 Un exemple de suggestion d'une requête. Des éléments visuels affichent la suggestion des arêtes. 67

3.9 La requête de la Figure 3.8 incluant des arêtes suggérées et valides. . 68

3.10 La *vue du graphe* affichant un jeu de données biologiques. La visualisation obtenue permet d'identifier facilement quatre communautés. . . 69

3.11 La *vue des résultats* affiche une liste de résultats (b) pour un ensemble de sommets (a). (c)(d) Exemples de résultats fusionnés. La saturation des couleurs des sommets indique le nombre de résultats auxquels le sommet appartient. L'épaisseur des arêtes représente le nombre d'agrégations. 71

3.12 Construction d'un résultat fusionné (c) à partir de la fusion des occurrences (a) et (b). L'épaisseur des arêtes montre le nombre d'agrégations. 71

3.13 L'architecture du système VERTIGo. Les flèches montrent les interactions entre les composants. 73

3.14 La *vue du graphe* affichant le jeu de données biologique de la Figure 3.10. Une carte de chaleur affiche les emplacements des résultats récupérés sur le graphe. 75

3.15 La vue du graphe affichant les sommets des résultats en surbrillance. La taille des sommets représente le nombre de résultats auxquels ils appartiennent. (a) Un champ de texte permet de rechercher un sommet spécifique à partir de son nom. 76

3.16 Approches pour visualiser un ensemble de sommets : (a) BubbleSets (b) LineSets et (c) Diagrammes Kelp. 78

3.17 Processus de suppression de chevauchements de sommets-liens : (a) Détection des chevauchements de sommets (b) Suppression des chevauchements des sommets en utilisant une approche basée sur une triangulation de Delaunay. (c) Ajout d'arêtes entre les sommets (orange) et détection de chevauchements possibles de ces arêtes avec les sommets (boîte rouge) (d) Détermination de la position des sommets de contrôle fixes (en rouge) (e) Les sommets de contrôle fixes sont utilisés pour recalculer la position des sommets de façon à supprimer les chevauchements sommets-liens. 81

3.18 Styles imbriqués basés sur les diagrammes Kelp: (a) les arêtes sont colorées et visibles (b) les arêtes ne sont pas affichées. 82

3.19 Visualisation du jeu de données DBLP dans la vue du graphe. Nous pouvons constater deux zones denses dans la disposition des sommets du graphe correspondant aux deux communautés de recherche représentées. 84

3.20 Processus d'affinement de requête grâce aux suggestions fournies par VERTIGo 85

3.21 (a) L'histogramme montre qu'un grand nombre de résultats ont une faible valeur de MBR. (b) Résultats après filtrage des valeurs faibles de MBR. 87

3.22 Les diagrammes Kelp associés aux résultats sélectionnés dans la liste de droite. 88

Liste des tables

2.1 Comparaison de diverses approches basées sur des stacked graphs et des techniques d'interaction pour afficher des séries temporelles hiérarchiques. 20

3.1 Comparaison des diverses approches pour leur capacité à prendre en charge les tâches de traitement de requêtes. 61

Chapitre

1

Introduction

Sommaire

1.1 Contexte 2
1.2 Contributions 3
1.3 Organisation du mémoire 5

1.1 Contexte

Avec le développement des nouvelles technologies, les données disponibles ne cessent, non seulement, de croître mais arrivent aussi de plus en plus rapidement. Il suffit pour s'en convaincre de regarder le développement des réseaux sociaux comme Facebook [1] où, chaque minute, plus 5 millions de commentaires sont postés et 1,5 millions de photos sont téléchargées [2], ou de services de microblogging comme Twitter [3] où le nombre de tweets émis par jour en 2017 approchait les 500 millions [4]. Les nouveaux objets connectés ou les capteurs de mesures qui collectent de nombreuses données pour des domaines aussi variés que la météorologie, la climatologie ou la santé génèrent également de très nombreuses données à un grand débit.

Outre la vitesse, il peut exister de nombreuses relations intrinsèques associées aux données. Par exemple, dans le domaine des réseaux sociaux, des applications sont mises à disposition des utilisateurs pour leur permettre d'échanger des messages, des photos ou des vidéos. Ces échanges sont liés à des relations créées par les utilisateurs entre eux. Par exemple, un utilisateur de Facebook peut diffuser un message à une liste d'amis, un utilisateur Twitter peut suivre les messages diffusés par un autre utilisateur en s'abonnant à sa page, etc. Un même utilisateur peut aussi utiliser différents réseaux sociaux. Par exemple, il pourra utiliser Linkedin [5] pour ses activités professionnelles et Facebook pour ses activités personnelles. Une nouvelle complexité apparaît avec le fait que ces réseaux se retrouvent de plus en plus interconnectés. La Figure 1.1 illustre, sous la forme de graphes, des utilisateurs reliés par différents types de relations. A gauche, les personnes sont reliées entre elles par le lien d'amitié de Facebook. A droite, les personnes possèdent des relations issues de différents réseaux sociaux.

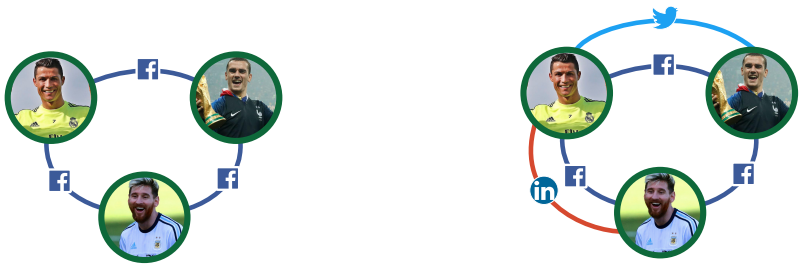


FIGURE 1.1 – A gauche, des membres sont liés via une relation d'amitié sur Facebook. A droite, les membres sont liés par plusieurs arêtes représentant les relations issues de trois médias sociaux.

1. https://facebook.com
2. https://zephoria.com/top-15-valuable-facebook-statistics/
3. https://twitter.com
4. https://www.omnicoreagency.com/twitter-statistics/
5. https://www.linkedin.com/

L'analyse de données de plus en plus complexes, volumineuses et issues de différentes sources est une tâche difficile. Elle reste cependant cruciale dans de très nombreux domaines d'application comme le marketing, la politique, la santé, l'environnement, etc. Elle implique, pour pouvoir en extraire des connaissances, de mieux comprendre la nature des données, leur évolution ou les nombreuses relations complexes qu'elles peuvent contenir. Cette analyse soulève de nombreuses questions comme par exemple : quels sont les mécanismes qui permettent de comprendre facilement les données ? Comment interagir avec les données pour pouvoir en extraire de l'information ? Pour aider à répondre à ces questions la visualisation d'informations s'avère particulièrement utile.

La **visualisation d'informations** permet de produire des représentations visuelles de données abstraites pour permettre d'en améliorer la compréhension. L'une des difficultés est de déterminer quelle représentation choisir pour bien répartir les données abstraites dans l'espace et montrer un message cohérent et pertinent qui tient compte des capacités visuelles et cognitives de l'être humain. Lorsque les données sont trop volumineuses, une simple visualisation n'est souvent pas suffisante pour en restituer la totalité. Dans ce cas, des techniques d'interaction sont utilisées pour enrichir les différentes tâches : navigation, exploration, filtrage, sélection. Enfin le choix des encodages visuels et des interactions à retenir est un véritable défi dans la mesure où celles-ci doivent être adaptées aux besoins des utilisateurs.

Dans le cadre de ce mémoire, nous nous intéressons plus particulièrement à deux types de données. Les premières sont des données dynamiques qui s'expriment sous la forme de multiples séries temporelles qui possèdent une structure hiérarchique. Les secondes correspondent à des graphes volumineux multicouches, *i.e.* pour lesquels il peut exister différents types d'arêtes. L'objectif de cette thèse est d'utiliser la visualisation d'informations pour proposer de nouvelles approches afin d'aider l'utilisateur à analyser et mieux comprendre les données.

1.2 Contributions

Dans ce mémoire, pour répondre aux problématiques de l'analyse de données dynamiques et complexes, nous proposons deux nouvelles approches visuelles : MultiStream [19, 20, 50] et VERTIGo [17, 18].

MultiStream (Cf. Figure 1.2) est une plateforme visuelle développée pour faciliter l'analyse de multiples séries temporelles. Cette approche facilite l'exploration et la comparaison de l'évolution de séries temporelles à différents niveaux de granularité grâce à quatre composants visuels interactifs : une *vue globale* (Cf. Figure 1.2a) qui présente les principales tendances du jeu de données, une *vue multirésolution* (Cf. Figure 1.2b) qui montre des détails sur un sous-ensemble du jeu de données, un

contrôleur (Cf. Figure 1.2c) qui permet de sélectionner des périodes spécifiques et un *gestionnaire de hiérarchie* (Cf. Figure 1.2d) qui permet de naviguer/explorer la structure hiérarchique.

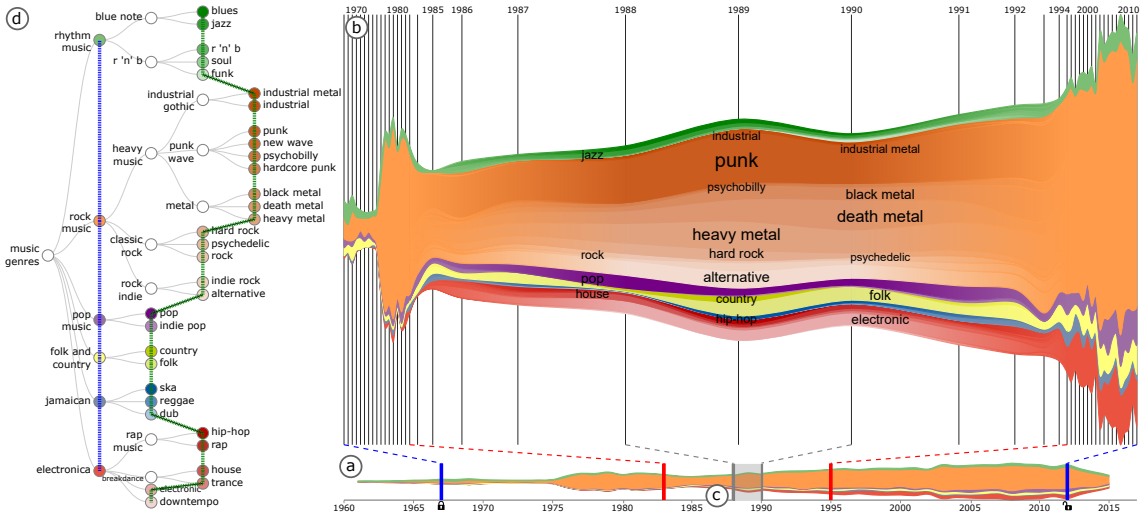


FIGURE 1.2 – L'approche MULTISTREAM est utilisée pour visualiser l'évolution de séries temporelles hiérarchiques. (a) Une vue globale représente les séries à un haut niveau d'abstraction. (b) Une vue multirésolution représente les séries à différents niveaux d'abstraction. (c) Un contrôleur permet de sélectionner un intervalle de temps et relie la vue globale et la vue multirésolution. (d) Un gestionnaire de hiérarchie permet de naviguer dans la structure hiérarchique des séries.

La deuxième contribution, VERTIGO (Cf. Figure 1.3), est une plateforme visuelle pour l'analyse de données complexes contenant différents types de relations et modélisées sous la forme de graphes multicouches. VERTIGO permet de définir un graphe multicouches, puis de rechercher et d'explorer interactivement toutes les occurrences de ce graphe dans un jeu de données volumineux. Pour cela, trois composants visuels principaux constituent le cœur de la plateforme : une *vue de la requête* (Cf. Figure 1.3a) permet le dessin/suggestion interactif d'une requête. Une *vue du graphe*(Cf. Figure 1.3(b, e)) montre la structure du graphe et permet de naviguer/explorer les résultats à différents niveaux de détail. Une *vue des résultats* (Cf. Figure 1.3d) permet de visualiser une liste de résultats pour d'autres analyses (*e.g.* spatiales ou topologiques).

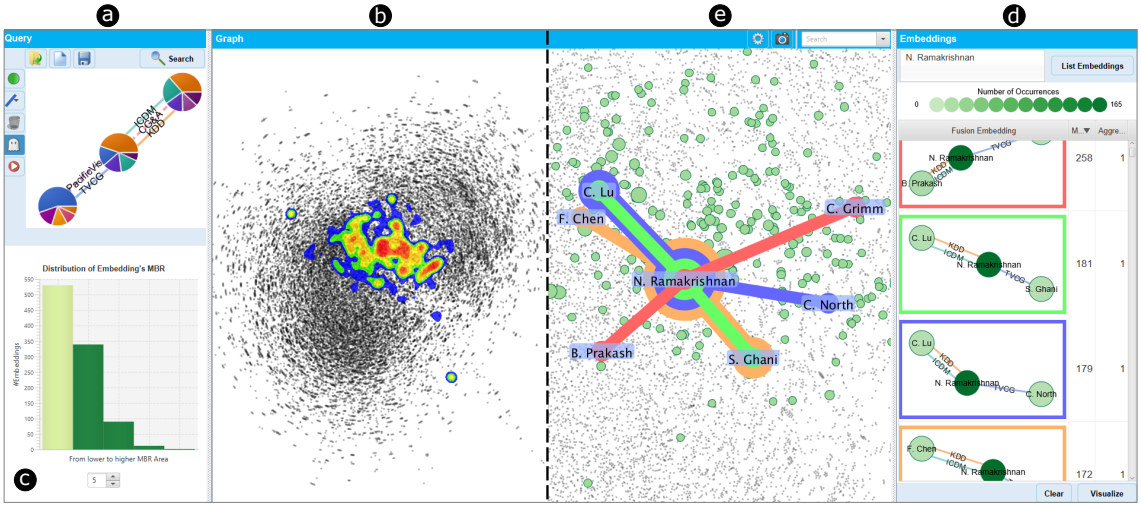


FIGURE 1.3 – L'approche VERTIGO est utilisée pour explorer un réseau de co-auteurs. (a) Une vue de la requête permet de construire visuellement une requête et de l'exécuter sur moteur de recherche pour extraire des résultats. (b) Une vue du graphe montre les emplacements des résultats grâce à une carte de chaleur. (c) Un histogramme permet de filtrer les résultats en fonction de la taille de leur boîte englobante. (d) Une vue de résultats montre les occurrences pour un ensemble d'entités sélectionnées, *e.g.* les résultats contenant l'auteur N. Ramakrishnan. (e) A un niveau de zoom élevé, la vue du graphe montre les relations d'un sous-ensemble de résultats sélectionnés grâce à un diagramme Kelp.

1.3 Organisation du mémoire

Ce mémoire est organisé de la manière suivante : ce chapitre présente le contexte de la problématique de recherche que nous traitons, et les principales contributions proposées. Le chapitre 2 et le chapitre 3 décrivent nos contributions. Dans le chapitre 2 nous abordons la problématique de la visualisation de séries temporelles multiples. Nous mettons en évidence des critères adaptés à l'analyse de ce type de données. Les différentes vues de l'approche MULTISTREAM sont décrites et analysées. Deux exemples d'utilisation, l'un basé sur des données composées de sentiments exprimés dans des tweets, l'autre contenant l'évolution des genres musicaux dans le temps, permettent d'illustrer l'approche dans des domaines différents. Ensuite, le chapitre 3 aborde l'analyse d'objets complexes modélisés grâce à des graphes multicouches. Comme dans le chapitre précédent, nous mettons en avant des critères permettant de faciliter l'analyse de ce type de données. Les différentes vues de l'approche VERTIGO sont présentées et analysées. Un exemple de recherche, d'analyse et de suggestions basé sur des données DBLP (communautés visualisation et fouille

de données / bases de données) permet d'illustrer l'utilisation des différentes vues. Finalement, dans le chapitre 4, nous concluons et proposons des perspectives de recherche associées à nos travaux.

CHAPITRE

2 Visualisation de séries temporelles hiérarchiques

Sommaire

2.1 Introduction 8
2.2 État de l'art 12
2.2.1 Visualisations de type stacked graphs et streamgraphs . . 13
2.2.2 Structure hiérarchique dans les séries temporelles 15
2.2.3 Techniques d'interaction 16
2.3 Critères 21
2.4 Encodages visuels et méthodes d'interaction 22
2.4.1 Conception visuelle 23
2.4.2 Vue globale 23
2.4.3 Vue multirésolution 24
2.4.4 Le contrôleur 26
2.4.5 Gestionnaire de hiérarchie 29
2.5 Contraintes techniques pour la vue multirésolution . . . 35
2.5.1 Longueurs de pas de temps 35
2.5.2 Les couleurs 38
2.6 Discussion 41
2.6.1 Une approche flexible 41
2.6.2 Comparaison avec des techniques alternatives 43
2.7 Exemples 44
2.7.1 L'élection présidentielle des États-Unis en 2016 45
2.7.2 L'évolution des genres musicaux 49
2.8 Conclusion 54

2.1 Introduction

Comme nous l'avons vu dans l'introduction, pour aider l'utilisateur dans la compréhension des données dynamiques, il est indispensable de lui offrir des outils adaptés. Dans ce chapitre nous abordons la problématique de la visualisation d'informations lorsque les données dynamiques peuvent être modélisées sous la forme de séries temporelles.

Une **série temporelle** (ou série chronologique) est définie comme une séquence de valeurs quantitatives prises à des moments successifs. Généralement l'évolution de ces valeurs au cours du temps offre des informations utiles pour analyser les données (*i.e.* motifs, pics, etc.).

Les données disponibles sous la forme de séries temporelles sont courantes dans de très nombreux domaines comme la médecine [49], la météorologie et la climatologie [64], la finance [5, 67], les réseaux sociaux [53], etc. En météorologie, elles correspondent, par exemple, à l'enregistrement des quantités de pluie sur une période de temps. En finance, elles peuvent correspondent à l'évolution du cours de la monnaie ou des actions au cours du temps. Dans le domaine du marketing, le fait de suivre le nombre de chansons entendues par les utilisateurs au cours du temps peut être une information utile pour une société de streaming musical.

Une approche traditionnelle pour représenter ces séries est d'utiliser un système de coordonnées cartésiennes en deux dimensions où l'axe horizontal encode la dimension temporelle et l'axe vertical la dimension quantitative.

Habituellement, les diagrammes linéaires sont bien adaptés pour représenter une seule série temporelle (le lecteur intéressé peut se référer à Harris *et al.* [33]). La Figure 2.1 illustre un tel diagramme : la position x des points représent le temps, leur position y représent leur valeur; les points sont ensuite reliés. Nous pouvons visualiser dans cette figure le nombre de fois qu'un utilisateur a écouté des chansons du genre musical *alternative metal* pendant une année. Nous remarquons que cette représentation visuelle montre clairement l'évolution de la série temporelle.

Cependant, la plupart des jeux des données sont composés de plusieurs séries temporelles (*e.g.* différents genres musicaux, cours des différentes monnaies, etc.). les **séries temporelles multiples** sont définies comme un ensemble de variables quantitatives suivies sur un même intervalle de temps, c'est-à-dire un ensemble de séries temporelles.

Les diagrammes linéaires souffrent d'un problème de surcharge lorsque le nombre de séries est important. Par exemple, la Figure 2.2 illustre le nombre de fois qu'un utilisateur a écouté des chansons de différents genres musicaux (*e.g. alternative metal*, *black metal*, *doom metal*, *classic jazz*, *soul jazz*, *contemporary jazz*, etc.) pendant

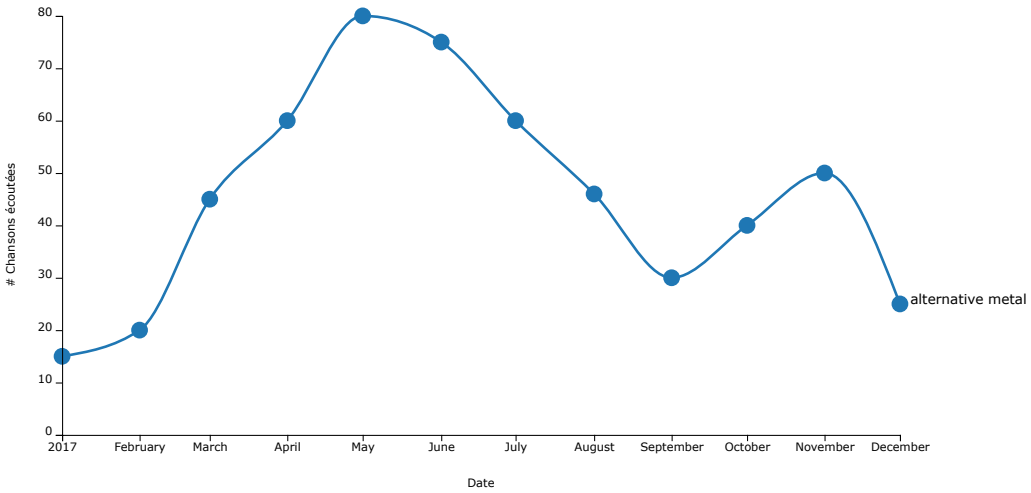


FIGURE 2.1 – Un diagramme linéaire montrant l'évolution du nombre de chansons écoutées du genre musical *alternative metal* par un utilisateur.

une année. Nous voyons que lorsque le nombre de séries augmente, il devient difficile de les visualiser. Pour faire face à ce problème, différentes représentations visuelles ont été proposées, telles que les *stacked graphs* [33] et les *streamgraphs* [10].

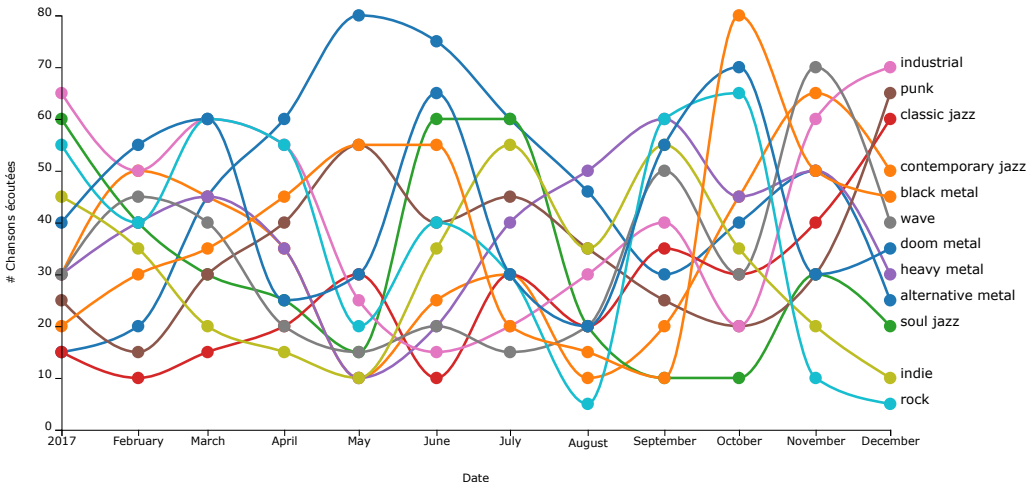
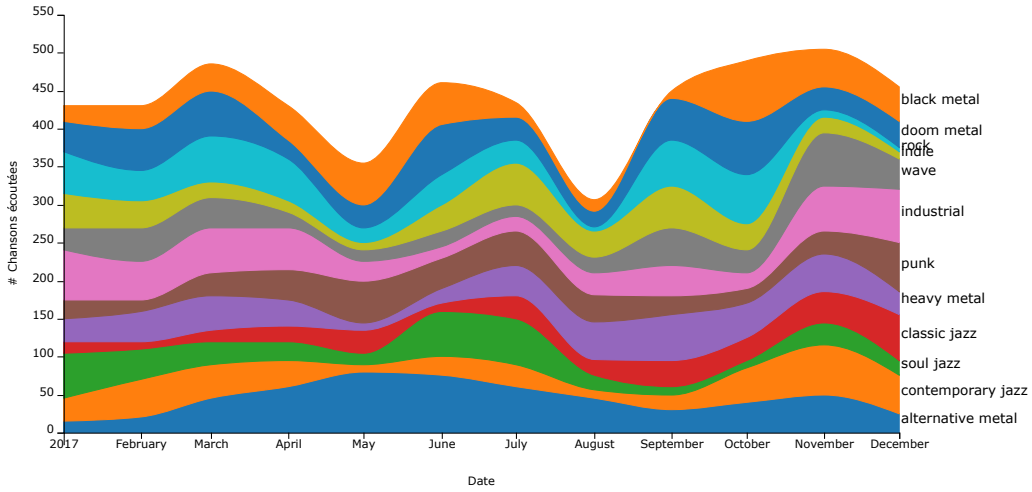


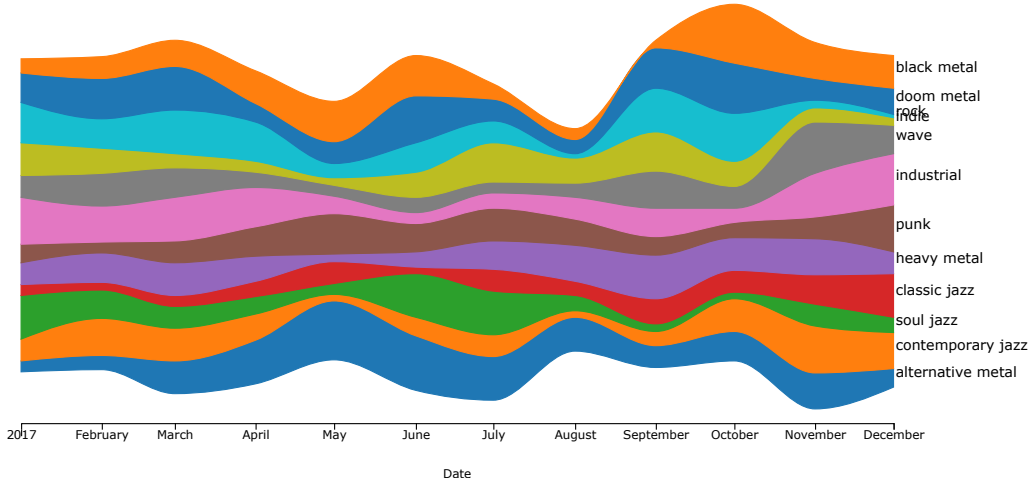
FIGURE 2.2 – Un diagramme linéaire montrant l'évolution du nombre de chansons écoutées par un utilisateur selon leur genre musical : *alternative metal*, *black metal*, *doom metal*, *classic jazz*, etc.

Les stacked graphs (Cf. Figure 2.3a) et les streamgraphs (Cf. Figure 2.3b) sont des représentations visuelles qui encodent aussi la dimension temporelle sur l'axe horizontal mais qui empilent les séries en **couches** sur l'axe vertical. Pour empiler les couches, les stacked graphs utilisent comme ligne de base l'axe du temps, tandis que

les streamgraphs utilisent une ligne de base centrale parallèle à l'axe du temps. La Figure 2.3 montre un stacked graph (a) et un streamgraph (b) des genres musicaux présentés dans la Figure 2.2.



(a)



(b)

FIGURE 2.3 – Un stacked graph (a) et un streamgraph (b) montrant l'évolution du nombre de chansons par genre musical écoutés par un utilisateur.

Même si ces deux représentations visuelles offrent une meilleure perception de l'évolution des séries, elles sont cependant pénalisées lorsque le nombre de séries est trop important. La Figure 2.4 illustre un streamgraph avec une centaine de séries (couches) qu'il est difficile d'analyser. Pour pallier ce problème, différentes approches proposent d'agréger les séries dans une structure hiérarchique afin de représenter l'information à différents niveaux d'abstraction.

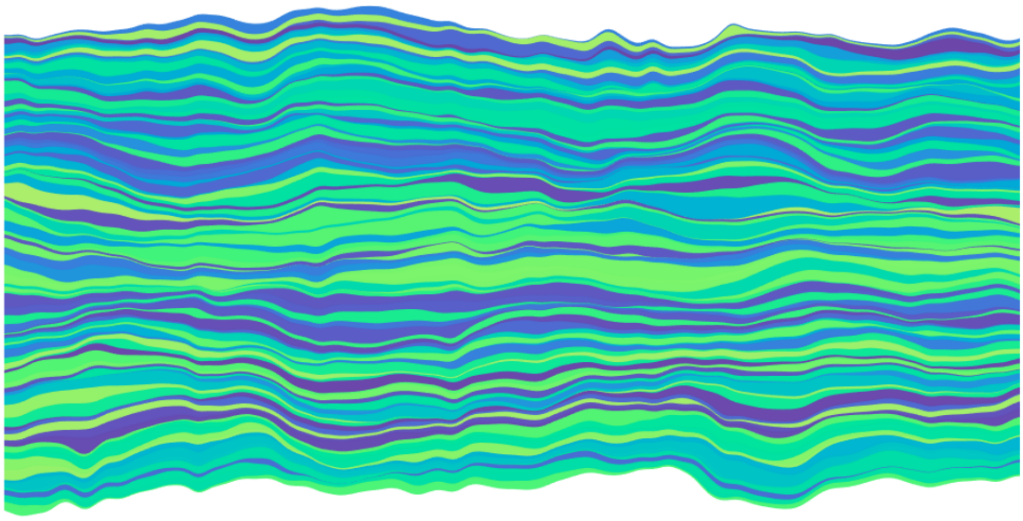


FIGURE 2.4 – Problème de surcharge d'un streamgraph avec une centaine de couches (séries temporelles) qu'il est difficile d'analyser.

La **structure hiérarchique des séries temporelles** est définie comme un ensemble de séries qui peuvent être organisées à l'intérieur d'une structure hiérarchique selon leur nature. Dans cette structure les niveaux agrégés sont égaux à la somme des éléments qu'ils contiennent. Par exemple, les genres musicaux peuvent être le résultat de l'agrégation de plusieurs sous-genres (*e.g.* le genre *metal* est l'agrégation de *alternative metal*, *black metal*, *doom metal*, etc.). La Figure 2.5b illustre le genre *metal* issu de l'agrégation des genres musicaux de la Figure 2.5a. Ainsi, l'organisation des séries temporelles en une structure hiérarchique atténue le problème de surcharge des représentations de type stacked graphs.

Lorsque le nombre de séries temporelles augmente, un nouveau défi technique se pose: comment interagir et extraire des informations ? Pour aider les utilisateurs dans cette tâche, de nombreuses techniques d'interaction connues peuvent être adaptées à une visualisation basée sur les stacked graphs: vue d'ensemble+détails [14], focus+contexte [14], *fisheye* [63], zoom [44], etc.

Dans ce chapitre, nous proposons une nouvelle approche nommée MULTISTREAM[1] [19, 20, 50] (Cf. Figure 1.2). Elle est basée sur une représentation de type streamgraph prenant en compte la structure hiérarchique de séries temporelles. Plus précisément, notre approche combine diverses techniques d'interaction pour étendre les avantages d'un streamgraph et ainsi faciliter l'exploration et l'analyse de séries temporelles. En conséquence, l'utilisateur final dispose d'un outil dynamique permettant d'explorer différents niveaux d'abstraction de la structure hiérarchique des séries.

1. http://advanse.lirmm.fr/multistream/ [dernier accès 01/07/2018]

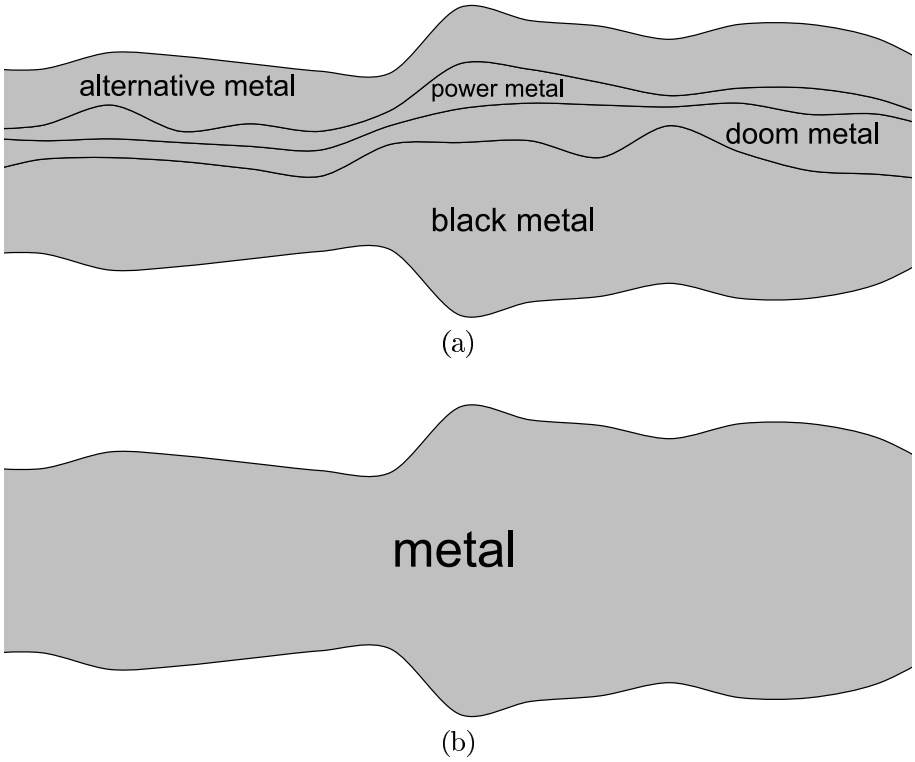


FIGURE 2.5 – Agrégation de séries temporelles. (a) Le streamgraph montre l'évolution des genres: *alternative metal*, *power metal*, *doom metal* et *black metal*. (b) Le streamgraph montre l'agrégation des genres de la Figure (a) pour le genre *metal*.

Ce chapitre est organisé de la manière suivante : l'état de l'art est présenté et discuté dans la section 2.2. Des critères pour l'outil sont définis dans la section 2.3. Dans la section 2.4, nous décrivons les encodages visuels et les différents modes d'interaction. Les considérations techniques sont présentées dans la section 2.5. Une discussion de notre approche est présentée dans la section 2.6 Deux exemples d'utilisation de MULTISTREAM sont proposés dans la section 2.7. Enfin, nous concluons ce chapitre dans la section 2.8.

2.2 État de l'art

De nombreuses approches ont été proposées en visualisation pour représenter des séries temporelles multiples. Aigner *et al.* [1] propose un état de l'art des principales techniques de visualisation de données qui évoluent au cours du temps. Dans cette section, nous nous concentrons sur les techniques de visualisation et d'interaction basées sur les représentations de type stacked graphs, ainsi que sur une structure hiérarchique incorporées dans la représentation visuelle.

2.2.1 Visualisations de type stacked graphs et streamgraphs

Comme nous l'avons vu précédemment, une approche traditionnellement utilisée pour représenter des séries temporelles multiples est appelée stacked graphs [33] (Cf. Figure 2.3a). Dans cette représentation, l'axe horizontal représente le temps et les séries sont empilées les unes sur les autres sur l'axe. Chaque série est ainsi visualisée en tant que *couche* colorée. L'épaisseur des couches représente la valeur quantitative à un pas de temps donné. Enfin, la direction des couches de gauche à droite indique l'évolution dans le temps et l'épaisseur des couches agrégées reflète la somme des séries temporelles.

La Figure 2.6 illustre l'approche stacked graphs utilisée par TIARA [47] pour visualiser le contenu thématique d'une collection de documents. Un autre exemple d'utilisation de stacked graphs se trouve dans NameVoyager [71] (Cf. Figure 2.7). Cette approche utilise un stacked graphs avec des mécanismes d'interaction pour explorer au cours du temps le choix des prénoms d'enfants.

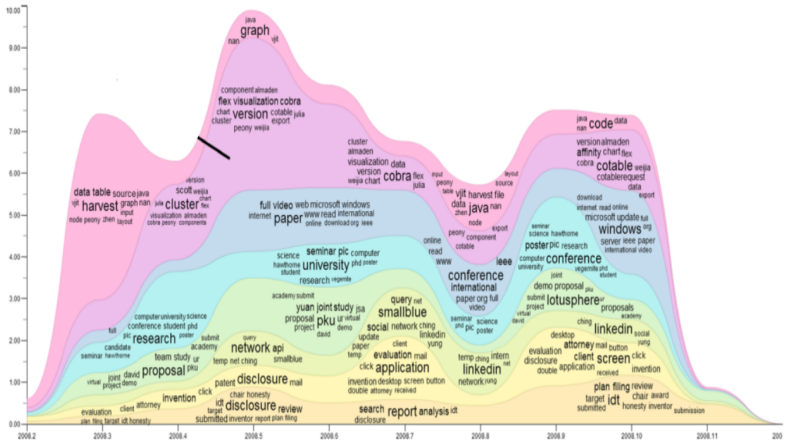


FIGURE 2.6 – TIARA [47] utilise une visualisation de type stacked graph pour représenter l'évolution du contenu thématique d'une collection de documents.

Différentes extensions des stacked graphs ont été proposées. Par exemple, dans ThemeRiver [34, 35] (Cf. Figure 2.8), les séries sont représentées le long d'une ligne de base centrale parallèle à l'axe de temps et les couches sont réparties le long de cet axe. Les streamgraphs [10] (Cf. Figure 2.9) se servent également d'une ligne de base centrale, mais utilisent un *décalage* pour minimiser la variation pondérée de la pente par rapport à l'épaisseur de la couche. Cette différence rend les flux plus lisses et plus naturels que ceux obtenues par ThemeRiver. Les streamgraphs sont utilisés par exemple dans le domaine de visualisation du mouvement du corps [59] ou dans l'évolution de données en temps réel (*e.g.* tweets, RSS, emails, etc.) [36].

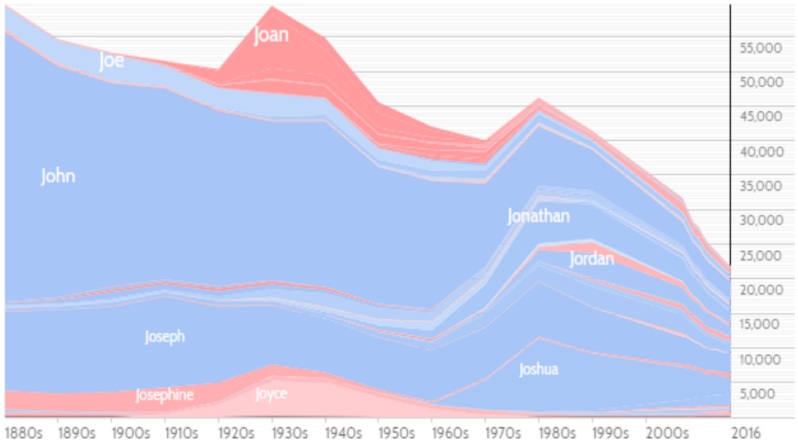


FIGURE 2.7 – NameVoyager [71] adopte une visualisation de type stacked graph pour explorer les tendances sur le choix des prénoms d'enfants au cours du temps.

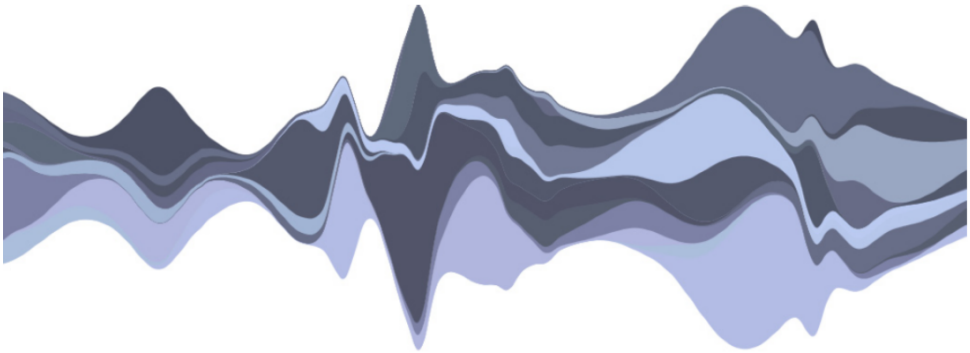


FIGURE 2.8 – Les ThemeRivers [34,35] utilisent une ligne de base centrale parallèle à l'axe temporel et les couches sont reparties le long de cet axe simulant ainsi le flux d'une rivière.

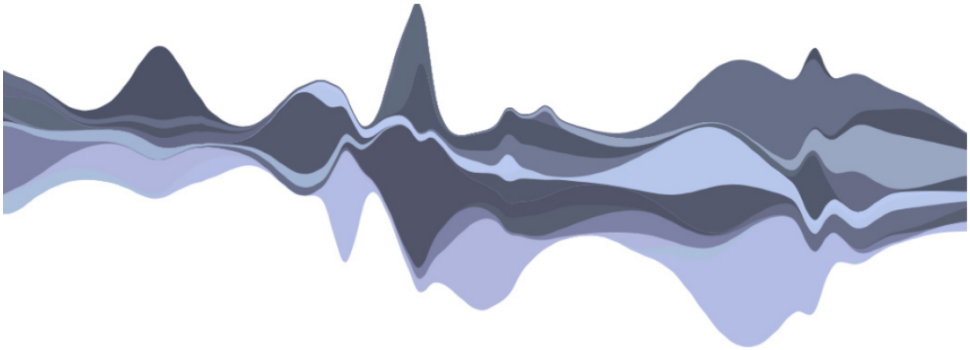


FIGURE 2.9 – Les Streamgraphs [10] permettent d'obtenir une représentation plus « lisse » par rapport à la représentation de type ThemeRiver.

Cependant, comme nous l'avons vu précédemment ces approches souffrent d'un problème lorsque le nombre de séries devient important (Cf. Figure 2.4). Afin de résoudre ce problème, certaines approches organisent les séries temporelles dans une structure hiérarchique.

2.2.2 Structure hiérarchique dans les séries temporelles

Le regroupement des séries dans une organisation hiérarchique dépend bien entendu de leur nature. Cette organisation hiérarchique est présente dans divers domaines. Par exemple, BookVoyager [72] (Cf. Figure 2.10) utilise un stacked graph pour visualiser l'historique des ventes de livres organisés dans une hiérarchie de catégories et de sous-catégories. ManyEyes [68] (Cf. Figure 2.11) propose un « stacked graph par catégories » pour montrer la répartition du budget fédéral historique des États-Unis dans les différents services (*e.g.* la défense générale, les armes atomiques, etc). Dans TouchWave [3] (Cf. Figure 2.12), un streamgraph montre un historique d'écoute de musiques organisées dans une hiérarchie de genres, d'artistes et de chansons.

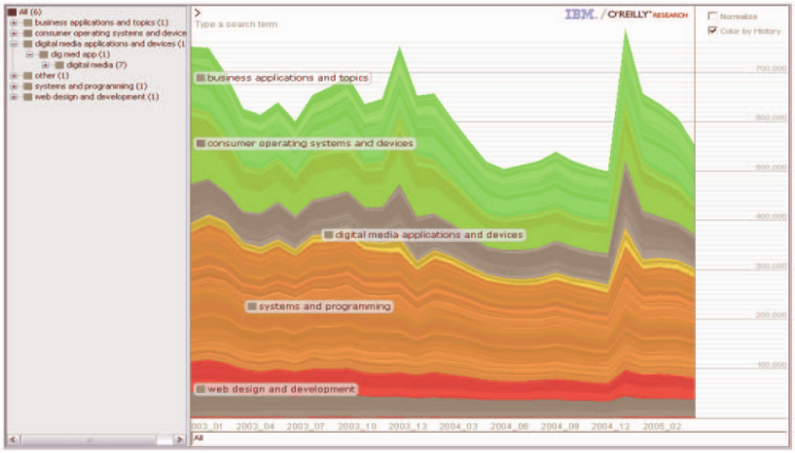


FIGURE 2.10 – BookVoyager [72] utilise un stacked graph pour visualiser l'évolution d'une hiérarchie de catégories et de sous-catégories de livres.

Plus récemment, la structure hiérarchique des séries temporelles a été utilisée dans des domaines thématiques. Par exemple, LensRiver sur NewsLab [29] (Cf. Figure 2.13) permettent d'explorer les relations entre les mots-clés de sujets organisés hiérarchiquement. HierarchicalTopics [24] (Cf. Figure 2.14) développe une visualisation basée sur ThemeRiver pour représenter des motifs temporels des sujets organisés en catégories. RoseRiver [22] (Cf. Figure 2.15) étend l'approche TextFlow [21] pour analyser les relations entre les sujets dans une structure hiérarchique.

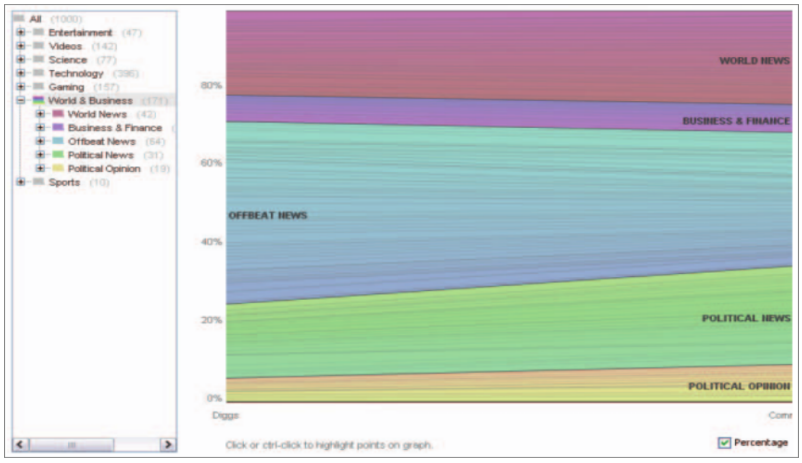


FIGURE 2.11 – ManyEyes [68] affiche sur un stacked graph la répartition du budget fédéral historique des États-Unis.

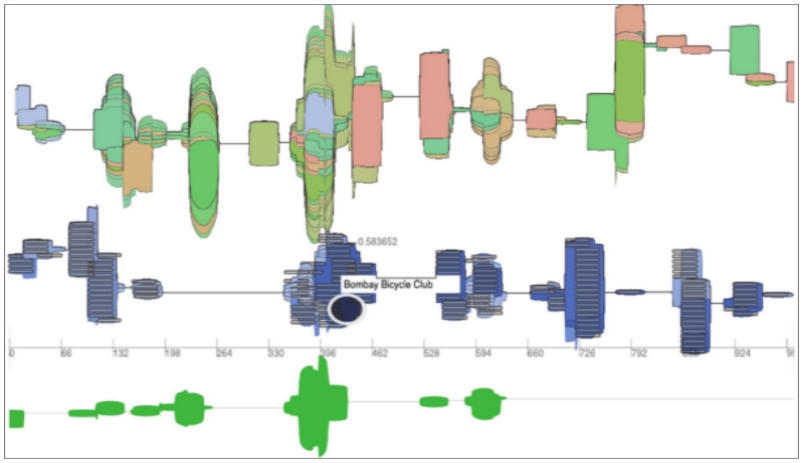


FIGURE 2.12 – TouchWave [3] utilise un streamgraph pour représenter un historique musical.

Toutes les approches mentionnées ci-dessus utilisent l'ordre des couches et les couleurs pour relier les séries temporelles individuelles d'un groupe. Afin d'interagir avec les séries temporelles (*e.g.* filtrer, survoler, explorer), des techniques d'interaction sont utilisés.

2.2.3 Techniques d'interaction

Les techniques d'interaction sont utilisées pour améliorer l'efficacité d'une visualisation. Il en existe un grand nombre comme par exemple: *zoom* [44], *focus+contexte* [27], *vue d'ensemble+détail* [14], *brushing&linking* [4], etc. Chacune de ces techniques interagit avec les données de différentes manières.

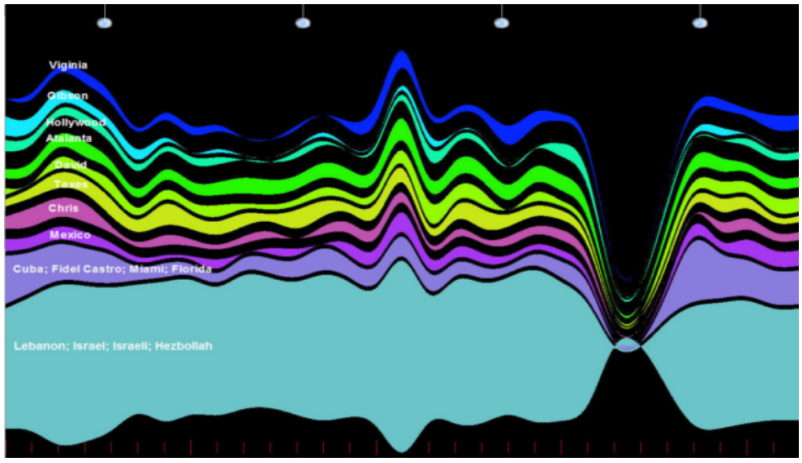


FIGURE 2.13 – NewsLab [29] permet de visualiser les mots-clés d'une collection de vidéos d'actualités organisées hiérarchiquement par thèmes.

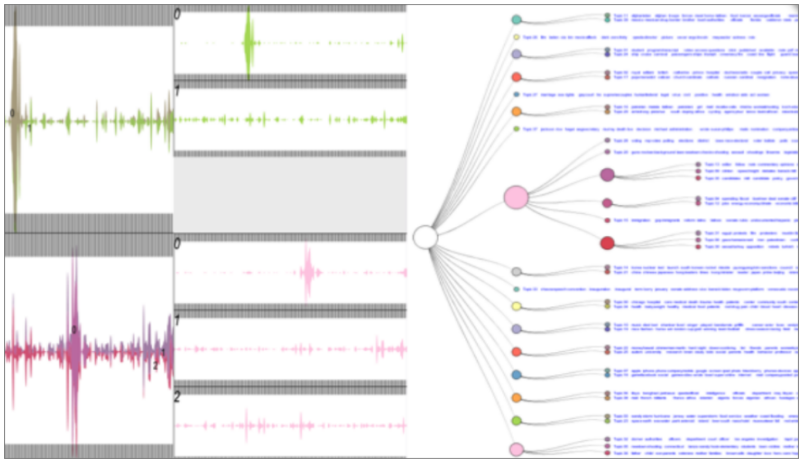


FIGURE 2.14 – HierarchicalTopics [24] permet de visualiser les sujets organisés en catégories et sous-catégories issus d'une collection de textes.

Le **zoom** est une technique qui consiste à montrer une zone spécifique en détail. Par exemple, la Figure 2.16 illustre un dessin des co-occurrences de personnages dans le roman de Victor Hugo « Les Misérables ». La Figure 2.16a montre une vue globale de ce dessin où les détails ne sont pas visibles. Dans la Figure 2.16b nous zoomons sur la position du personnage *Valjean* (en rouge). Nous remarquons qu'un des inconvénients du zoom est qu'en se focalisant sur ce personnage, le contexte est perdu.

La technique dite **focus+contexte** surmonte le problème du zoom en permettant à l'utilisateur de voir dans une même vue une partie des données détaillées (focus) tout en affichant le contexte. La Figure 2.17a reprend l'exemple de la Figure 2.16. Nous voyons dans la même vue le contexte (le réseau des personnages du roman

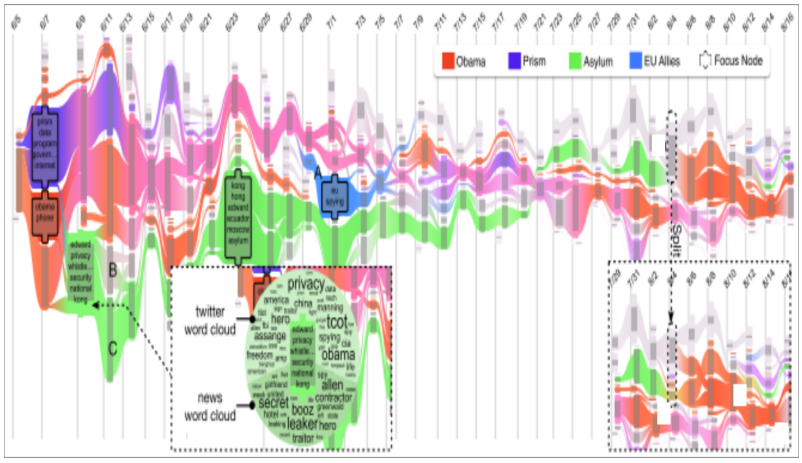
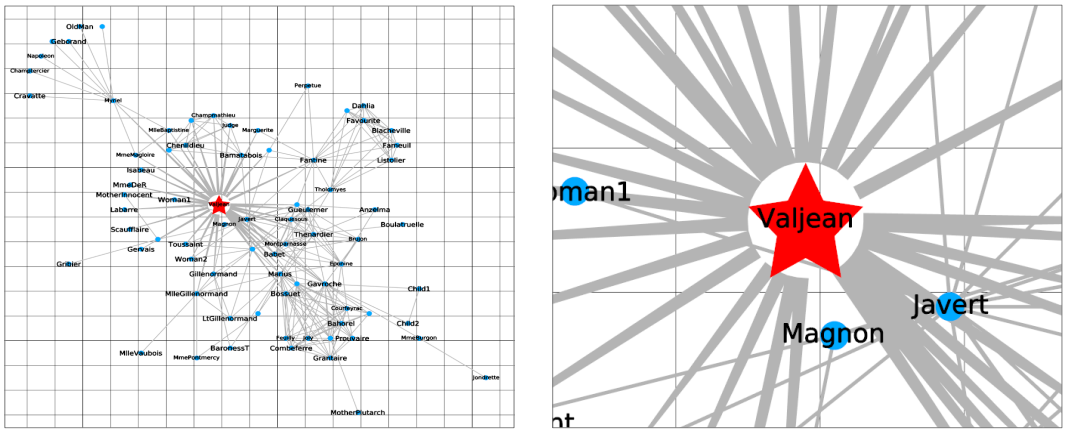


FIGURE 2.15 – RoseRiver [22] visualise les sujets hiérarchisés dans une collection de texte.

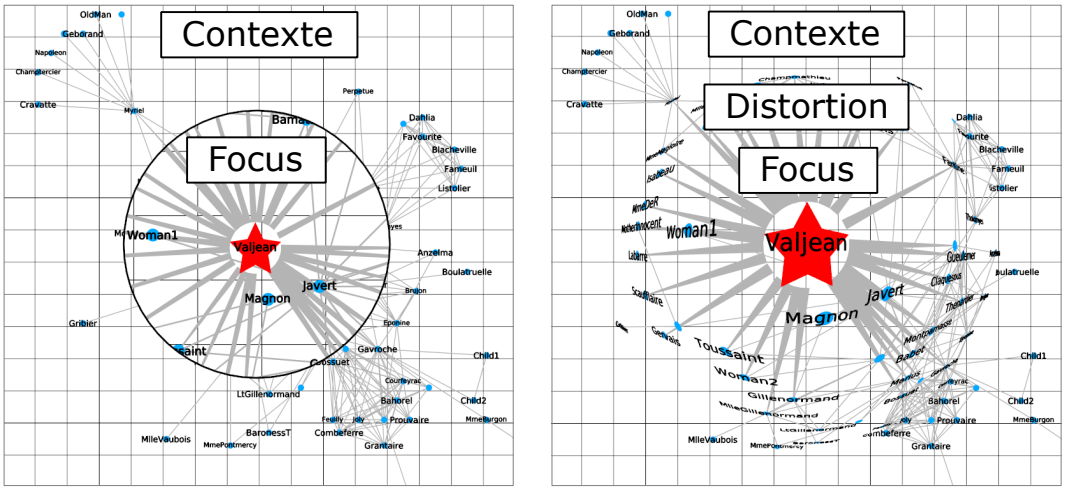


(a) (b)

FIGURE 2.16 – La technique de zoom. (a) Le contexte est affiché mais les détails ne sont pas disponibles. (b) En zoom avant, les détails sont affichés mais le contexte n'est plus disponible.

« Les Misérables ») et le focus (le personnage *Valjean*). Cependant, une partie de l'information est perdue dans la zone de transition entre le focus et le contexte, *i.e.* une partie du réseau est cachée par le focus.

Pour pallier la perte d'information dans la technique du focus+contexte, des **techniques de distorsion** ont été proposées. Ces techniques montrent le focus dans une zone étendue et le contexte dans une zone progressivement diminuée (*e.g.* fisheye [63], PerspectiveWall [48], présentation élastique de l'espace [12]). La Figure 2.17b montre un fisheye qui génère une zone de transition entre le focus et le contexte. La zone de transition est « distordue » pour passer du focus au contexte



(a) (b)

FIGURE 2.17 – (a) Technique dite focus+contexte. (b) Distorsion de type *fisheye*.

et vice versa. Nous voyons que grâce à cette transition toute l'information est affichée. Cependant, parfois, la distorsion présente dans la transition peut provoquer une confusion chez les utilisateurs.

La technique de **vue d'ensemble+détail** surmonte cette confusion en affichant deux vues séparées: une pour la vue d'ensemble et une pour les détails. Ainsi, l'utilisateur peut se concentrer sur les détails sans perdre le contexte. La Figure 2.18 présente cette technique : à gauche une vue dédiée aux détails et à droite une vue dédiée au contexte. Généralement, la technique du brushing&linking est utilisée pour lier ces deux vues. Cette technique consiste à sélectionner (avec la souris) un sous-ensemble des éléments dans la vue d'ensemble qui seront affiches dans la vue des détails.

Nous comparons, dans le Tableau 2.1, les approches basées sur les stacked graphs et les différents techniques d'interaction utilisées. Nous remarquons que certaines approches permettent d'interagir avec des séries organisées hiérarchiquement. Par exemple, BookVoyager [72], ManyEyes [68] et HierarchicalTopics [24] intègrent une vue séparée (*e.g. tree-control*) pour grouper les séries et permettre à l'utilisateur de les explorer. NewsLab [29] utilise une approche basée sur une structure hiérarchique [26] pour explorer un ensemble de thèmes. HierarchicalTopics [24] utilise un arbre pour explorer et fusionner des sujets dans la structure hiérarchique. En revanche, TouchWave [3] propose des stacked graphs « touchables » pour interagir avec la hiérarchie et aider les utilisateurs à accomplir des tâches spécifiques (*e.g.* ordonner les couches, filtrer les couches et changer la disposition des couches).

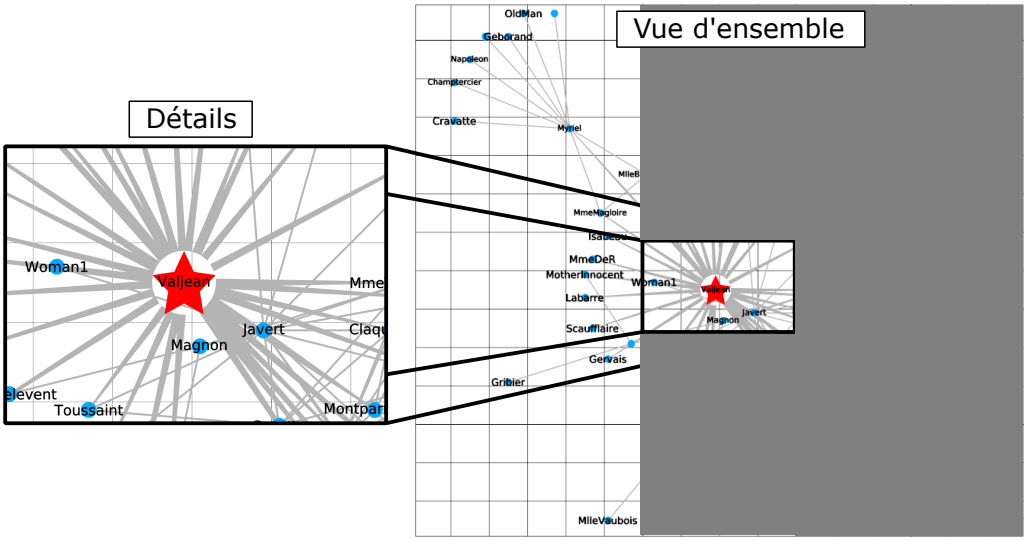


FIGURE 2.18 – La technique de vue d'ensemble+détail composée par deux vues. La vue de gauche montre les détails et la vue de droite montre en même temps le contexte.

TABLE 2.1 – Comparaison de diverses approches basées sur des stacked graphs et des techniques d'interaction pour afficher des séries temporelles hiérarchiques.

Approches	Séries temporelles hiérarchiques	Vue d'ensemble +détail	Zoom	Fisheye
TIARA [47]			x	x
NameVoyager [71]			x	
BookVoyager [72]	x		x	
ManyEyes [68]	x		x	
HierarchicalTopics [24]	x		x	
TouchWave [3]	x		x	x
NewsLab [29]	x	x	x	
MULTISTREAM	x	x	x	x

MULTISTREAM vise à combiner différentes techniques d'interaction (*e.g.* vue d'ensemble+détail, zoom, focus+contexte) pour améliorer l'analyse de la structure hiérarchique des séries temporelles et décrire différents niveaux d'abstraction.

2.3 Critères

Dans le cadre de l'analyse de séries temporelles, des questions intéressantes peuvent être proposées pour extraire des informations pertinentes des données. Nous allons partir d'un exemple d'utilisation des séries afin d'établir une série de critères pour notre approche.

Le scénario est le suivant : dans une société de marketing, l'analyse des sentiments exprimés dans les médias sociaux pourrait aider à définir ou adopter une stratégie de communication. La société choisit Twitter comme réseau social pour récupérer des tweets qui pourraient contenir des émotions. Une modèle affectif, comme par exemple, le modèle circumplex de Russell [25] (Cf. Figure 2.19) peut être utilisé pour classer les émotions. Dans ce cas, chaque série temporelle correspond à une émotion et les valeurs de la série temporelle de l'émotion « X » se forment à partir du nombre de tweets contenant l'émotion « X » dans un intervalle de temps.

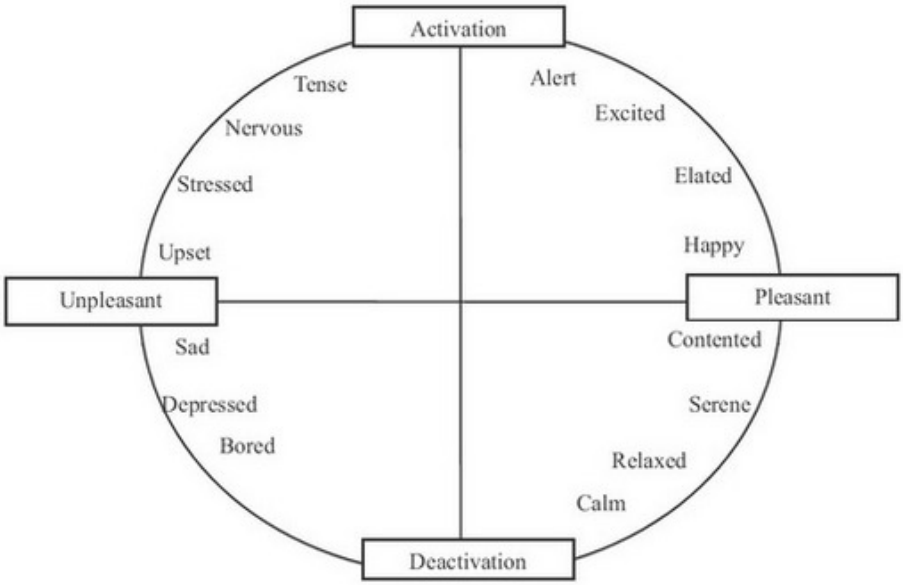


FIGURE 2.19 – Modèle émotionnel circumplex de Russell.

A partir du modèle de Russell nous pouvons définir une structure hiérarchique à 2 niveaux. Par exemple, le groupe *pleasant-activation* regroupe les émotions associées à *happy, elated, exited* et *alert*. Le groupe *unpleasant-deactivation* est constitué d'émotions comme *sad, depressed, bored* et *lethargic*.

Une question fondamentale est de savoir comment représenter cette organisation hiérarchique pour analyser l'expression des sentiments au fil du temps. L'utilisateur peut également être intéressé par l'évolution des sentiments au cours du mois de

lancement d'une campagne (*e.g.* pendant les soldes d'été). Enfin, l'utilisateur pourrait être intéressé par l'affichage de la structure hiérarchique de cette information affective au cours de la période de la campagne.

Afin de répondre efficacement à ces questions, nous identifions la liste de critères suivante:

[R1] Visualisation de motifs temporels sur toutes les séries. Ce critère se réfère à l'évolution des séries temporelles dans le temps afin de révéler des motifs, des tendances, des pics, etc. Dans l'exemple précédent, l'utilisateur peut voir la distribution des sentiments à un niveau élevé d'abstraction, afin d'avoir une vue générale des grandes tendances et de pouvoir visualiser l'émergence de motifs temporels particuliers.

[R2] Sélection d'un segment de temps intéressant. Ce critère consiste à permettre la sélection d'un segment de temps pour une analyse plus approfondie. Dans l'exemple précédent, l'utilisateur peut se concentrer sur le mois de mai (début des soldes d'été) pour comparer finement les sentiments exprimés (*e.g.* happy, elated, sad, etc.).

[R3] Traitement de segments de temps sélectionnés à différents niveaux d'abstraction. Ce critère consiste à afficher la structure hiérarchique des séries temporelles à différents niveaux de détails. Dans l'exemple précédent, l'utilisateur sélectionne le mois de mai pour représenter la distribution des sentiments avec un grand niveau de détails (*e.g.* happy, elated, sad, etc.). Afin de maintenir la carte mentale et mieux appréhender le contexte général, les périodes précédentes et suivantes sont affichées à un niveau plus élevé d'abstraction (*e.g.* pleasant-activation et unpleasant-deactivation).

La liste des critères ci-dessus tente de répondre aux tâches communes et spécialisées pour l'analyse visuelle de séries temporelles. Étant donné que ce type de données est présent dans de nombreux domaines, les utilisations possibles sont très vastes.

2.4 Encodages visuels et méthodes d'interaction

Cette section décrit nos techniques de visualisation et d'interaction. Notre processus d'exploration suit le *mantra* de recherche visuelle d'information proposé par Shneiderman: « tout d'abord, vue d'ensemble, puis zoom et filtre, et enfin, détails à la demande » [65]. Dans un premier temps, une vue d'ensemble permet à l'utilisateur de se faire une idée générale des données. Ensuite, en zoomant et en filtrant, il identifie l'information intéressante. Enfin, il peut accéder au détail des données.

2.4.1 Conception visuelle

Basé sur la liste de critères évoqués ci-dessus, nous proposons MULTISTREAM, une nouvelle approche pour faciliter l'analyse visuelle des séries temporelles hiérarchiques. Elle est basée sur quatre vues interactives :

— Une *vue globale* montre les séries temporelles à un haut niveau d'abstraction pour faciliter la reconnaissance de motifs (*e.g.* tendances, pics, etc.) et l'évolution globale de ces motifs au fil du temps *[R1]* (Cf. Figure 1.2a).
— Une *vue multirésolution* représente les séries temporelles à différents niveaux de détails *[R3]* (Cf. Figure 1.2b).
— Un *contrôleur* permet de sélectionner un segment de temps intéressant (sur la vue globale) et de mettre à jour la vue multirésolution *[R2, R3]* (Cf. Figure 1.2c).
— Un *gestionnaire de hiérarchie* permet d'explorer et de naviguer à travers les différents niveaux de la hiérarchie *[R3]* (Cf. Figure 1.2d).

Ces quatre vues sont décrites en détail dans les sections suivants.

2.4.2 Vue globale

Comme nous l'avons décrit dans l'état de l'art de ce chapitre, une visualisation de type stacked graph est bien adaptée pour représenter des séries temporelles multiples et révéler des motifs au fil du temps *[R1]*. Plus précisément, nous adoptons une approche streamgraph basée sur une métaphore de flux (Cf. Figure 2.20).

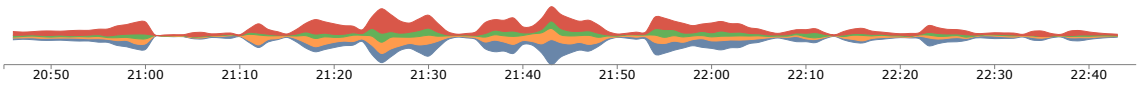


FIGURE 2.20 – La vue globale montre le niveau supérieur de la hiérarchie; elle permet à l'utilisateur d'observer les principaux motifs temporels.

Nous représentons les séries sous forme de couches colorées dans un système de coordonnées cartésiennes en deux dimensions, où la dimension temporelle est codée sur une échelle discrète le long de l'axe horizontal et la dimension quantitative est codée le long de l'axe vertical.

La Figure 2.20 montre un streamgraph de l'ensemble des séries à un haut niveau d'abstraction. Ce niveau représente le sommet de la hiérarchie et l'épaisseur d'une couche à ce niveau correspond à la somme des séries du groupe à chaque pas de temps. Ainsi, la vue globale montre les motifs temporels et permet de suivre leur évolution dans le temps *[R1]*.

Afin d'explorer et d'afficher les différents niveaux de la hiérarchie *[R3]*, nous proposons une vue multirésolution.

2.4.3 Vue multirésolution

La vue multirésolution est basée sur une approche *focus+contexte* pour montrer une partie des données en détail tout en conservant leur contexte. Cette vue représente les séries selon différents niveaux de granularité *[R3]*. Notre objectif est de montrer un haut niveau (Cf. Figure 2.5a) et un bas niveau (Cf. Figure 2.5b) de la hiérarchie dans une seule et même vue. Dans ce but, nous proposons une vue intégrant des zones de granularité différente.

La Figure 2.21 illustre le design de la vue multirésolution contenant l'exemple de l'évolution des genres musicaux décrit dans l'introduction de ce chapitre. Nous définissons les trois types de zones : zones de contexte, zone détaillée et zones de transition.

— Une *zone de contexte* correspond à un intervalle de temps où les séries sont affichées à un haut niveau d'abstraction (Cf. Figure 2.21a). Dans ce cas, les catégories de *metal* et de *jazz* sont tracées en utilisant deux teintes différentes.

— Une *zone détaillée* représente l'intervalle de temps pour lequel un bas niveau de la hiérarchie est affiché (Cf. Figure 2.21c). Dans ce cas, nous pouvons voir les sous-genres des catégories *metal* et *jazz* (*e.g. black metal*, *doom metal*, *jazz classic*, *jazz contemporain*, etc.). Dans cette zone, les séries relatives à une catégorie sont colorées avec la même teinte que la catégorie, mais avec un niveau de saturation différent permettant de les différencier.

— Une *zone de transition* correspond à un segment temporel dédié à la transition entre deux niveaux d'abstraction, c'est-à-dire entre une zone de contexte et une zone détaillée (Cf. Figure 2.21b). Dans une telle zone, une interpolation de couleur est appliquée pour adoucir la transition.

Les intervalles de temps dans la vue multirésolution ne sont pas uniformes afin de représenter différents niveaux de détails dans le temps (voir l'axe horizontal sur la Figure 2.21). Cette distorsion est liée à l'approche focus+contexte où les détails (focus) sont affichés sur une zone agrandie et le contexte sur une zone plus petite. Ainsi, la longueur des pas de temps dans la zone détaillée (Cf. Figure 2.21c) est plus grande que la longueur des pas de temps dans la zone de contexte (Cf. Figure 2.21a) et dans la zone de transition (Cf. Figure 2.21b).

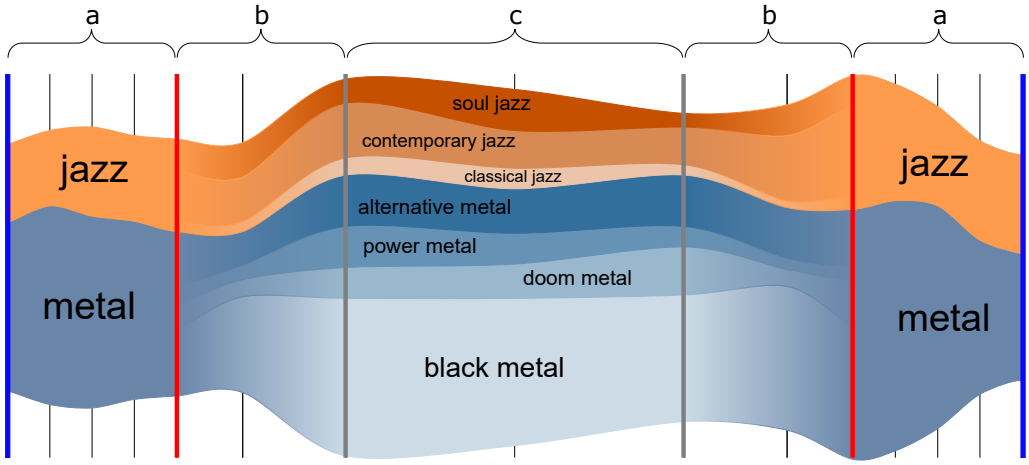


FIGURE 2.21 – La vue multirésolution. (a) Les zones de contexte représentent un haut niveau de la hiérarchie, (c) la zone détaillée représente un bas niveau de la hiérarchie et (b) les zones de transition représentent la transition entre les zones de contexte et la zone détaillée.

Comme notre visualisation utilise un système de coordonnées cartésiennes 2D, les longueurs des intervalles des zones de transition (Cf. Figure 2.21b) sont calculées à l'aide d'une fonction de distorsion cartésienne pour fournir un effet de lissage entre les zones de contexte et la zone détaillée (la fonction de distorsion est décrite dans la section 2.5).

La mise à l'échelle verticale dans la vue multirésolution (voir l'axe vertical sur la Figure 2.21) décrit la structure hiérarchique, où différentes couleurs sont utilisées pour discrétiser les séries dans l'organisation hiérarchique. De cette façon, en donnant à chaque couche une teinte distincte, nous faisons la distinction entre elles et diminuons l'ambiguïté.

Diverses approches utilisent l'ordre des couches dans une représentation de type streamgraph pour améliorer l'efficacité de la visualisation. Par exemple, dans NameVoyager [71] l'ordre des couches est alphabétique pour mettre en évidence la visualisation par nom. Dans notre approche, l'ordre des couches fournit un moyen de représenter les groupes de la hiérarchie des séries temporelles. Sur la Figure 2.21, les couches des sous-genres sont juxtaposées. Un point important à souligner est que l'axe vertical n'est pas déformé afin d'éviter de perdre le contexte de la distribution globale, ce qui rendrait la comparaison difficile.

L'information contextuelle peut s'avérer être une solution efficace pour améliorer l'expressivité d'une visualisation. Parfois, le problème de lisibilité lié au nombre de couches dans un streamgraph rend impossible d'étiqueter toutes les couches. Nous utilisons un algorithme de force brute [10] pour trouver le meilleur endroit pour placer l'étiquette et éviter le chevauchement. La taille de la police des étiquettes

encode la quantité de données représentées. Ainsi, plus la taille de la police est grande, plus la valeur associée est importante. L'étiquetage est affiché dans la zone détaillée pour indiquer en un coup d'œil quelle couche a la valeur la plus élevée dans cet intervalle de temps (Cf. Figure 2.21c).

La distribution des zones dans la vue multirésolution permet à l'utilisateur d'afficher les différents niveaux de granularité d'une hiérarchie. Par conséquent, l'utilisateur observe deux niveaux de la hiérarchie dans une vue unique. Afin de gérer les intervalles de temps des différents zones (*i.e.* détaillée, contexte et transition), nous avons conçu un *contrôleur* qui permet de les personnaliser interactivement, en fonction des besoins de l'utilisateur.

2.4.4 Le contrôleur

Comme les jeux de données des séries temporelles sont souvent volumineux, le filtrage des données non pertinentes aide les utilisateurs à se concentrer sur des périodes intéressantes (*e.g.* motifs, pics, etc.).

Basé sur la technique de *brushing&linking* [4], nous avons conçu un *contrôleur* (Cf. Figure 2.22) pour interagir avec la vue globale le long de l'axe horizontal (*i.e.* la dimension temporelle) et nous concentrer sur des périodes spécifiques *[R2]*. Cet outil est conçu sur la vue globale pour gérer les intervalles de temps utilisés par les différentes zones de la vue multirésolution (*i.e.* la zone détaillée, les zones de contexte et les zones de transition).

La Figure 2.22 montre le contrôleur. Des lignes interactives verticales contrôlent la position et l'extension des zones. Les zones de contexte sont gérées par les lignes bleues (Cf. Figure 2.22a). Les zones de transition sont traitées par les lignes rouges (Cf. Figure 2.22b). Enfin, les lignes grises définissent la zone détaillée (Cf. Figure 2.22c). Toutes les lignes du contrôleur peuvent être étendues et réduites de manière interactive à l'aide de la souris (les contraintes associées sont décrites dans la section 2.5).

Comme les lignes guident la position des zones, nous devons les gérer ensemble : la modification de l'une d'entre elles doit être répercutée sur les autres. Nous décrivons à présent ce processus.

Déplacement de la zone détaillée. Lorsque l'utilisateur fait glisser la zone détaillée à l'aide de la souris, toutes les autres zones sont mises à jour dynamiquement, en conservant leur longueur. La Figure 2.23a illustre la zone détaillée déplacée vers la gauche : toutes les autres zones sont mises à jour dans leur nouvelle position après avoir été déplacées vers la gauche. Sur la Figure 2.23b,

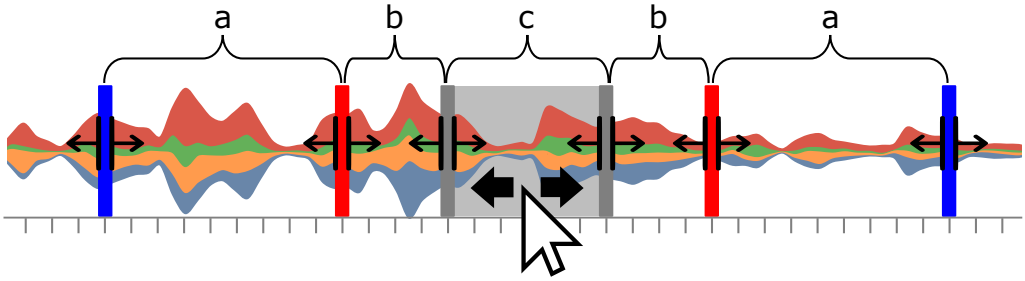


FIGURE 2.22 – Le contrôleur. les zones de contexte (a), les zones de transition (b) et la zone détaillée (c) sont gérées par des lignes colorées qui peuvent être étendues et réduites de manière interactive. La zone détaillée (c) reste colorée en gris pour rappeler à l'utilisateur le point de focalisation.

la zone détaillée est agrandie vers la droite en faisant glisser la ligne verticale grise droite. Cette action met à jour la position des zones adjacentes (zones de transition et de contexte) dans la même direction.

Déplacement d'une zone de transition ou de contexte. Lorsqu'une zone de transition ou une zone de contexte est agrandie ou réduite, seule la longueur de cette zone est mise à jour. La Figure 2.23c montre une zone de transition étendue vers la droite. Nous remarquons que toutes les autres zones restent statiques. La position des zones de contexte peut être verrouillée en cliquant sur l'icône du cadenas sous les lignes de contexte. La Figure 2.23d montre les deux zones de contexte verrouillées : la zone détaillée est déplacée vers la droite. Notons que lorsque la zone détaillée est déplacée, la longueur des zones de transition est mise à jour, mais pas la longueur des zones de contexte.

En utilisant le contrôleur, l'utilisateur peut voir les segments de temps à différents niveaux d'abstraction *[R2, R3]*. La Figure 2.24 montre la projection de zones sélectionnées par le contrôleur sur la vue multirésolution. Les modifications apportées dans la zone du contrôleur mettent automatiquement à jour la vue multirésolution. Cette animation aide l'utilisateur à conserver une carte mentale des vues (*i.e.* la vue globale et la vue multirésolution).

MULTISTREAM permet aussi de changer la ligne de base pour la disposition initiale des couches. La ligne de base par défaut est centrale parallèle à l'axe horizontal (*i.e.* un streamgraph) mais l'utilisateur peut facilement passer à une ligne de base au niveau de l'axe horizontal (*i.e.* un stacked graph) pour comparer les couches (Cf. Figure 2.34a).

Dans un streamgraph, il est souvent difficile de repérer les couches représentant des petites valeurs parmi les couches représentant des valeurs beaucoup plus grandes. Nous avons implémenté une *règle verticale* pour pallier ce problème et pour visualiser la valeur exacte codée pour une série à un pas de temps donné (Cf. Figure 2.25).

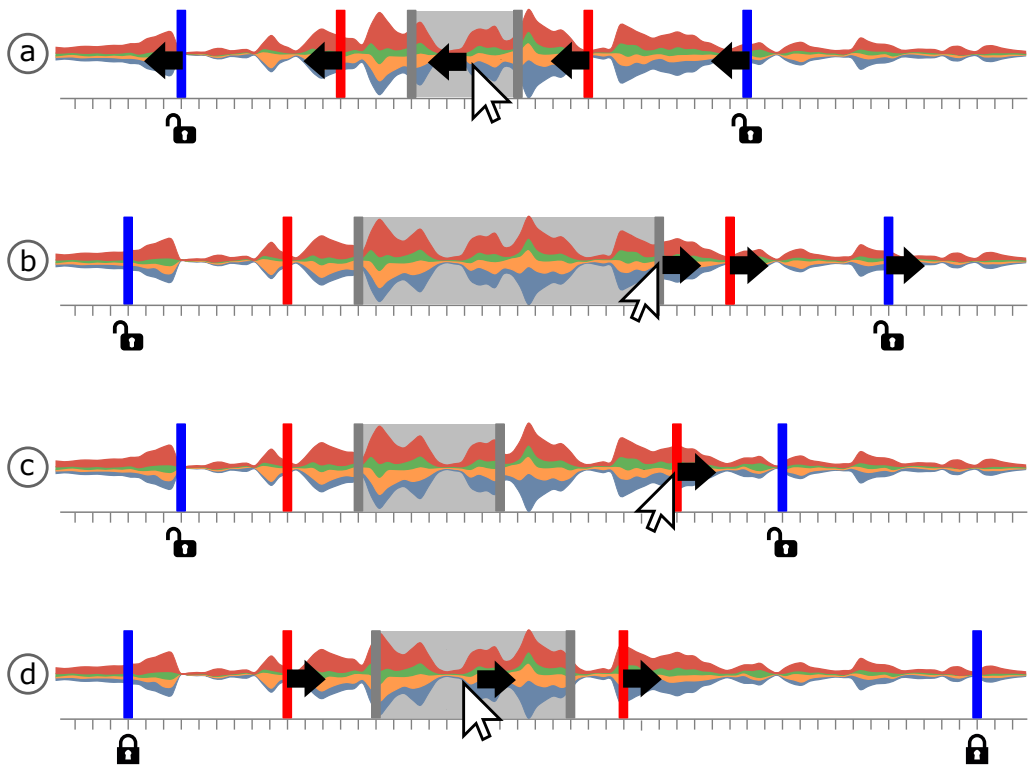


FIGURE 2.23 – Utilisation du contrôleur (a) La zone détaillée est déplacée vers la gauche. (b) La zone détaillée est agrandie vers la droite. (c) La zone de transition droite est agrandie vers la droite. (d) La position des zones de contexte est verrouillée en cliquant sur l'icône du cadenas.

Lorsque la souris survole une couche, la saturation change et la règle verticale apparaît, produisant une *infobulle* avec les informations temporelles et quantitatives associées. En cliquant sur cette infobulle, un panneau s'affiche, montrant des informations plus détaillées sur les données.

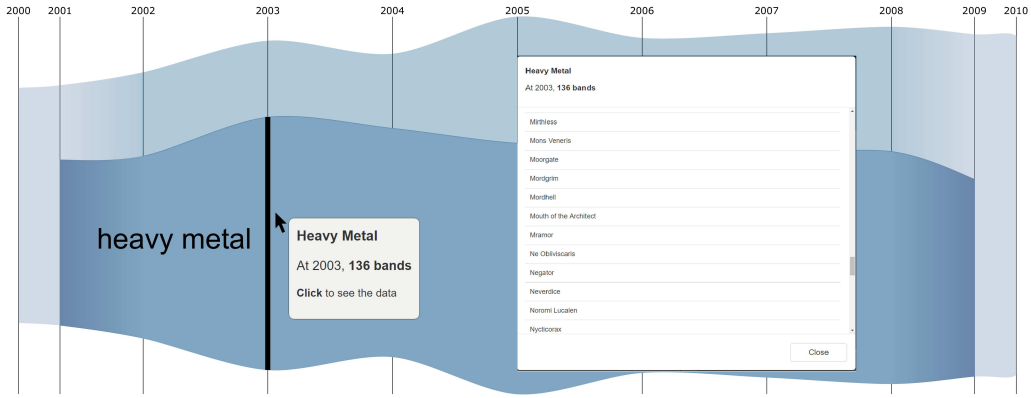


FIGURE 2.25 – Une infobulle affiche la valeur à chaque pas de temps dans le streamgraph.

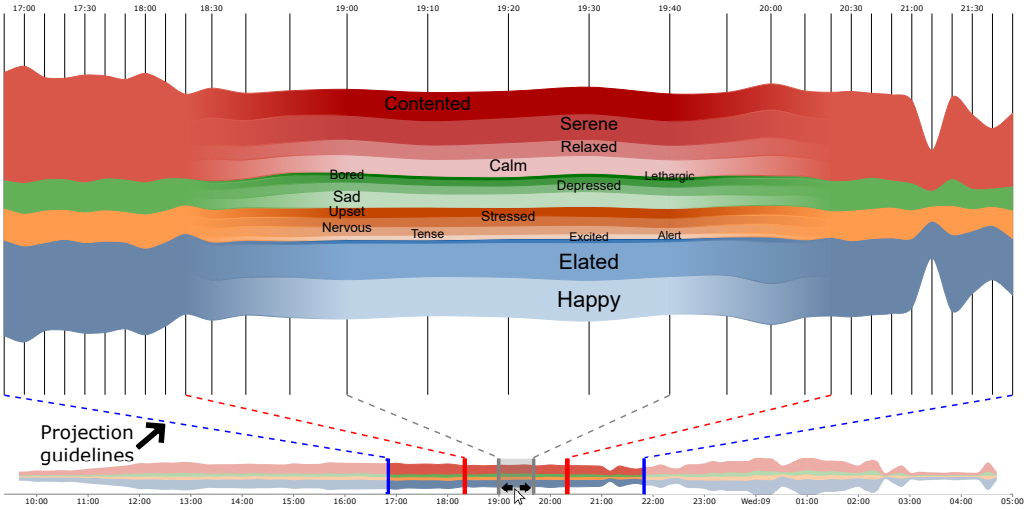


FIGURE 2.24 – En bas, le contrôleur montre les zones affichées dans la vue multirésolution. En haut, la vue multirésolution, résultat de la projection des zones définies par le contrôleur. Les lignes de projection (en pointillés) aident l'utilisateur à suivre ces projections.

2.4.5 Gestionnaire de hiérarchie

Deux niveaux de la structure hiérarchique sont affichés dans la vue multirésolution: les *zones de contexte* affichent un haut niveau d'abstraction, *i.e.* des flux provenant d'un niveau élevé de la hiérarchie, tandis que la *zone détaillée* affiche des détails, *i.e.* des flux provenant d'un bas niveau de la hiérarchie. Lorsque la profondeur de la hiérarchie est 2, les deux niveaux sont donc affichés. Cependant, la plupart des structures hiérarchiques contiennent plus de 2 niveaux. Pour traiter des hiérarchies plus profondes, et donc mieux adapter *[R3]*, nous proposons un *gestionnaire de hiérarchie*.

La Figure 2.26 montre la classification des genres musicaux représentée dans le gestionnaire de hiérarchie. Dans cette vue, nous adoptons une disposition sous forme d'arbre car une telle représentation permet de regrouper les séries temporelles et facilite l'exploration de la hiérarchie. Cet arbre est orienté horizontalement, le premier nœud à gauche représentant la racine de la hiérarchie. Chaque nœud feuille représente une série temporelle et les nœuds non-feuille expriment l'agrégation des séries. L'étiquette est placée à droite des nœuds feuilles et à gauche pour les autres nœuds.

Le gestionnaire de hiérarchie est coordonné avec la vue multirésolution (Cf. Figure 1.2(b,d)). Par conséquent, l'ordre des nœuds dans l'arborescence est le même que celui des couches dans la vue multirésolution. De plus, la couleur des nœuds utilise le même code de couleurs que les couches dans la vue multirésolution.

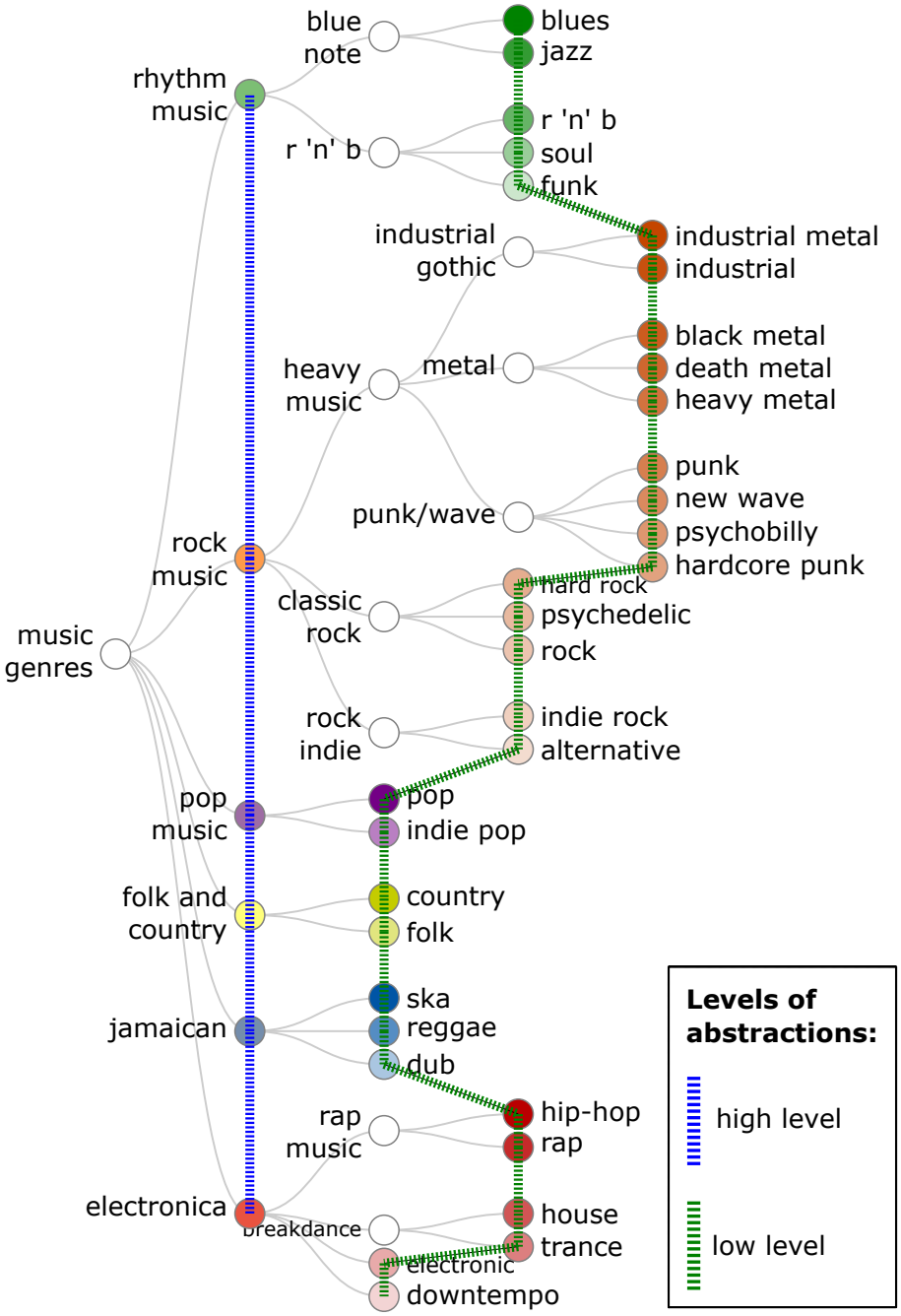


FIGURE 2.26 – Le gestionnaire de hiérarchie. Une hiérarchie de séries temporelles est représentée sous la forme d'un arbre, où les lignes verticales en pointillés indiquent les niveaux d'abstraction affichés sur la vue multirésolution. La ligne verticale en pointillés bleue indique les couches représentées dans les zones de contexte (*i.e.* haut niveau d'abstraction), la ligne verticale en pointillés verte indique les couches représentées dans la zone détaillée (*i.e.* bas niveau d'abstraction).

Le gestionnaire de hiérarchie permet de contrôler les niveaux de la hiérarchie affichés dans la vue multirésolution. Des lignes verticales en pointillés sont mises en œuvre pour représenter les niveaux utilisés dans les zones de contexte et la zone détaillée.

Sur la Figure 2.26, une ligne verticale en pointillés bleue traverse les nœuds qui sont représentés dans les zones de contexte (*i.e.* haut niveau d'abstraction). De manière similaire, une ligne verticale en pointillés verte traverse les nœuds qui sont affichés dans la zone détaillée (*i.e.* bas niveau d'abstraction). Suivant cette configuration, la ligne verticale verte traversera toujours des nœuds d'un niveau inférieur à ceux traversés par la ligne verticale bleue.

Afin de fournir une vue initiale optimale de l'organisation de la hiérarchie, notre approche montre initialement les nœuds de premier niveau de la hiérarchie dans les zones de contexte, et les nœuds des niveaux les plus bas de la hiérarchie dans la zone détaillée.

Nous décrivons, à présent, les interactions du gestionnaire de hiérarchie.

Agrégation et désagrégation dans la structure hiérarchique. Pour une analyse détaillée, le gestionnaire de hiérarchie permet aux utilisateurs de naviguer dans la hiérarchie (*e.g.* agrégation/désagrégation) en modifiant interactivement les lignes de coupe verte et bleue.

La désagrégation peut être effectuée sur les nœuds qui ont des enfants et l'agrégation peut être réalisée sur les nœuds qui ont un parent. Une *infobulle* est implémentée pour fournir ces deux fonctionnalités. Cette infobulle s'affiche lorsque l'utilisateur déplace la souris sur un nœud traversé par l'une des lignes verticales en pointillés. Un bouton flèche droite est affiché pour effectuer une désagrégation (*i.e.* descendre la ligne d'un niveau dans la hiérarchie) et un bouton flèche gauche pour effectuer une agrégation (*i.e.* remonter la ligne d'un niveau dans la hiérarchie).

La Figure 2.27a illustre la désagrégation du genre musical *heavy*. Ce genre est traversé par la ligne bleue en pointillés (*i.e.* la couche est tracée dans les zones de contexte). Après la désagrégation, la ligne bleue en pointillés traverse les trois enfants: *industrial gothic*, *metal* et *punk/wave*. Les zones de contexte de la vue multirésolution sont également mises à jour pour montrer ces trois genres.

La Figure 2.27b montre l'interaction d'agrégation. Dans cet exemple, les genres *black metal*, *death metal* et *heavy metal* sont regroupés dans le genre *metal*. Après l'agrégation, la ligne verte en pointillés traverse ce genre. La zone détaillée de la vue multirésolution est également mise à jour, montrant le genre métal.

Mise en évidence des séries. Le gestionnaire de hiérarchie fournit des interactions utiles qui améliorent les tâches de navigation. Les utilisateurs peuvent mettre en évidence un nœud en déplaçant la souris dessus.

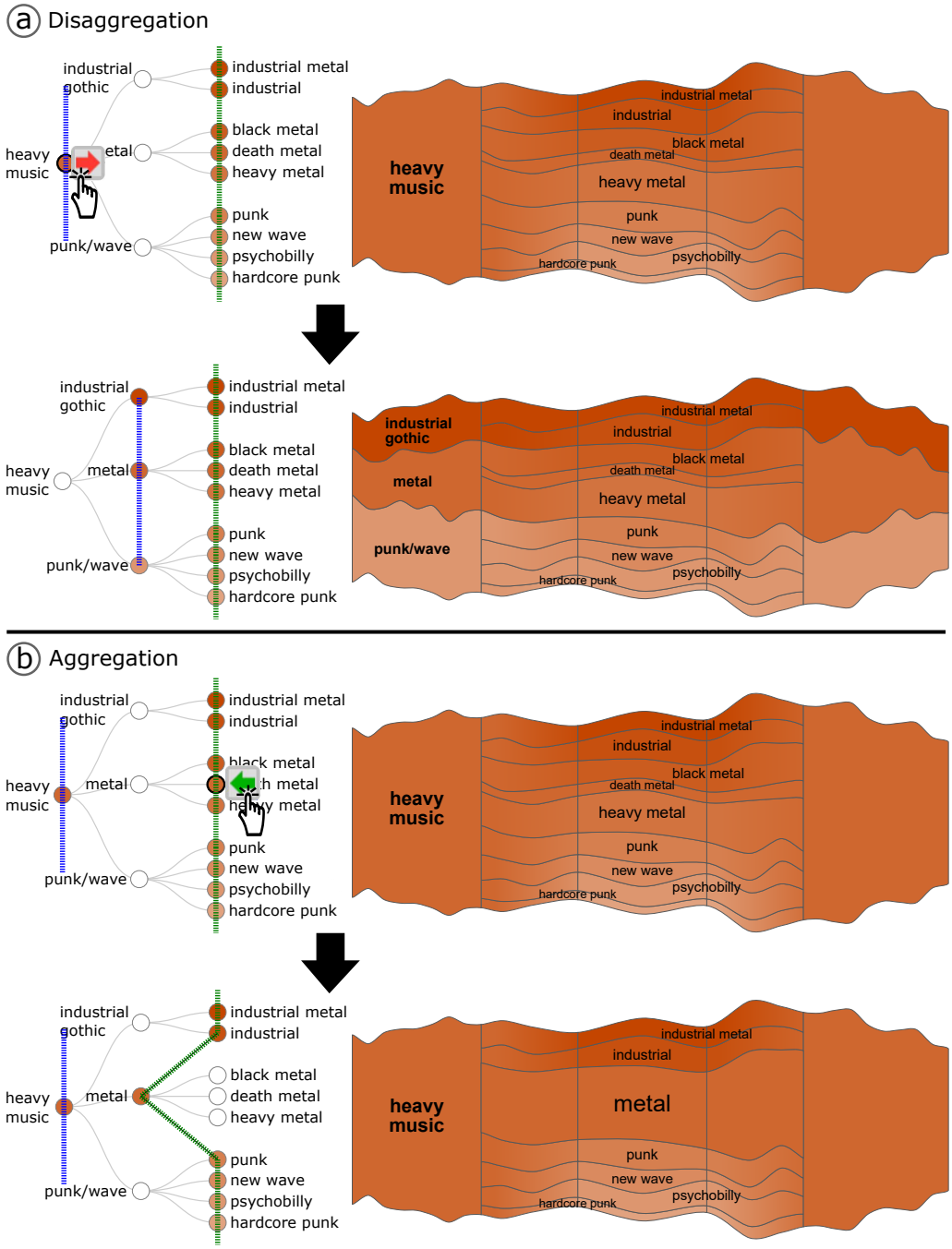


FIGURE 2.27 – Interactions permettant d'explorer différents niveaux d'abstraction dans la hiérarchie. (a) Désagrégation: diviser le nœud sélectionné en nœuds enfants. (b) Agrégation: regrouper les nœuds enfants dans leur nœud parent.

Si le nœud en surbrillance est traversé par la ligne verte (*i.e.* les couches sont tracées dans la zone détaillée) alors seule cette couche est mise en évidence dans la vue multirésolution, comme nous pouvons le voir sur la Figure 2.28a. Si le nœud

en surbrillance est traversé par la ligne bleue (*i.e.* que les couches sont tracées dans les zones de contexte) alors tous les descendants du nœud qui sont traversés par la ligne verte sont mis en évidence, comme nous pouvons le voir sur la Figure 2.28b. Ces animations sont destinées à aider les utilisateurs dans la tâche de navigation en mettant en évidence de manière coordonnée les relations hiérarchiques dans le gestionnaire de hiérarchie et la vue multirésolution.

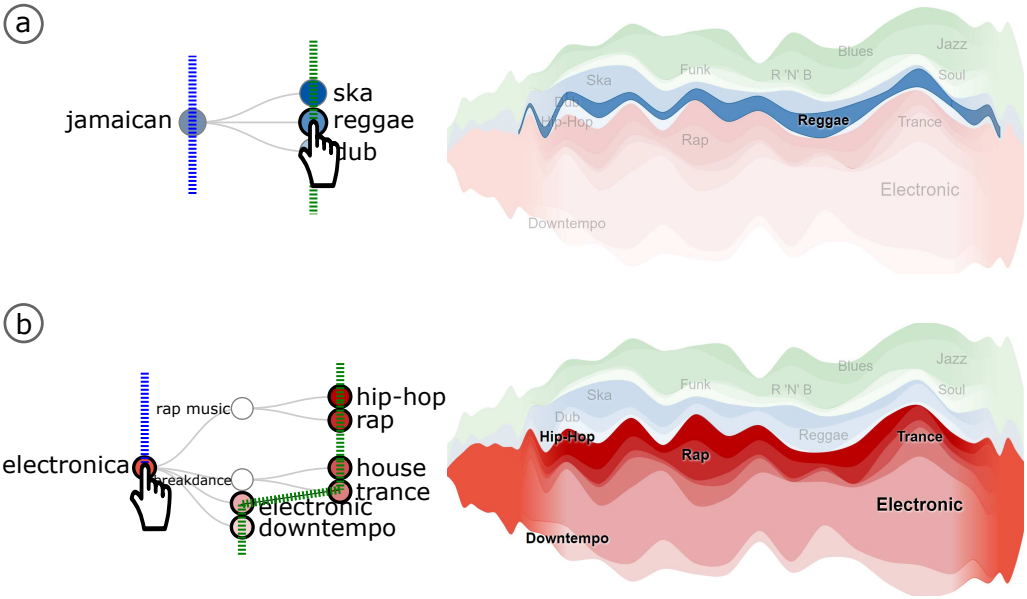


FIGURE 2.28 – Mise en évidence des séries. (a) Comme le nœud sélectionné *reggae* est représenté dans la zone détaillée (*i.e.* traversé par la ligne verte), seule la couche correspondante est mise en évidence. (b) Comme le nœud sélectionné *electronica* est représenté dans les zones de contexte (*i.e.* traversées par la ligne bleue), tous les descendants qui sont affichés dans la zone détaillée sont mis en surbrillance.

Filtrage des nœuds. Pour aider les utilisateurs à explorer leurs données, le gestionnaire de hiérarchie permet également de filtrer certaines séries temporelles. Cette fonctionnalité est effectuée en cliquant sur un nœud qui est traversé par l'une des lignes verticales en pointillés.

La Figure 2.29a montre le filtre du nœud de genre *electronica* qui est représenté dans la zone détaillée (*i.e.* traversé par la ligne verte). Après le filtrage des nœuds, les couches de la vue multirésolution sont mises à jour. La Figure 2.29b montre le filtrage d'un nœud qui est représenté dans les zones de contexte (*i.e.* traversées par la ligne bleue). Dans ce cas, comme ce nœud a des descendant représentés dans la zone détaillée (*i.e.* traversée par la ligne verte), tous les enfants sont également filtrés. Toutes ces interactions mettent à jour la vue multirésolution.

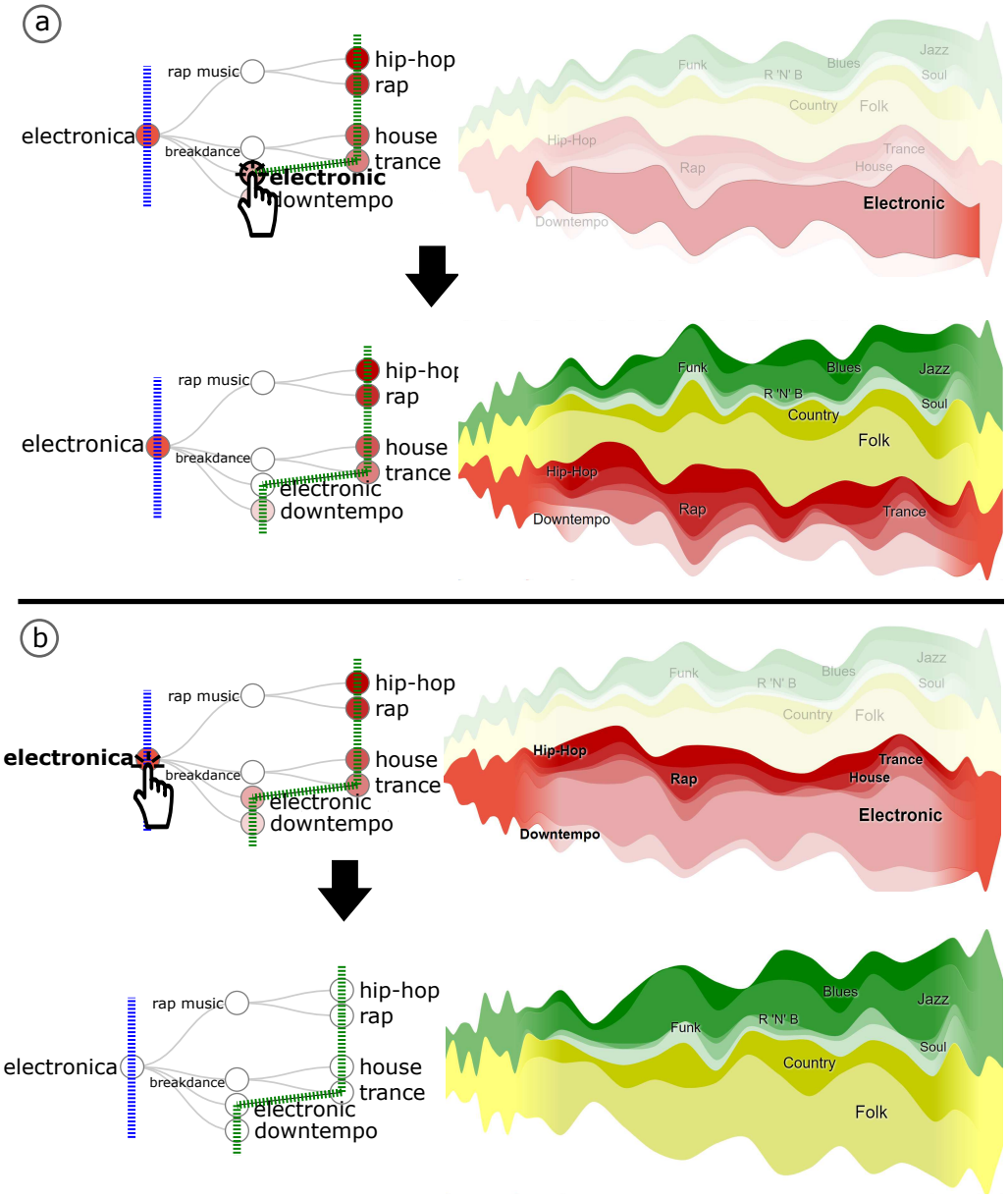


FIGURE 2.29 – Filtrage des nœuds. (a) Comme le nœud de genre *electronica* est représenté dans la zone détaillée (*i.e.* traversé par la ligne verte) alors seule cette couche est filtrée. (b) Comme la couche de genre *electronica* est représentée dans les zones de contexte (*i.e.* traversé par la ligne bleue), alors tous leurs descendants sont également filtrés.

En résumé, le gestionnaire de hiérarchie présente la structure hiérarchique sous la forme d'une arborescence et propose de nombreuses interactions pour améliorer la tâche d'exploration et de navigation des séries.

2.5 Contraintes techniques pour la vue multirésolution

Dans cette section, nous expliquons la projection du contrôleur sur la vue multirésolution ainsi que les contraintes que l'application doit gérer dans cette projection. Nous discutons également des transitions des couleurs.

2.5.1 Longueurs de pas de temps

Le contrôleur et la vue multirésolution sont basés sur une technique vue d'ensemble+détail. Les zones sélectionnées par le contrôleur sont projetées sur la vue multirésolution, créant un lien cognitif entre elles. La Figure 2.30 montre les variables impliquées dans cette conception où les pas de temps sont colorés en fonction de la zone qu'ils représentent: bleu pour les zones de contexte, rouge pour les zones de transition et gris pour la zone détaillée.

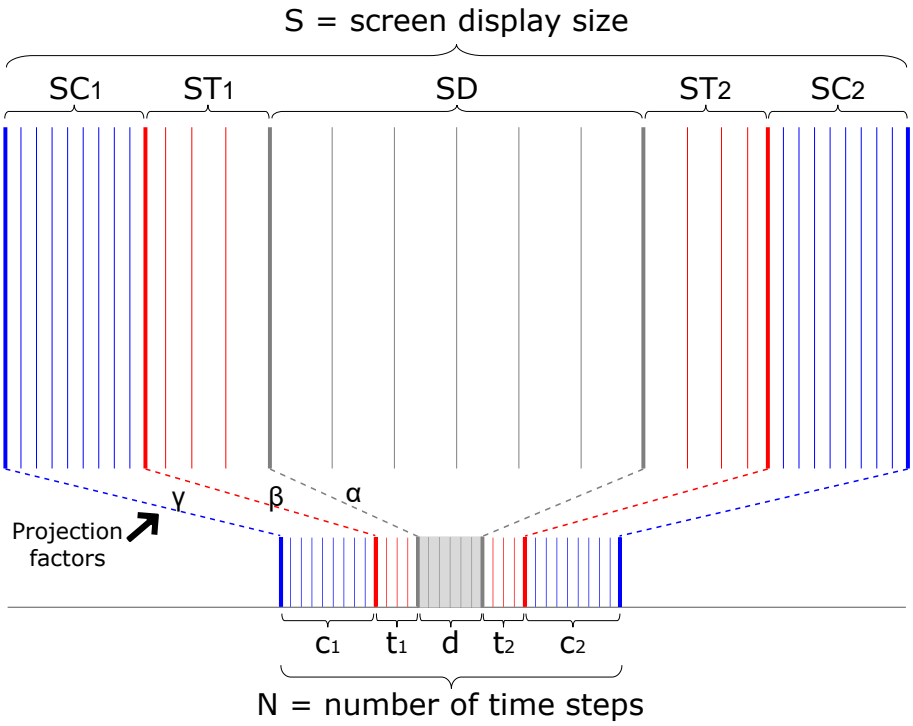


FIGURE 2.30 – Conception de la projection des zones du contrôleur (en bas) sur la vue multirésolution (en haut).

Dans le contrôleur (en bas de la Figure 2.30), le nombre de pas de temps est défini par: c_i pour les zones de contexte, t_i pour les zones de transition, et d pour la zone détaillée; où $i \in \{1, 2\}$. Puisque ces zones sont projetées sur la vue multirésolution, nous devons connaître le nombre total de pas de temps. Nous écrivons cette somme comme N où

$$N = c_1 + t_1 + d + t_2 + c_2.$$

L'étape suivante consiste à gérer la projection des zones du contrôleur sur la vue multirésolution (en haut de la Figure 2.30). L'espace horizontal disponible sur l'écran est représenté par S.

Afin d'obtenir des pas de temps dans la zone détaillée plus grands que les pas de temps dans les autres zones, nous utilisons des *facteurs de projection* (Cf. Figure 2.30). La zone détaillée est gérée par le facteur α, les zones de transition sont gérées par β, et les zones de contexte sont gérées par γ, où $\{\alpha, \beta, \gamma\} \in \mathbb{N}$. La contrainte $\alpha > \beta > \gamma$ est imposée sur ces facteurs pour préserver la relation entre les pas de temps de ces zones. Sans cette restriction, la longueur des pas de temps dans les zones de contexte pourrait être plus grande que la longueur des pas de temps dans la zone détaillée, perdant le sens de la technique de focus+contexte.

Afin de calculer la longueur horizontale relative pour chaque zone, nous divisons le nombre de pas de temps d'une zone par le nombre total de pas de temps (N) et le multiplions par le facteur de projection correspondant. Par conséquent, $RSC_i = \gamma \cdot \frac{c_i}{N}$ est la longueur relative pour les zones de contexte, $RST_i = \beta \cdot \frac{t_i}{N}$ est la longueur relative pour les zones de transition, et $RSD = \alpha \cdot \frac{d}{N}$ est la longueur relative pour la zone détaillée; $i \in \{1, 2\}$. La somme totale de ces longueurs relatives est représentée par RS, où

$$RS = RSC_1 + RST_1 + RSD + RST_2 + RSC_2.$$

La longueur horizontale de chaque zone dans la vue multirésolution est obtenue en utilisant la longueur horizontale relative et l'espace horizontal disponible (S). Ainsi, la longueur des zones de contexte est $SC_i = S \cdot \frac{RSC_i}{RS}$, la longueur des zones de transition est $ST_i = S \cdot \frac{RST_i}{RS}$, et la longueur de la zone détaillée est $SD = S \cdot \frac{RSD}{RS}$.

La longueur des pas de temps est obtenue en utilisant ces longueurs horizontales. La Figure 2.31 montre la distribution des pas de temps par zone de la vue multirésolution. Chaque pas de temps est représenté par une région sur l'axe horizontal. Cependant, la longueur d'un pas de temps dépend de la longueur de la zone et du nombre de pas de temps à tracer dans cette zone. Ainsi, les zones de contexte (SC_i) et la zone détaillée (SD) ont des longueurs de pas de temps uniformes. Les longueurs de pas de temps dans une zone de contexte sont $IC_i = \frac{SC_i}{c_i}$ et $ID = \frac{SD}{d}$ dans la zone

détaillée. Les pas de temps dans les zones de transition (ST_i) sont déformés car ils représentent la transition entre deux niveaux de granularité différents (*e.g.* zone de contexte à zone détaillée). Par conséquent, ils ne sont pas uniformes et une fonction exponentielle est utilisée pour décrire ce comportement.

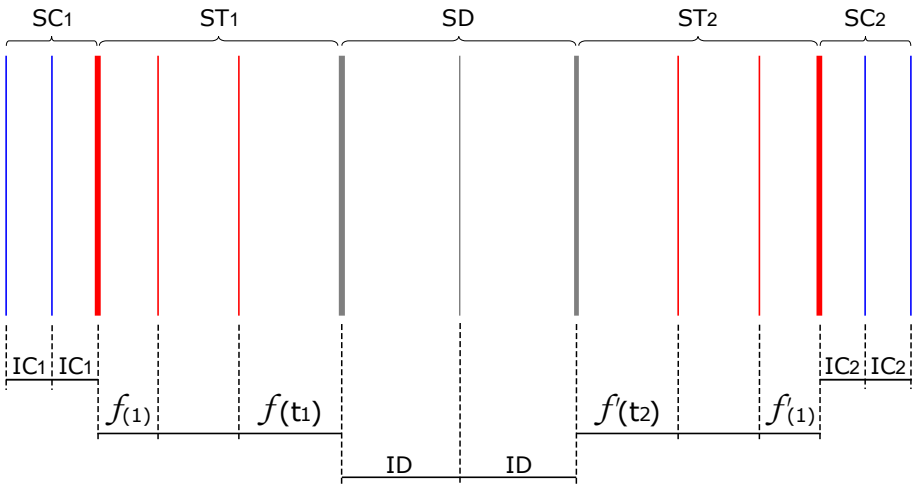


FIGURE 2.31 – La répartition horizontale des pas de temps dans la vue multirésolution.

Soit $T = \{1, \ldots, t_i\}$ l'ensemble des pas de temps dans une zone de transition. La longueur de chaque pas de temps $x \in T$ dans cette zone est donnée par la fonction :

$$f(x) = a \cdot b^x \tag{2.1}$$

où a est une constante représentant la valeur minimale possible de la fonction, et b la base de la fonction. La *base* doit toujours être supérieure à 1 pour assurer une fonction strictement croissante (*i.e.* $f(x+1) > f(x)$). Afin que les pas de temps de la zone de distorsion soient continus, cette fonction doit satisfaire deux exigences :

[E1] $f(1)$ doit être supérieure à IC_i.
[E2] $f(t_i)$ doit être inférieure à ID.

En supposant que $a = IC_i$ et $b > 1$ pour accomplir *[E1]*, nous devons trouver la base (b) satisfaisant *[E2]* pour atteindre la croissance exponentielle désirée.

Nous utilisons une fonction polynomiale réelle de degré t_i

$$ST_i = a \cdot b^1 + a \cdot b^2 + \ldots + a \cdot b^{t_i} \tag{2.2}$$

où ST_i est la longueur de la zone de transition.

Dans notre visualisation, alors que l'on fait glisser les zones dans le contrôleur (en bas de la Figure 2.30), la base (b) de l'Équation 2.2 est calculée pour chaque zone de transition. Nous utilisons l'algorithme RPOLY [41], puis nous appliquons les règles suivantes pour mettre à jour la vue multirésolution:

Si $b <= 1$ (*[E1]* non respecté) $\Rightarrow$ nous affichons un message d'alerte et nous ne mettons pas à jour la vue multirésolution.

Sinon si $f(t_i) >= ID$ (*[E2]* non respecté) $\Rightarrow$ nous affichons un message d'alerte et nous ne mettons pas à jour la vue multirésolution.

Sinon nous mettons à jour les zones dans la vue multirésolution avec les valeurs f.

Cette interaction donne à l'utilisateur l'autonomie nécessaire pour agrandir et réduire le contrôleur tout en respectant *[E1]* et *[E2]*. La Figure 2.32 montre trois configurations de la vue multirésolution qui montrent l'importance de respecter les exigences.

L'image du haut de la Figure 2.32 illustre les zones de transition (b) respectant *[E1]* et *[E2]*. Par conséquent, nous obtenons une vue où la relation entre les pas de temps de toutes les zones est correcte.

Dans l'image du milieu de la Figure 2.32, les zones de transition (b) ne respectent pas *[E2]*. Par conséquent, certains pas de temps dans les zones de transition (b) sont plus grands que les pas de temps dans la zone détaillée (c). Dans cet exemple, la zone « détaillée » n'est plus la zone montrant le plus de détails.

L'image en bas de la Figure 2.32 montre l'importance d'utiliser les zones de transition dans notre approche. Remarquez comment l'absence de zone de transition à gauche de la zone détaillée (c) entraîne un changement soudain entre les pas de temps de la zone de contexte (a) et de la zone détaillée (c). Cette absence peut gêner la perception des changements entre les différents niveaux de granularité. D'autre part, la droite de la zone détaillée (c) montre une zone de transition (b) où la longueur des pas de temps diminue à partir de ces deux zones, améliorant ainsi la perception des changements entre les niveaux de granularité.

Afin de contrôler les facteurs de projection (*i.e.* α, β et γ), un menu est implémenté au bas de l'interface de l'utilisateur. Ce menu affiche un message d'alerte si $\alpha < \beta$ ou $\beta < \gamma$.

2.5.2 Les couleurs

Les zones de transition représentent l'intervalle de temps entre deux niveaux d'abstraction (zone de contexte et zone détaillée). Ainsi, pour une transition en douceur entre ces deux niveaux, une interpolation de couleur est nécessaire.

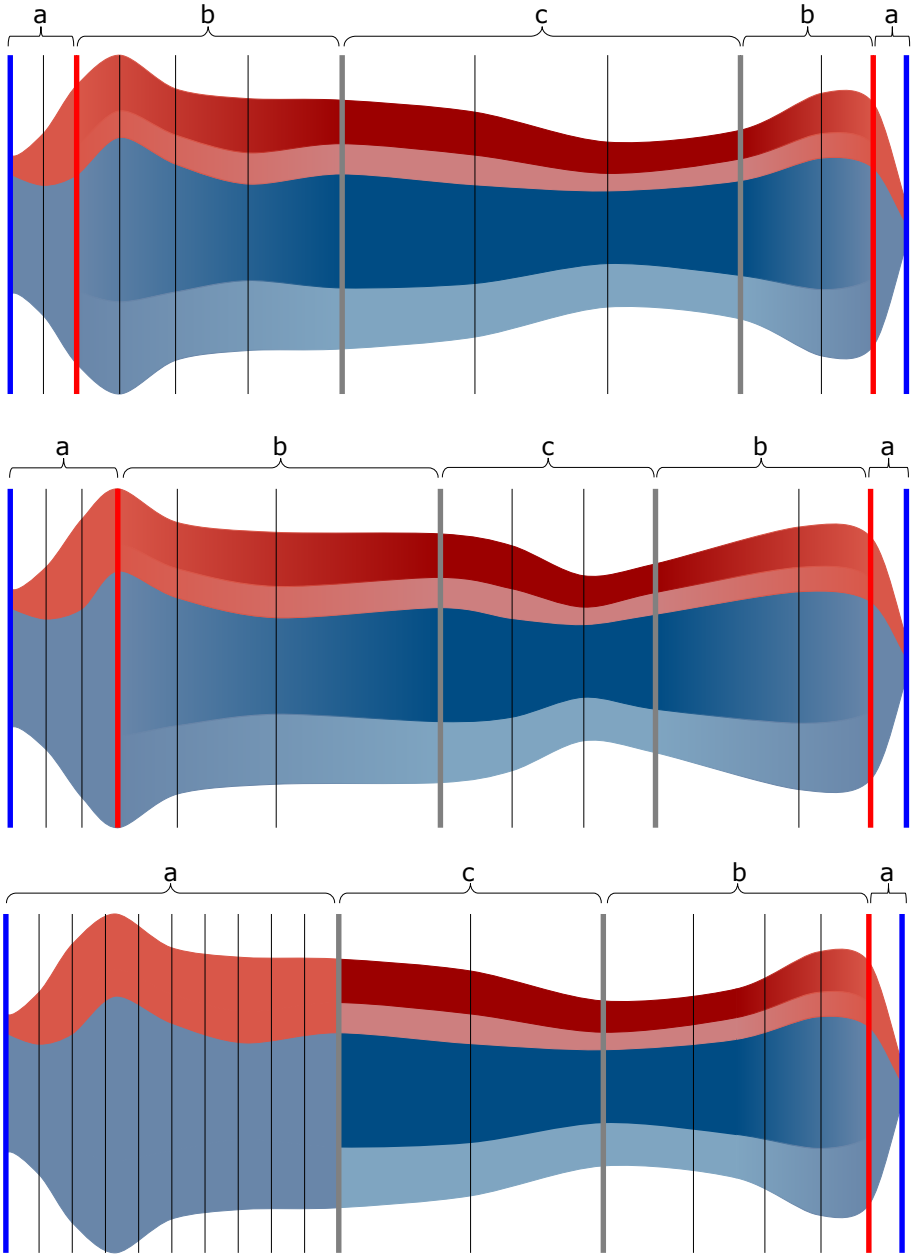


FIGURE 2.32 – Dans la première image, une vue multirésolution représente correctement la transition (b) entre une zone de contexte (a) et une zone détaillée (c). Dans l'image du milieu, les zones de transition (b) ne respectent pas l'exigence *[E2]*. La troisième image montre l'importance de l'utilisation de la zone de transition (b) où la zone de transition à gauche de la zone détaillée (c) n'est pas affichée.

La Figure 2.33 montre la vue multirésolution. La zone de transition gauche (voir b dans la partie gauche de la Figure 2.33) représente le passage entre la zone de contexte (a) et la zone détaillée (c) en augmentant la longueur des intervalles de

temps. Cependant, les changements soudains de couleur entre ces deux zones rendent la perception de la continuité difficile. Pour résoudre ce problème, nous utilisons une interpolation de couleurs le long de la zone de transition.

La zone de transition droite (voir b sur la droite de la Figure 2.33) montre une interpolation de couleurs entre la zone détaillée (c) et la zone de contexte (a).

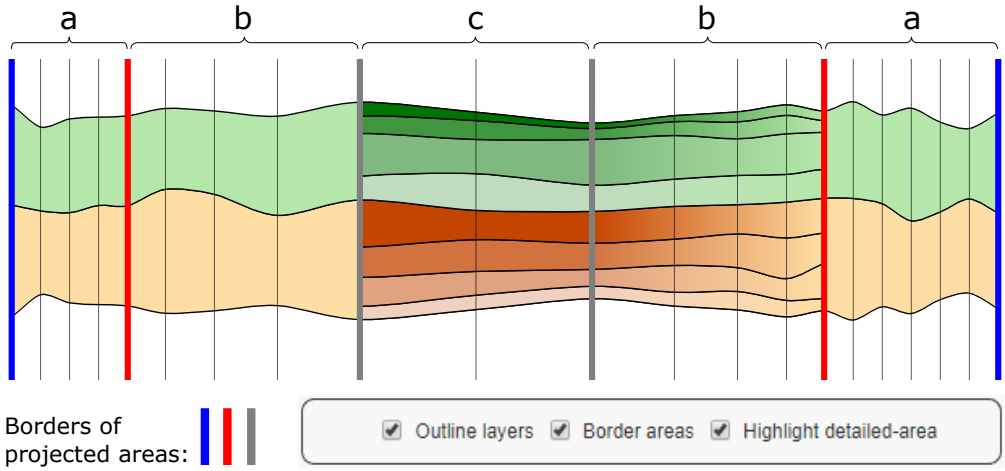


FIGURE 2.33 – L'importance d'une interpolation des couleurs dans les zones de transition (b) dans la vue multirésolution.

Le choix des couleurs est un problème lorsqu'il s'agit de hiérarchies profondes et de séries temporelles multiples. Pour gérer ce problème, notre approche adopte l'ensemble de 20 couleurs disponibles dans la bibliothèque D3.js [2]. Chaque couche d'un haut niveau d'abstraction utilise une teinte spécifique (voir « a » sur la Figure 2.33), où les teintes entre les couches voisines doivent être différentes pour éviter les ambiguïtés. Les couches des descendants (voir « c » sur la Figure 2.33) conservent la teinte de leur parent tout en faisant varier la saturation.

La Figure 2.33 illustre également trois caractéristiques supplémentaires qui visent à améliorer le problème de lisibilité des couleurs lorsque le nombre de couches augmente :

— **Lisibilité des bords des zones.** Pour améliorer la perception entre la vue globale, le contrôleur et la vue multirésolution, les bordures des zones projetées peuvent être affichées dans la vue multirésolution. Cette option est activée en cliquant sur la case à cocher « border areas ».

— **Lisibilité de la zone détaillée.** Lorsque le nombre de couches augmente dans la zone détaillée, il devient difficile de distinguer les couches avec des couleurs moins saturées. Pour surmonter ce problème, le contour des couches

2. https://d3js.org/

peut être affiché en noir, renforçant ainsi le lien entre la couche parentale et ses descendants. Cette fonctionnalité est activée en cliquant sur la case à cocher « outline layers ».

— **Lisibilité de la zone de transition.** Pour améliorer la perception de la transition entre les zones de contexte et détaillée, les couleurs des couches de la zone détaillée peuvent être mises en évidence en augmentant leur saturation par rapport aux couleurs des couches dans les zones de contexte. Cette option est activée en cliquant sur la case à cocher « highlight detailed-area ».

2.6 Discussion

Dans cette section, nous discutons de la flexibilité de MULTISTREAM pour atteindre différentes techniques d'interaction. Ensuite, nous la comparons avec des approches alternatives qui supportent une organisation hiérarchique des séries.

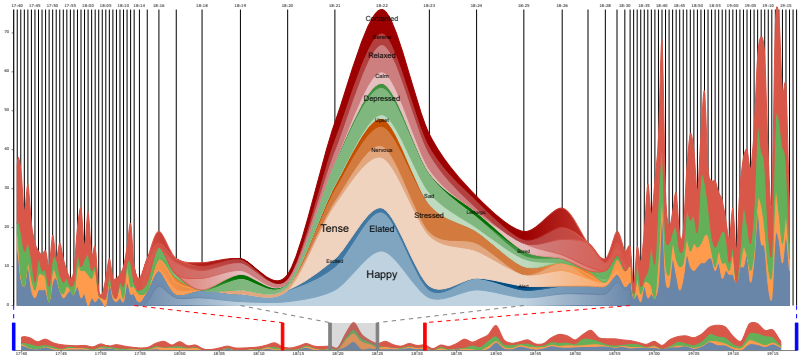
2.6.1 Une approche flexible

La flexibilité offerte par le contrôleur permet à l'utilisateur d'accomplir divers types d'interactions, telles que vue d'ensemble+détail ou fisheye. Ces techniques surmontent les problèmes de lisibilité et de comparaison des couches dans un streamgraph.

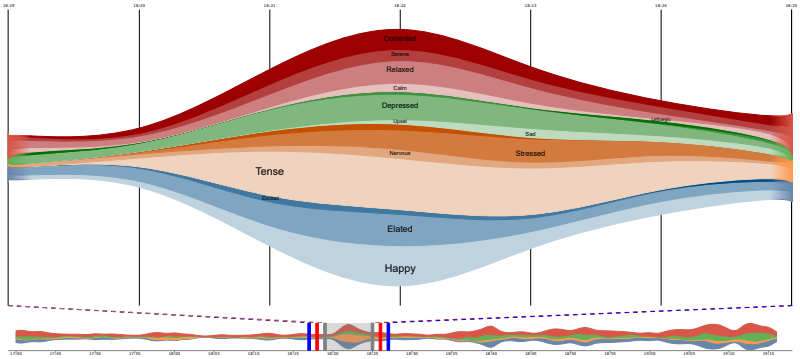
La Figure 2.34b illustre une vue d'ensemble+détail qui est réalisée en agrandissant la zone détaillée (en gris) et en réduisant les autres zones (*i.e.* les zones de contexte et de transition). Par conséquent, la zone sélectionnée dans la vue globale est projetée en détail sur la vue multirésolution *[R2]*.

Le contrôleur permet à l'utilisateur de gérer la longueur de la zone détaillée, qui peut être un intervalle de temps ou l'intégralité du jeu de données *[R1]*. La Figure 2.34c montre une distorsion du fisheye, où les positions des zones de contexte (lignes bleues) sont verrouillées sur les extrémités de l'axe de temps. Ainsi, les zones de contexte sont compressées et la zone détaillée (en gris) est agrandie, donnant l'effet d'un fisheye classique. Lorsque l'on fait glisser le contrôleur, la position des zones détaillées et de transition change, tandis que les limites des zones de contexte restent verrouillées *[R2, R3]*.

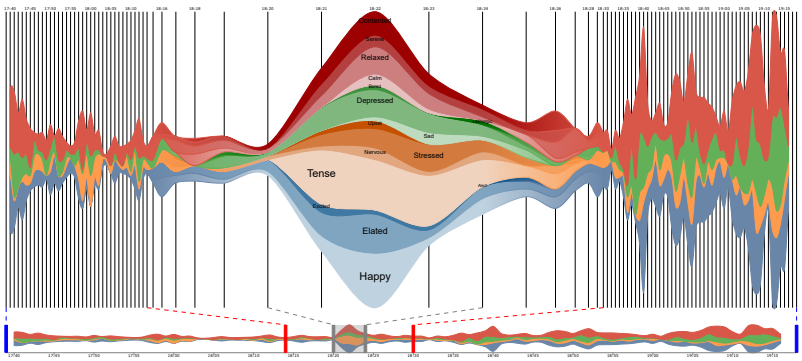
La Figure 2.34d montre une combinaison entre vue d'ensemble+détail et fisheye, où les intervalles dans la zone de contexte sont plus étendus que la zone détaillée. À l'aide du contrôleur, l'utilisateur peut personnaliser les intervalles de temps affichés de haut et bas niveau de granularité dans la vue multirésolution *[R3]*.



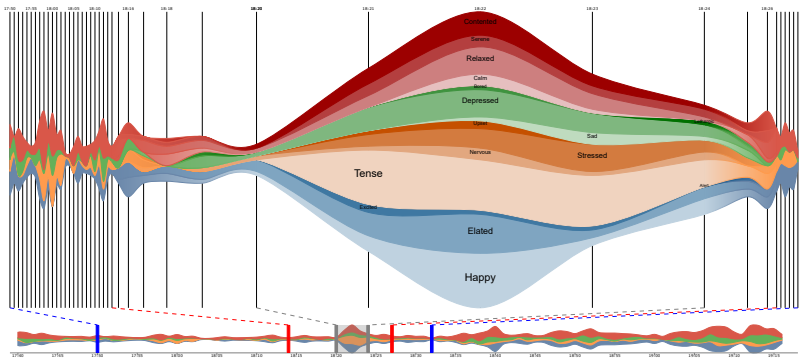
(a)



(b)



(c)



(d)

FIGURE 2.34 – Techniques d'interaction pour les séries temporelles hiérarchiques sur un streamgraph. (a) Modification de la ligne de base dans un streamgraph (stacked graphs ou streamgraph). (b) Vue d'ensemble+détail. (c) Distorsion fisheye. (d) Combinaison entre une vue d'ensembre+détail et une distorsion fisheye.

2.6.2 Comparaison avec des techniques alternatives

Nous comparons MULTISTREAM avec les approches basées sur des stacked graphs qui supportent une organisation hiérarchique, comme décrit dans l'état de l'art de ce chapitre (Cf. section 2.2). En particulier, nous nous focalisons sur trois aspects : la visualisation de longues séries temporelles, la navigation à travers une structure hiérarchique, et les niveaux représentés de cette hiérarchie.

Visualisation de longues séries temporelles. Un des défis dans la visualisation de larges jeux de données est l'affichage en raison de la limitation de la taille de l'écran. Certaines approches précédentes [3, 24, 29, 68, 72] souffrent de problèmes de lisibilité lorsqu'elles traitent de longues séries.

Par exemple, HierarchicalTopics [24] montre la hiérarchie d'un sujet dans différents panels, conduisant à un encombrement visuel lorsque le nombre de sous-sujets augmente. Dans d'autres approches, telles que BookVoyager [72], ManyEyes [68] et NewsLab [29], la longueur des séries est fixée à la taille de l'écran d'affichage (Cf. Figure 2.35). Par conséquent, dans de grands jeux de données, il est difficile de comparer des séries ou de trouver des motif récurrents au fil du temps *[R1]*. Pour résoudre ce problème, TouchWave [3] permet d'exécuter une technique focus+contexte (*e.g.* fisheye) pour afficher les détails dans une zone étendue et le contexte dans une zone plus petite. Cependant, TouchWave fixe également les extrémités du streamgraph aux bordures de l'écran.

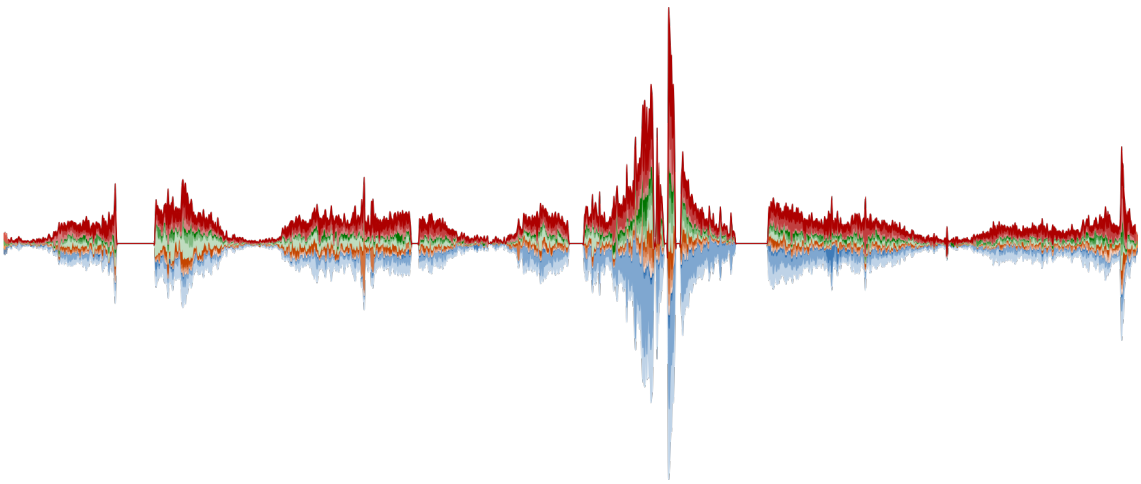


FIGURE 2.35 – Un streamgraph de longues séries temporelles limité par la taille de l'écran.

MULTISTREAM offre plus de flexibilité que TouchWave et les techniques précédentes, puisqu'en utilisant le *contrôleur*, les utilisateurs peuvent sélectionner les extrémités des zones de contexte et également limiter la zone détaillée qui est projetée dans la vue multirésolution *[R2, R3]*. Ce comportement permet de mettre en œuvre facilement différentes techniques d'interaction telles que zoom, focus+contexte et

distorsion de type fisheye. La flexibilité offerte par notre approche est utile lorsqu'un utilisateur se concentre sur un segment temporel précis, tout en conservant un contexte modulable de données historiques (Cf. Figure 2.34 (c,d)).

Navigation dans la structure hiérarchique. Pour naviguer dans la structure hiérarchique, les approches précédentes proposent différentes solutions. Par exemple, NewsLab [29] et TouchWave [3] utilisent des techniques d'interaction telles que *drill-down* et *roll-up*. Ces techniques permettent de naviguer à travers les couches à l'aide de la souris. Néanmoins, avec une telle navigation, il est difficile de comparer les relations entre couches sur le streamgraph, puisque les niveaux courants de la hiérarchie ne sont pas indiqués. Par conséquent, l'utilisateur peut se perdre pendant l'exploration de ses données.

Pour résoudre ce problème, BookVoyager [72], ManyEyes [68] et HierarchicalTopics [24] utilisent une représentation visuelle externe sous forme d'arbres pour naviguer dans la structure hiérarchique. Notre approche étend cette représentation arborescente au *gestionnaire de hiérarchie* en ajoutant un ensemble d'interactions *[R3]*.

MULTISTREAM améliore les inconvénients de TouchWave et NewsLab où seul le niveau courant analysé dans la hiérarchie est affiché. De plus, le gestionnaire de hiérarchie permet d'améliorer la navigation et l'exploration dans la hiérarchie en modifiant la granularité des séries (*e.g.* agrégation/désagrégation), en mettant en évidence des séries et en filtrant des séries *[R2, R3]*.

Niveaux représentés de la hiérarchie. Dans les approches précédentes telles que TouchWave [3] et NewsLab [29], l'utilisateur ne peut visualiser qu'un niveau d'abstraction de séries en même temps. Notre travail diffère de ces approches puisque la *vue multirésolution* permet de visualiser deux niveaux différents de la hiérarchie en même temps: contexte (*e.g.* un haut niveau de la hiérarchie) et focus (*e.g.* un bas niveau de la hiérarchie) *[R3]*.

En utilisant le *gestionnaire de hiérarchie* et le *contrôleur*, l'utilisateur peut explorer/naviguer dans la hiérarchie à une certaine période de temps pour une analyse détaillée *[R1, R2, R3]*. Ainsi, notre outil montre comment les parents et leurs descendants évoluent au fil du temps dans une même vue.

2.7 Exemples

Cette section présente deux exemples d'utilisation de MULTISTREAM. Le premier porte sur l'évolution des émotions exprimées dans des tweets le jour de l'élection présidentielle des États-Unis en 2016. Le second montre l'évolution des genres musicaux de 1960 à 2016.

2.7.1 L'élection présidentielle des États-Unis en 2016

Notre premier exemple est basé sur des tweets recueillis lors de l'élection présidentielle des États-Unis en 2016 (du 8 au 9 novembre 2016 UTC). Ce jeu de données contient les émotions des tweets extraits de 371 584 tweets contenant le hashtag #Hillary ou #Trump.

Les émotions sont classées selon le modèle affectif de Russell [25] (Cf. Figure 2.19) décrit dans [69]. Comme nous l'avons vu précédemment, ce modèle forme une structure hiérarchique des émotions. Dans cette organisation, le groupe *deactivation-pleasant* contient les émotions *contented*, *serene*, *relaxed* et *calm*. Le groupe *unpleasant-deactivation* contient les émotions *lethargic*, *bored*, *depressed* et *sad* . Le groupe *activation-unpleasant* contient les émotions *upset*, *stressed*, *nervous* et *tense*. Enfin, le groupe *pleasant-activation* contient les émotions *alert*, *excited*, *elated* et *happy*.

La Figure 2.37 illustre notre approche, où chaque couche dans le streamgraph représente une émotion. La couleur est utilisée pour distinguer les 4 principales catégories des émotions. Le groupe *pleasant-activation* est représenté en bleu, le groupe *activation-unpleasant* en orange, le groupe de *unpleasant-deactivation* en vert, et le groupe de *deactivation-pleasant* en rouge. Sur la Figure 2.37, la vue globale (en bas) représente l'évolution des émotions à un haut niveau d'abstraction, montrant ainsi une vue globale des principales émotions exprimées. La vue multirésolution (à droite) et le gestionnaire de hiérarchie (à gauche) sont liés pour représenter les émotions à différents niveaux de détails (catégories principales et désagrégées). Dans cet exemple, la hiérarchie contient 2 niveaux, dans lesquels les nœuds traversés par la ligne bleue en pointillés sont représentés dans les zones de contexte et les nœuds traversés par la ligne verte en pointillés sont affichés dans la zone détaillée.

La vue globale de la Figure 2.36 montre une répartition égale des émotions de 04h00 à 21h00. Cependant, en faisant glisser le contrôleur sur cette période nous voyons apparaître à 19h00 un pic concernant l'émotion *sad* (teinte verte). Pour nous concentrer sur l'émotion *sad*, nous pouvons utiliser le gestionnaire de hiérarchie pour filtrer des feuilles qui ne sont pas de ce groupe (unpleasant-deactivation), par exemple, dans notre cas, nous pouvons cacher celles du groupe *pleasant-activation* (teinte bleue) dont les valeurs sont très élevées et empêchent de voir correctement les autre émotions. Dans ce cas, des pics correspondant à l'émotion *sad* apparaissent plus clairement. Dans la vue multirésolution, lorsque nous cliquons sur la couche de l'émotion *sad*, nous observons que de nombreux tweets contiennent le texte « Trump just filed his first election lawsuit ». Nous voyons que ce pic exprime l'émotion de tristesse quand Donald Trump a porté plainte contre la ville de Clark dans l'état du Nevada.

La vue globale de la Figure 2.37 montre quelques pics intéressants à la fin du jour du scrutin. La vue multirésolution représente les détails de 23h00 à 06h00. Nous observons que jusqu'à 23h00, toutes les couches évoluent de façon homogène. Ensuite, les premiers résultats sont diffusés, ce qui fait augmenter toutes les couches, en particulier la couche *elated* (Cf. Figure 2.37a). Entre minuit et 02h00, quand la plupart des sondages ont été fermés, le volume de l'émotion *elated* reste constant. Nous pouvons remarquer facilement que dans cette période, la hauteur de l'étiquette de *elated* est plus grande que les autres. Le pic de la Figure 2.37b montre le moment où les résultats de l'état de Floride ont été diffusés. Un autre pic à 04:30 montre le moment où les résultats de la Californie ont été diffusés (Cf. Figure 2.37c).

La vue multirésolution permet à l'utilisateur de se concentrer sur une zone tout en conservant un contexte personnalisable. Par exemple, alors que nous nous concentrons sur les pics entre minuit et 05h00, nous pouvons voir que le plus haut sommet de l'ensemble de données est à 09h00, quand Donald Trump a été déclaré vainqueur de l'élection.

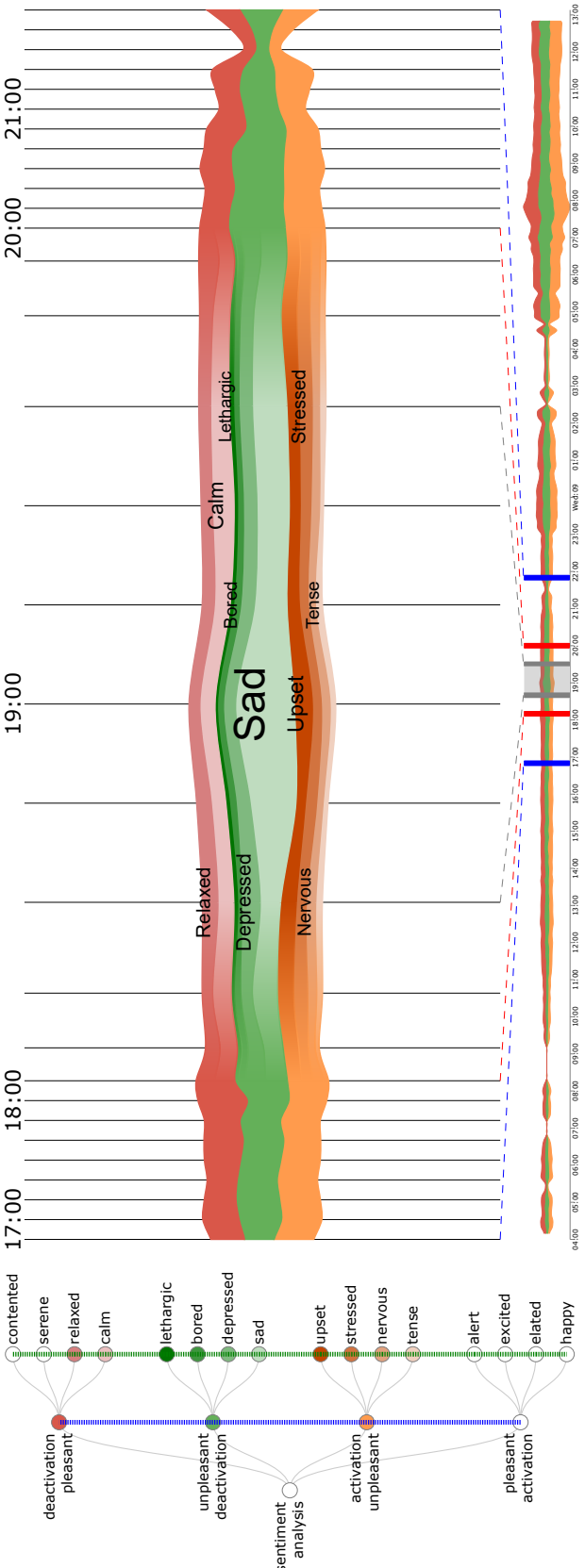


FIGURE 2.36 – Évolution des émotions exprimées dans les tweets le jour de l'élection présidentielle des États-Unis en 2016: pic de tristesse à 19h00.

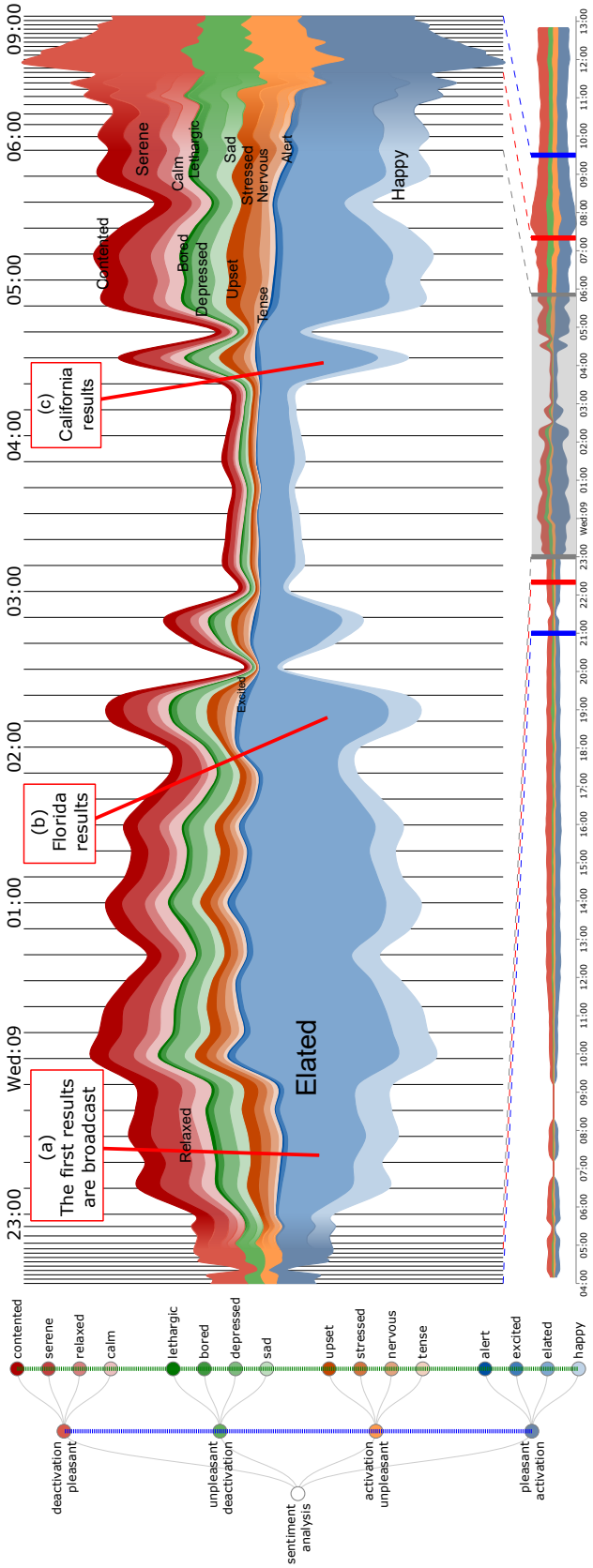


FIGURE 2.37 – Évolution des émotions exprimées dans les tweets le jour de l'élection présidentielle des États-Unis en 2016: les résultats.

2.7.2 L'évolution des genres musicaux

Cet exemple est basé sur des données musicales extraites du site web MusicBrainz [3]. Les informations portent sur les métadonnées d'environ 10 642 groupes de 1960 à 2016. Nous avons utilisé cette information pour concevoir notre propre taxonomie de genres musicaux. Cette taxonomie peut être organisée en une structure hiérarchique (Cf. Figure 2.26). Ainsi, le premier niveau est composé de 6 catégories principales (*e.g. rhythm music*, *rock music*, *pop music*, *folk and country*, *jamaican*, et *electronica*), et le niveau le plus bas est composé de 32 sous-catégories (*e.g. blues*, *jazz, r 'n' b*, *soul*, *funk*, *industrial metal*, *industrial*, *black metal*, *death metal*, *heavy metal*, *punk*, *new wave*, *psychobilly*, *hardcore punk*, *hard rock*, *psychedelic*, *rock*, *indie rock*, *alternative*, *pop*, *indie pop*, *country*, *folk*, *ska*, *reggae*, *dub*, *hip-hop*, *rap*, *house*, *trance*, *electronic*, et *downtempo*). Cette taxonomie vise à aider l'utilisateur à naviguer dans l'histoire de la musique avec différents niveaux de granularité (*i.e.* genre et sous-genres).

La Figure 2.38 illustre la visualisation proposée. Chaque couche dans le streamgraph représente un genre, et leur longueur horizontale représente leur longévité au fil du temps. En bas, la vue globale montre l'ensemble des données à un haut niveau d'abstraction. Le gestionnaire de hiérarchie (à gauche) et la vue multirésolution (à droite) sont coordonnés pour représenter les couches du genre à différents niveaux de détails. Différentes couleurs sont utilisées pour décrire les principaux genres. Ainsi, le genre *rhythm music* est représenté en vert, le genre *rock music* en orange, le *pop music* en violet, le genre *folk and country* en jaune, le genre *jamaican* en bleu et le genre *electronica* en rouge. Dans cet exemple, la profondeur de la hiérarchie varie en fonction du niveau de détail de chaque genre musical. L'épaisseur de la couche d'un genre représente le nombre de groupes créés par année. Par exemple sur la vue multirésolution de la Figure 2.38, la règle verticale montre que 167 nouveaux groupes de musique rock ont été créés en l'an 2000.

La vue globale de la Figure 2.38 montre un pic intéressant du *rock* (orange) entre le milieu des années 1970 et le milieu des années 1980. Afin d'explorer les détails de la composition des genres dans cette période, nous utilisons le contrôleur pour filtrer les autres périodes et le gestionnaire de hiérarchie pour naviguer à travers les différents niveaux de granularité. Ainsi, nous représentons la période des années 1970 aux années 1990 en détail (*i.e.* les genres traversés par la ligne verticale verte). Afin de ne pas perdre le contexte de cette période, nous avons pris en compte des périodes de temps avant et après.

3. MusicBrainz est une encyclopédie musicale ouverte qui collecte des métadonnées musicales. https://musicbrainz.org/ [dernier accès 01/07/2018]

Nous observons que le pic du *rock* est dû à un pic du sous-genre *heavy music* dans les années 70. Nous utilisons le gestionnaire de hiérarchie pour afficher plus de détails sur ce sous-genre. La Figure 2.39 montre les sous-genres de la musique rock. Nous observons que le pic de *heavy music* est dû à un pic du sous-genre *punk/wave* dans les années 70. Cependant, au début des années 80, le *punk/wave* perd de sa popularité, entraînant une baisse du *heavy music* et du *rock*. Après 1985, d'autres sous-genres tels que le *black metal*, le *death metal*, le *heavy metal* et *l'indie rock* gagnent du terrain, ce qui fait que le *rock* retrouve son niveau de popularité des années 70. Remarquez comme sur la Figure 2.39, le gestionnaire de hiérarchie montre la profondeur dans chaque genre et sous-genre (*i.e.* lignes verticales bleues et vertes). Dans cet exemple, il est particulièrement important d'avoir un large contexte autour de notre segment de focus pour distinguer clairement les tendances de tous les principaux genres. Ainsi, nous observons en détail une période spécifique ainsi que la croissance du genre rock (orange) au fil du temps.

La vue globale de la Figure 2.38 montre que le *rock* (orange) est le genre le plus dominant. Cela provoque des problèmes dans l'analyse des autres genres. En utilisant le gestionnaire de hiérarchie, nous pouvons filtré le genre *rock*, cela nous permet d'améliorer la comparaison entre les genres. La Figure 2.40 montre le streamgraph filtré. Il est intéressant de détecter le moment où le genre *rap music* est né (Cf. Figure 2.40a). Un pic du genre *rhythm music* (teinte verte) de 1965 à 1970 exprime la domination de ce genre (Cf. Figure 2.40b). Le genre *electronic* (teinte rouge) commence à grandir à partir de 1975.

La vue multirésolution de la Figure 2.40 montre la décennie 2000 en détail. Détecter les pics les plus élevés d'une période est facile, grâce à la position et à la hauteur des étiquettes. Aussi, l'étiquetage des couches donne un aperçu du genre qui domine dans cette période. Dans ce cas, le genre *electronica* atteint un sommet en 2007 avec 26 groupes formés.

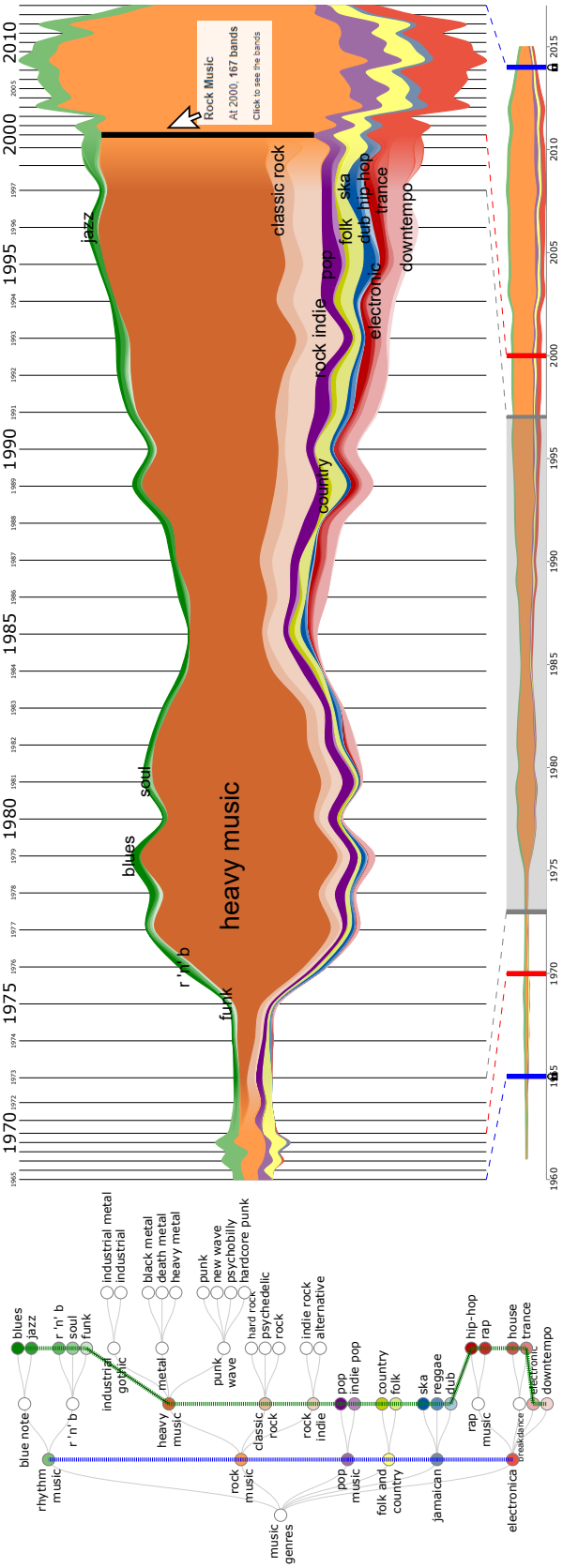


FIGURE 2.38 – Évolution des genres musicaux de 1960 à 2016: focus sur la période 1975-1995.

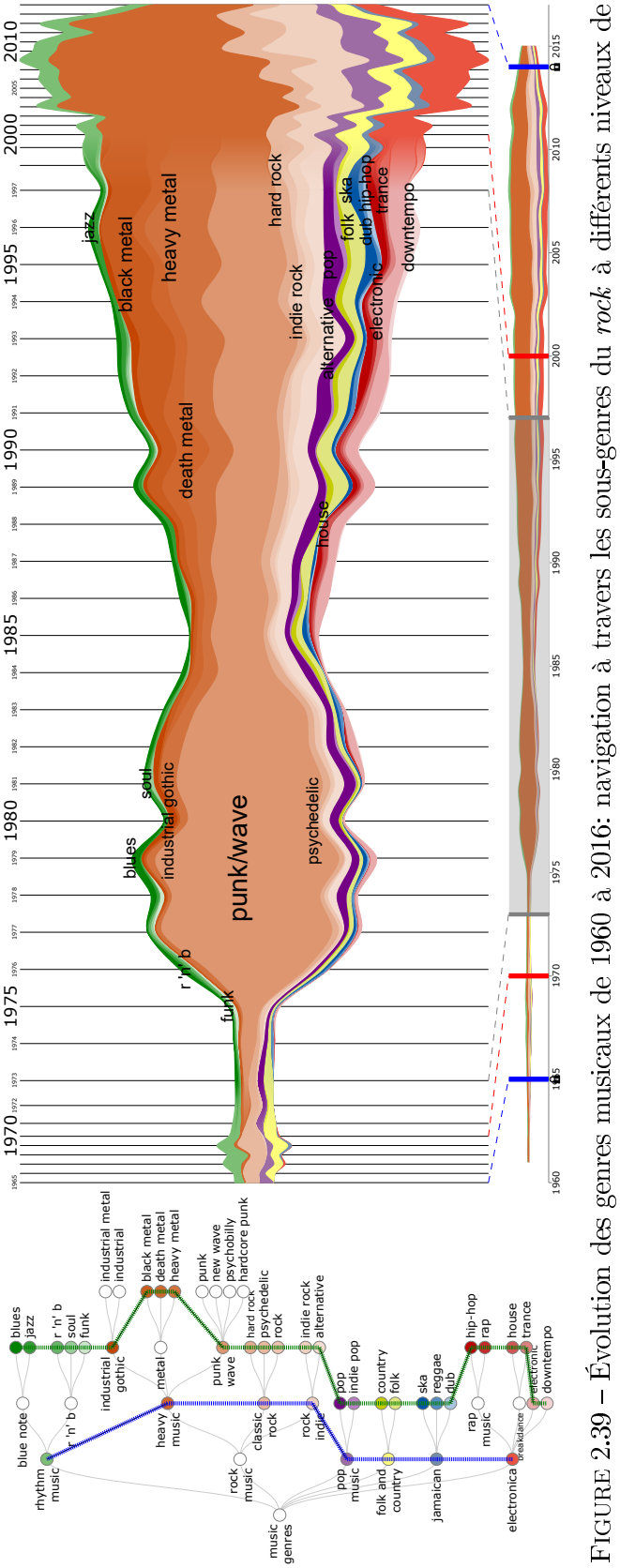


FIGURE 2.39 – Évolution des genres musicaux de 1960 à 2016: navigation à travers les sous-genres du *rock* à différents niveaux de détail.

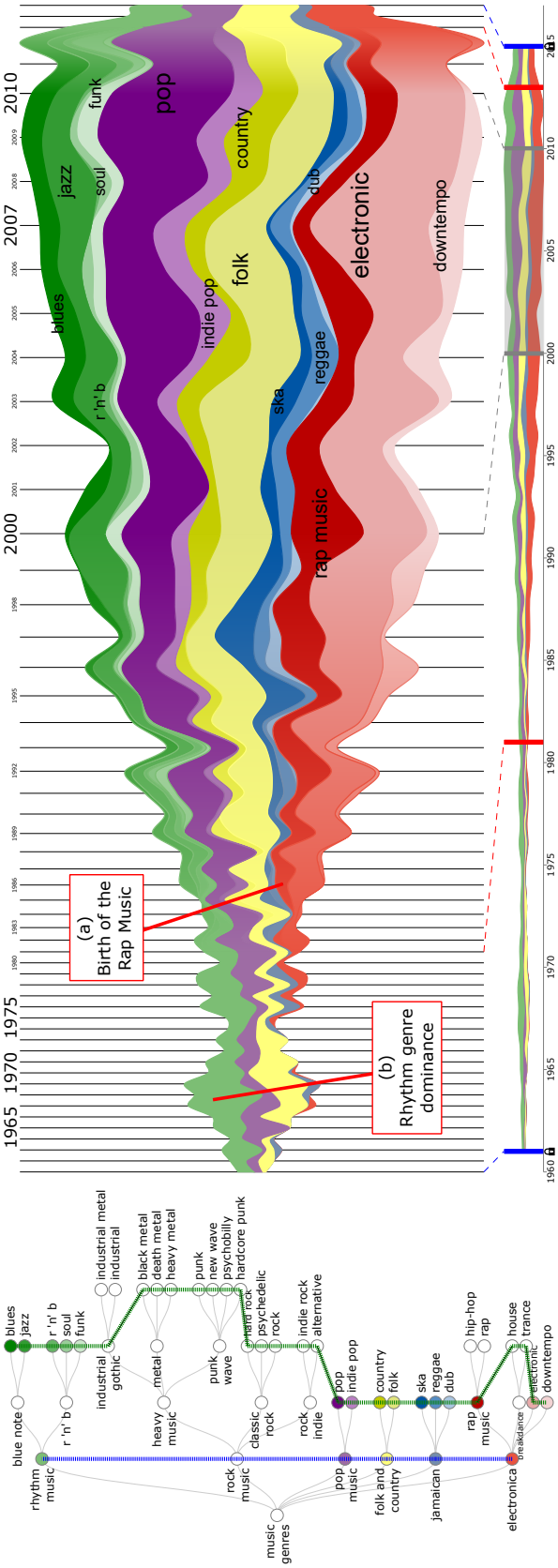


FIGURE 2.40 – Évolution des genres musicaux de 1960 à 2016: focus sur la période 2000-2010.

2.8 Conclusion

Dans ce chapitre, nous avons présenté MULTISTREAM, une nouvelle approche pour visualiser de longues séries temporelles multiples organisées dans une structure hiérarchique. Elle est basée sur quatre composants interactifs : une *vue globale* qui montre les principales tendances du jeu de données, une *vue multirésolution* qui montre des détails sur un sous-ensemble du jeu de données, un *contrôleur* qui filtre des périodes spécifiques et un *gestionnaire de hiérarchie* qui permet de naviguer/explorer la structure hiérarchique.

L'organisation hiérarchique des séries est affichée à différentes granularités sur la vue multirésolution. En suivant le mantra proposé par Shneiderman « aperçu d'abord, zoom et filtre, puis détails sur demande », les interactions mises en place couvrent un large éventail de possibilités, comme nous l'avons vu dans la section 2.6.

L'un des défis lors de la phase de conception de notre approche était l'interaction entre les différentes composants visuelles. Nous avons constaté qu'un ensemble d'interactions nous permettait d'atteindre nos objectifs. Ainsi, ces techniques aident à sélectionner un segment temporel et à naviguer dans la hiérarchie de ce segment. En conséquence, la vue multirésolution représente la hiérarchie à différents niveaux de détails dans une période de temps sélectionnée. De plus, toutes les interactions sont intégrées et peuvent être réalisées à n'importe quel moment de l'exploration sans perdre le contexte.

Deux exemples montrent que MULTISTREAM peut être appliqué dans de nombreux domaines tels que les données sociales ou les données de l'industrie musicale. Sa flexibilité permet de naviguer et d'explorer les hiérarchies en aidant les utilisateurs à transmettre les pics saisonniers et les tendances au fil du temps.

Chapitre

3 Visualisation de graphes multi-couches

Sommaire

3.1 Introduction 56
3.2 État de l'art 57
3.2.1 Systèmes visuels de requêtage de graphes 57
3.2.2 Suggestion visuelle de requêtes 60
3.2.3 Comparaison des approches 61
3.3 Critères 62
3.4 Encodages visuels et modes d'interaction 64
3.4.1 Conception visuelle 64
3.4.2 Vue de la requête 64
3.4.3 Vue du graphe 69
3.4.4 Vue des résultats 70
3.5 Interactions 72
3.5.1 Visualiser les résultats 74
3.5.2 Explorer les résultats 77
3.6 Une approche basée sur les diagrammes Kelp 79
3.6.1 Positionner les sommets et les résultats 79
3.6.2 Génération d'une visualisation imbriquée 80
3.6.3 Mise en œuvre et équipement 83
3.7 Exemple 83
3.8 Conclusion 88

3.1 Introduction

Alors que le chapitre précédent s'intéressait à la prise en compte de multiples séries temporelles avec une structure hiérarchique, nous abordons dans ce chapitre la problématique associée à la visualisation d'informations pour des données décrites sous la forme de graphes.

Les **graphes** sont des structures qui peuvent être utilisées pour modéliser des relations (les **arêtes**) entre des entités (les **sommets**). Lorsque les sommets sont reliés par des arêtes de différents types ces graphes sont appelés **graphes multi-couches** [7, 39].

Il existe de très nombreux domaines dans lesquels nous trouvons des graphes multicouches, comme par exemple : des réseaux sociaux portant sur un même ensemble de personnes mais impliquant différents types de relations (*e.g.* des réseaux sociaux : Facebook, Twitter, Google+; des réseaux professionnels : LinkedIn - ResearchGate, Academia, etc.); des réseaux d'interaction protéine-protéine où chaque relation représente un type différent d'interaction [75]; des réseaux bibliographiques où les sommets sont des auteurs et les relations représentent un type de publication (conférence/journal) [7]; un graphe de connaissances au format RDF dans lequel la même paire de sommets sujet/objet est connectée par différents prédicats [46].

Considérant l'importance d'une telle structure pour représenter et modéliser l'information du monde réel, la conception d'outils visuels pour améliorer l'analyse des grands graphes multicouches est essentielle. L'une des manières d'analyser ces données consiste, à partir d'une requête décrivant un sous-graphe, de rechercher ses occurrences dans le graphe global. C'est dans ce contexte que se situent les travaux présentés dans ce chapitre.

Les différents systèmes de « **requêtage**[1] » de graphes [6, 13, 40, 55, 57, 58] se focalisent sur des graphes constitués d'un seul type d'arêtes et ne traitent donc pas l'aspect multicouches évoqué précédemment. De plus, leur capacité d'exploration des résultats reste limité car ils ne permettent pas de visualiser les résultats sur le graphe d'origine. Récemment, des chercheurs de la communauté *InfoVis* ont commencé à proposer des techniques de visualisation pour traiter les graphes multicouches (voir par exemple [9, 60, 61]) mais aucune de ces approches ne propose de définir une requête graphique et d'interagir avec le moteur de requête associé.

En considérant un système visuel de requêtage de graphes, nous pouvons mettre en évidence quatre tâches importantes : i) la construction de requêtes, ii) la visualisation des résultats, iii) l'exploration des résultats et iv) la suggestion de requêtes.

1. Nous utilisons le néologisme *requêtage* pour décrire un système de requête de graphes qui visualise des résultats.

A notre connaissance, il n'existe pas d'approches qui traitent simultanément ces quatre tâches sur des graphes classiques (*i.e.* un seul type de relation) ou multicouches. La plupart des travaux de recherche se concentrent principalement sur la *construction de requêtes* [6, 13, 40, 55, 57, 58] et la *visualisation de résultats* [6, 13, 55, 57, 58]. Peu d'approches abordent *l'exploration de résultats* [57] et la *suggestion de requêtes* [40].

Dans ce chapitre, nous proposons VERTIGo [2] [17,18] (Cf. Figure 1.3), une nouvelle plateforme visuelle qui permet d'interroger des grands graphes multicouches (ou simples), de visualiser/explorer les résultats et de suggérer de nouvelles extensions de requêtes basées sur la structure du graphe d'origine et les résultats obtenus. VERTIGo fournit de nouveaux mécanismes pour gérer les quatre tâches listées précédemment et généralement impliquées dans un processus de requêtage visuel. Ces tâches sont prises en charge par des vues coordonnées pour naviguer et explorer l'ensemble des résultats récupérés à travers différents niveaux de détail.

VERTIGo prend également en charge l'interaction entre l'utilisateur et le moteur de recherche. L'utilisateur peut démarrer, mettre en pause et relancer le moteur et ainsi naviguer/explorer les résultats partiellement collectés.

Ce chapitre est organisé comme suit : la section 3.2 présente l'état de l'art ainsi qu'une discussion. Les critères pour l'outil sont introduits dans la section 3.3. La section 3.4 décrit les encodages visuels utilisés et la section 3.5 les modes d'interaction entre les différents composants visuels. Les considérations techniques sont présentes dans la section 3.6. Un exemple d'utilisation de VERTIGo est proposé dans la section 3.7. Enfin, la section 3.8 conclut ce chapitre.

3.2 État de l'art

3.2.1 Systèmes visuels de requêtage de graphes

Généralement le requêtage visuel de graphes consiste à utiliser une interface pour construire la structure d'un sous-graphe à rechercher et ensuite à lancer cette requête sur un moteur de recherche.

La recherche de toutes les occurrences d'un sous-graphe dans un graphe est coûteuse en temps de calcul car elle implique de résoudre le problème d'isomorphisme de sous-graphes [32]. Récemment, pour palier ce problème, différents algorithmes ont été proposés [11, 16, 32, 66, 73].

2. https://youtu.be/9AqP3YcwcDY

Ces algorithmes sont actuellement utilisés par différents systèmes. Par exemple, GRAPHITE [13] (Cf. Figure 3.1) propose une interface visuelle pour construire une requête sur un réseau d'auteurs et de leurs publications. Il exploite l'algorithme G-Ray [66] pour récupérer les sous-graphes. VISAGE [58] (Cf. Figure 3.2) fournit aussi une interface visuelle pour guider la construction de requêtes. Dans cette approche, les données sont gérées par la plateforme Neo4j [3]. Dans VIGOR [57], la requête est d'abord exprimée en utilisant le langage Cypher [4], puis transformée en une requête visuelle. VIGOR utilise également Neo4j pour gérer les données.

FIGURE 3.1 – L'interface de GRAPHITE [13]. A gauche, la requête est construite visuellement. A droite, un panneau montre les résultats un par un.

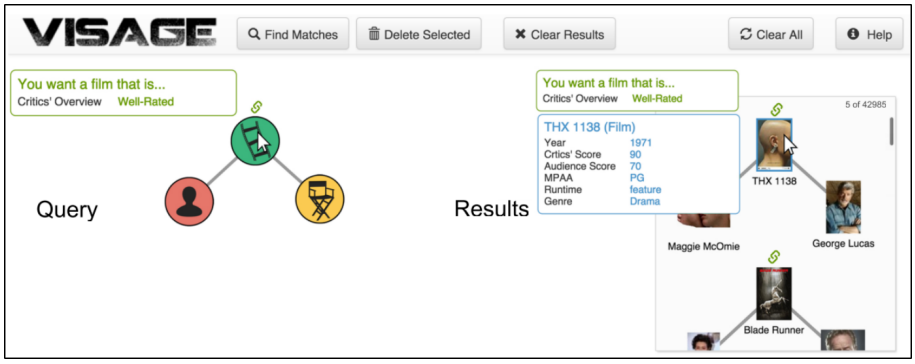


FIGURE 3.2 – L'interface de VISAGE [58]. A gauche, la requête est construite. A droite, les résultats sont affichés sous la forme d'une liste.

GraphVista [55] (Cf. Figure 3.4) exploite un mécanisme de streaming qui n'attend pas la fin du processus de requête pour visualiser les résultats. Ce système utilise un moteur de recherche spécifique [54] pour récupérer les résultats de la requête.

3. https://neo4j.com/
4. https://neo4j.com/docs/developer-manual/current/cypher/

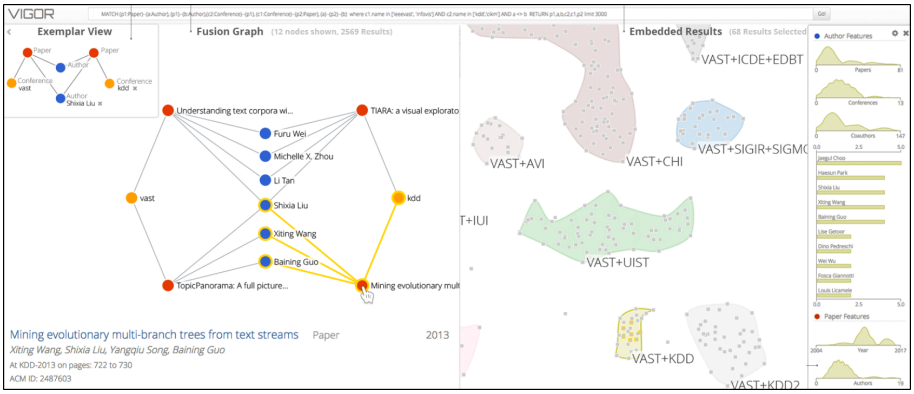


FIGURE 3.3 – L'interface de VIGOR [57]. En haut, la requête est exprimée en langage Cypher. A gauche, la requête est transformée en une requête visuelle. A droite, les résultats sont affichés sous la forme des clusters.

Une approche plus sophistiquée est proposée dans VOGUE [6] (Cf. Figure 3.5) où des algorithmes d'exploration de graphes (gSpan [73]) et d'isomorphisme de sous-graphes (VF2 [16]) guident le processus de requêtage de l'utilisateur. Cependant, cette approche considère une base de données comportant plusieurs graphes (*i.e.* une collection de graphes) et non un graphe unique comme dans notre contexte.

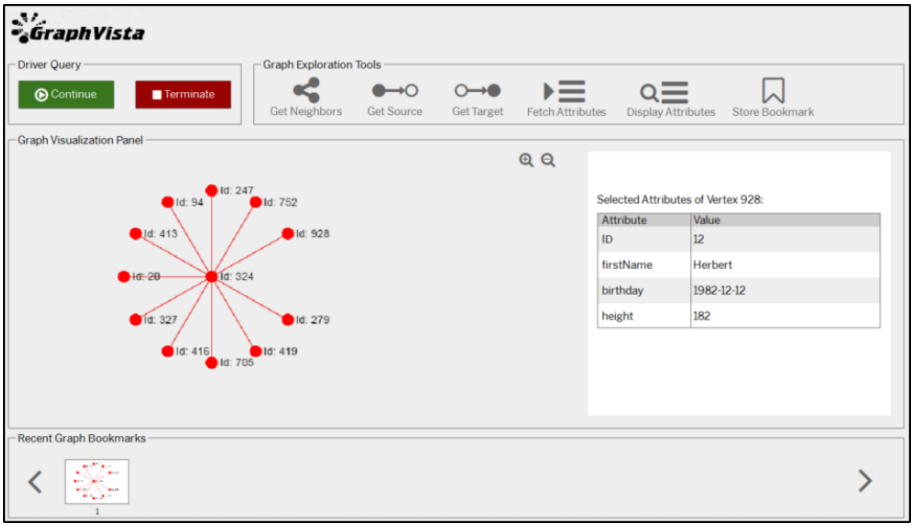


FIGURE 3.4 – GraphVista [55] permet de visualiser un à un les résultats dès qu'ils sont disponibles.

Il est également important de remarquer que les approches précédentes n'offrent pas de mécanismes d'interaction avec le moteur de recherche. Étant donné que la recherche d'un sous-graphe peut être une opération très longue, le fait d'avoir déjà des résultats partiels peut aider l'utilisateur à analyser des données. Seul GraphVista offre quelques fonctionnalités d'affichage de résultats partiels mais ne permet aucune interaction.

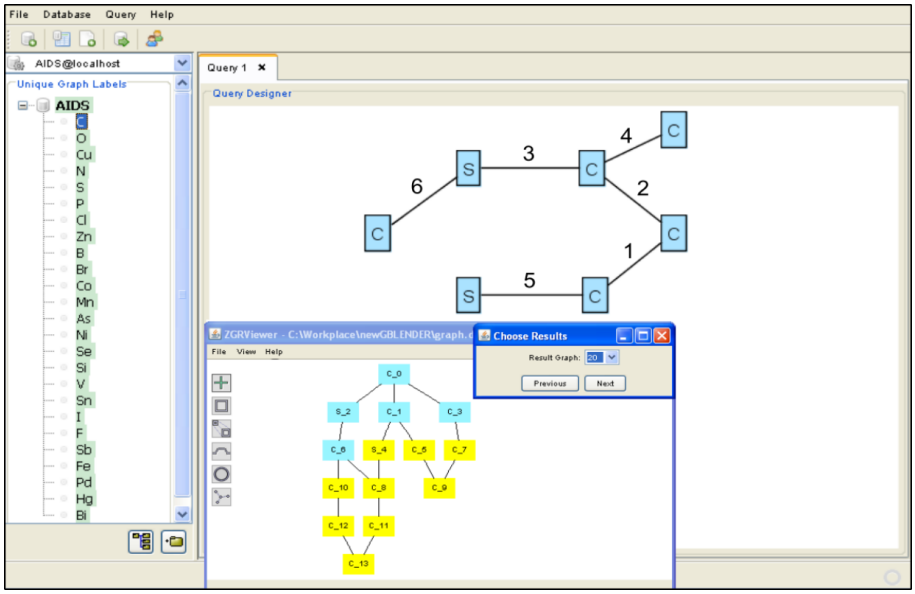


FIGURE 3.5 – VOGUE [6] permet de guider la construction de requêtes dans une base de données comportant plusieurs graphes.

3.2.2 Suggestion visuelle de requêtes

Traditionnellement la suggestion de requêtes se base sur les résultats de requêtes précédentes ou sur l'historique de l'utilisateur pour affiner la requête courante. Très peu de recherches ont été menées dans le cadre de la suggestion visuelle de requêtes [40, 52, 74].

Tous les systèmes précédents se concentrent principalement sur deux aspects : i) construire visuellement la requête et ii) afficher les résultats. Très peu d'efforts ont été faits pour réutiliser les résultats afin d'affiner la requête, c'est-à-dire suggérer une nouvelle structure de requête [40, 52, 74].

Par exemple, VIIQ [40] (Cf. Figure 3.6) propose une interface visuelle pour suggérer k arêtes pertinentes en se basant sur les actions précédentes de l'utilisateur sauvegardées dans un fichier journal. Cette approche échantillonne de manière aléatoire l'information d'un sous-ensemble de fichiers journaux, puis suggère des extensions de sommets/arêtes pour enrichir la structure de la requête courante. Ces suggestions ne sont pas visuellement intégrées dans l'interface de la construction de la requête mais sont fournies dans un panneau additionnel.

D'autres approches telles que [52, 74] proposent des suggestions de requêtes en utilisant une liste de sommets/arêtes possibles. Cependant, ces approches sont dédiées aux bases de données contenant plusieurs graphes.

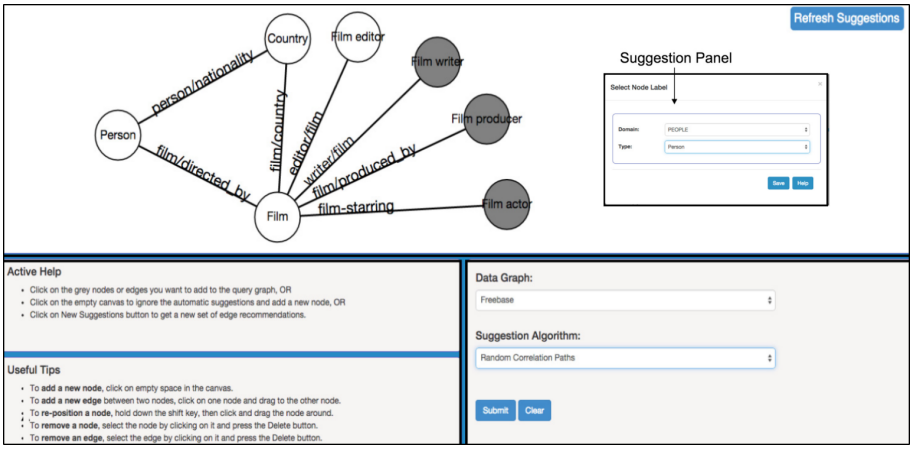


FIGURE 3.6 – VIIQ [40]. Le panneau de suggestion est présenté à droite de la requête.

3.2.3 Comparaison des approches

Le Tableau 3.1 présente une comparaison des différents systèmes de requêtage de graphes prenant en compte les tâches courantes généralement impliquées dans le processus : *construction de requêtes*, *visualisation des résultats*, *exploration des résultats* et *suggestion de requêtes*.

TABLE 3.1 – Comparaison des diverses approches pour leur capacité à prendre en charge les tâches de traitement de requêtes.

Approches	Construction de requêtes	Visualisation des résultats	Exploration des résultats	Suggestion de requêtes
GRAPHITE [13]	x	x		
VOGUE [6]	x	x		
GraphVista [55]	x	x		
VISAGE [58]	x	x		
VIGOR [57]	x	x	x	
VIIQ [40]	x			x
VERTIGO	x	x	x	x

Nous pouvons noter que les deux premières tâches sont supportées par presque tous les systèmes. GRAPHITE [13], VOGUE [6], VIIQ [40] et VISAGE [58] utilisent des techniques d'interaction standard pour dessiner visuellement la requête (*e.g.* sommets, arêtes et attributs) dans une vue dédiée et après visualiser successivement les résultats de la requête. Inversement, VIGOR [57] ne fournit pas de mécanisme visuel pour construire la requête mais utilise le langage Cypher. GraphVista [55] permet à l'utilisateur de construire la requête via une liste d'attributs de sommets, plutôt que de construire visuellement la structure de la requête.

Si l'on considère la tâche *d'exploration des résultats*, seul le système VIGOR [57] fournit des mécanismes permettant de parcourir ces résultats via des vues coordonnées. En ce qui concerne la tâche de *suggestion de requêtes*, VIIQ [40] est le seul système qui fournit des suggestions en fonction des actions précédentes de l'utilisateur.

Pour résumer, la plupart des approches se concentrent sur des tâches telles que la *construction de requêtes* et la *visualisation des résultats* plutôt que sur *l'exploration des résultats* et la *suggestion de requêtes*. En outre, à notre connaissance, il n'y a aucun système visuel qui traite les quatre tâches simultanément. De plus, les approches précédentes ne considèrent que les graphes standards (*i.e.* les graphes où toutes les arêtes sont de même type). VERTIGO vise à combiner ces quatre tâches dans le cas de graphes multicouches.

Dans ce chapitre, nous présentons l'approche VERTIGO, un système visuel de requêtage de graphes spécialement conçu pour les graphes multicouches (*i.e.* des graphes comportant différents types d'arêtes). VERTIGO introduit de nouveaux mécanismes pour prendre en charge l'interaction entre l'utilisateur et le moteur de recherche, l'exploration/navigation des résultats et la suggestion de requêtes basée sur la structure du graphe.

3.3 Critères

De nombreuses questions peuvent se poser pour l'analyse de graphes multicouches. Comme dans le chapitre précédent, nous considérons un exemple d'utilisation pour mettre en évidence une liste de critères pour notre approche. Dans ce scénario, nous considérons un réseau de co-auteurs dans deux domaines de recherche : *Information Visualization* et *Data Mining*.

Dans ce cadre, un utilisateur cherche à acquérir des connaissances sur des auteurs qui ont publié dans les deux domaines de recherche. A cet effet, l'utilisateur commence par une requête où l'objectif est de récupérer des groupes d'auteurs qui collaborent ensemble et dont l'un d'eux a des publications dans les deux domaines : il/elle a publié un article dans *TVCG* avec un auteur et il/elle a également publié dans *ICDM* et *KDD* avec un autre auteur. Ce type de requête donnera à l'utilisateur une indication sur les auteurs qui servent de lien entre les deux communautés.

Ensuite, l'utilisateur démarre le moteur de recherche sur le graphe. Une fois que les résultats (groupes d'auteurs) sont extraits, l'utilisateur s'attend à en avoir un aperçu. À partir de cette vue globale, l'utilisateur peut repérer dans quelles communautés

les auteurs sont situés par rapport au réseau de co-auteurs, et ainsi obtenir un point de départ pour une analyse plus détaillée. Dans cette analyse, l'utilisateur pourrait être intéressé par :

— Les emplacements des auteurs dans le réseau de co-auteurs.
— Tous les résultats où un auteur est présent.
— Quel est l'auteur qui a publié le plus ?
— L'auteur X et l'auteur Y ont-ils publié ensemble ?

Sur la base de ces informations, l'utilisateur pourrait affiner sa requête initiale. Par exemple, y a-t-il une autre conférence (*e.g.* InfoVis, Sigmod, etc.) qui n'a pas été prise en compte dans la requête initiale ?

À partir de ce scénario, nous pouvons identifier la liste de critères suivante :

[R1] Construction des requêtes. Ce critère vise à fournir à l'utilisateur un outil pour construire visuellement une requête. Cette requête servira de point de départ pour trouver toutes ses occurrences dans le graphe. Dans l'exemple précédent, l'utilisateur peut facilement représenter les auteurs et les conférences/journaux reliant ces auteurs (TVCG, ICDM et KDD).

[R2] Navigation et exploration des résultats. Ce critère vise à fournir des visualisations et des interactions pour faciliter l'exploration des résultats à différents niveaux de détail.

[R3] Manipulation des graphes multicouches. Ce critère vise à permettre à l'utilisateur de modéliser et de représenter un graphe traditionnel comme un graphe multicouches, *i.e.* un graphe où chaque couche contient des arêtes d'un certain type. En se référant à l'exemple précédent, une couche peut être une conférence/journal spécifique dans le réseau de co-auteurs.

[R4] Suggestion de requêtes. Ce critère vise à suggérer à l'utilisateur une extension de la requête initiale. En citant l'exemple précédent, il pourrait proposer une autre conférence basée sur la structure du graphe afin d'affiner les résultats.

[R5] Passage à l'échelle. Ce critère vise à gérer les requêtes sur des graphes volumineux et à récupérer les résultats dans un temps raisonnable (SRT, *System Reponse Time*). SRT est défini comme le temps pris par le moteur de recherche pour évaluer la requête entière.

Cette liste de critères permet de concevoir un système capable de supporter les tâches communes et spécialisées du processus de requêtage de graphes. Puisque les graphes sont omniprésents, les utilisations possibles sont très vastes. Notre intention est de permettre l'utilisation de VERTIGo par le grand public, ainsi que par des utilisateurs spécialisés.

3.4 Encodages visuels et modes d'interaction

Cette section décrit notre système visuel. De la même façon que dans le chapitre précédent, nous suivons le principe du mantra de recherche visuelle d'information proposé par Shneiderman : « tout d'abord, vue d'ensemble, puis zoom et filtre, et enfin, détails à la demande » [65].

3.4.1 Conception visuelle

Sur la base des critères définis précédemment, nous proposons VERTIGo (Cf. Figure 1.3). Il est basé sur trois composants visuels principaux :

— Une *vue de la requête* qui permet de construire/suggérer visuellement la structure de la requête et de la lancer sur le moteur de recherche *[R1, R4, R5]* (Cf. Figure 1.3a).
— Une *vue du graphe* qui présente la structure du graphe ainsi que les résultats de la requête à différents niveaux de détails *[R2, R3, R5]* (Cf. Figure 1.3(b,e)).
— Une *vue des résultats* qui décrit en détail la structure des résultats pour un sous-ensemble d'entre eux *[R2, R3]* (Cf. Figure 1.3d).

Ces composants visuels sont décrits en détail dans les sections suivantes.

3.4.2 Vue de la requête

Dans le but d'interroger le graphe, nous proposons une interface graphique interactive et intuitive pour construire visuellement une requête. La Figure 3.7 montre l'interface de la *vue de la requête*. Cette vue permet à l'utilisateur de définir la structure de la requête (sommets, arêtes, attributs de sommets) via des fonctionnalités interactives *[R1]*. Cette interface est constituée d'un espace pour dessiner, de deux barres d'outils et d'une barre d'état.

— **L'espace de dessin** (Cf. Figure 3.7a) représente la zone où la requête est construite et où les interactions se produisent.
— **La barre de conception** (Cf. Figure 3.7b) fournit différentes actions pour construire la requête. De haut en bas, ces actions sont : ajouter un sommet, ajouter une arête d'un certain type, supprimer un sommet/arête, lancer le mécanisme de suggestion d'arêtes et organiser la disposition des sommets de la requête en appliquant un algorithme de force.

— **La barre standard** (Cf. Figure 3.7c) inclut des fonctionnalités fréquemment utilisées : télécharger, créer et enregistrer une requête. Elle contient également le bouton *Search* pour la lancer sur le moteur de recherche.

— **La barre d'état** (Cf. Figure 3.7d) fournit des informations textuelles sur l'état actuel de la vue.

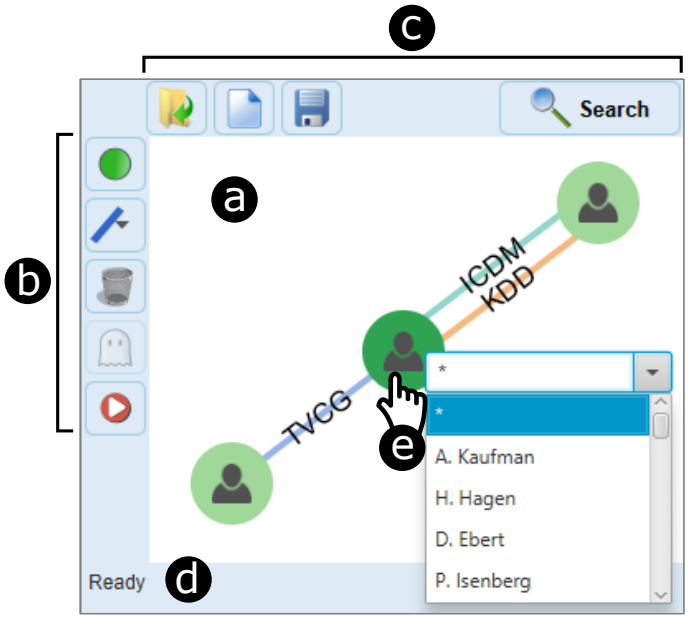


FIGURE 3.7 – La *vue de la requête* affichant un exemple de requête d'un graphe multi-couches. (a) La requête est représentée dans l'espace de dessin. (b) La barre de conception fournit des interactions pour construire la requête. (c) La barre standard fournit des fonctionnalités classiques et contient également le bouton *Search* qui lance la requête sur le moteur de recherche. (d) La barre d'état affiche des informations textuelles sur la vue. (e) L'utilisateur peut spécifier un attribut sur un sommet.

Des actions telles que la spécification d'un attribut sur un sommet, la spécification de différents types d'arêtes, l'exécution du moteur de recherche et le lancement du mécanisme de suggestion d'arêtes nécessitent des interactions spécifiques décrites ci-dessous.

3.4.2.1 Spécification d'un attribut sur un sommet

Un attribut spécifique peut être ajouté à un sommet de la requête afin d'affiner les résultats et de diminuer le temps de réponse (SRT) du processus de requêtage *[R5]*. Cette action est réalisée via un clic droit sur le sommet souhaité et en sélectionnant une valeur dans la liste déroulante affichée. Par exemple sur la Figure 3.7e, on peut sélectionner le nom d'un auteur dans un réseau de co-auteurs.

3.4.2.2 Spécification d'un type d'arête

VERTIGo permet d'interroger des graphes standards (sommets connectés par des arêtes d'un seul type) ou des graphes multicouches (sommets connectés par différents types d'arêtes) *[R3]*. Nous utilisons différentes couleurs pour visualiser les différents types d'arêtes.

Quand une paire de sommets est connectée par plusieurs arêtes, certains problèmes se posent : i) comment organiser les différents liens et ii) comment visualiser les différents types d'arêtes. Pour surmonter ces problèmes, les arêtes entre une paire de sommets sont disposées de manière parallèle. La distance relative entre les arêtes parallèles reste la même, même lorsque leur nombre augmente, ce qui permet de voir clairement toutes les arêtes, comme l'illustre la Figure 3.7. La limite du nombre d'arêtes est de 10 en raison du problème connu de distinction de couleurs [70]. En outre, les arêtes sont également étiquetées pour faciliter la reconnaissance de leur type.

Pour ajouter une arête à une requête, l'utilisateur doit d'abord sélectionner un type à partir de la barre de conception (Cf. Figure 3.7b). Après, en utilisant le pointeur de la souris, il dessine l'arête d'un sommet à un autre. Enfin, l'arête est automatiquement positionnée.

La Figure 3.7 montre l'exemple de la requête décrite dans la section 3.3 où un auteur a publié un article dans *TVCG* avec un auteur et il/elle a également publié un autre article à *ICDM* et *KDD* avec un autre auteur. Nous pouvons observer que la structure de graphe multicouches nous permet de spécifier facilement plusieurs arêtes entre la même paire de sommets.

3.4.2.3 Exécution du moteur de recherche

Comme un graphe multicouches implique différents types de liens entre les sommets, les moteurs de recherche standards tels que TurboISO [32] ou VF3 [11] ne sont pas adaptés. Pour traiter les graphes multicouches *[R3]*, nous avons intégré dans VERTIGo un moteur de recherche de graphes multicouches récent nommé *SuMGra* [37]. Ce moteur exploite des techniques d'indexation spécialisées afin d'accélérer l'algorithme de *backtraking* couramment utilisé pour traiter le problème de l'isomorphisme de sous-graphes [32]. Le bouton *Search* dans la barre standard (Cf. Figure 3.7c) démarre l'exécution du moteur.

Un paradigme commun dans le processus de requêtage est basé sur une approche *Requête→Résultats*, c'est-à-dire, construire la requête, puis l'envoyer au moteur de recherche, et enfin visualiser les résultats [55]. Cependant, dans ce paradigme, aucune interaction entre l'utilisateur et le moteur de recherche n'est proposé, ce qui

peut augmenter le temps d'attente (SRT) lorsque le nombre de résultats augmente. Afin d'améliorer le SRT *[R5]*, VERTIGO interagit étroitement avec le moteur SuMGra pour permettre à l'utilisateur de démarrer, de mettre en pause et de reprendre l'exécution de la requête avec la possibilité de naviguer/explorer des résultats partiellement collectés *[R2, R5]* (cette fonctionnalité est décrite dans la section 3.5).

3.4.2.4 Visualisation des suggestions de requêtes

Basé sur la structure de la requête initiale, VERTIGO suggère automatiquement k arêtes à l'utilisateur *[R4]*. Afin d'obtenir la liste des *arêtes candidates*, VIIQ [40] utilise un sous-ensemble d'arêtes extraites aléatoirement d'un fichier journal historique. VERTIGO améliore cette approche en classant toutes les arêtes adjacentes à chaque sommet des résultats, puis en suggérant les top-k arêtes les plus fréquemment trouvées. De plus, ce mécanisme utilise la structure réelle du graphe pour suggérer des extensions et non l'historique de l'utilisateur.

VERTIGO propose deux types d'arêtes candidates: des *arêtes internes* qui relient les sommets déjà présents dans la requête et des *arêtes externes* qui relient les sommets de la requête à d'autres sommets. Les arêtes internes et externes sont affichées directement sur la requête.

La Figure 3.8 montre la requête de la Figure 3.7 avec des suggestions visuelles. Les arêtes internes sont représentées par des lignes en pointillés pour les différencier des arêtes existantes (par exemple dans la Figure 3.8(a,b) les liens en pointillés représentent la conférence de *PacificVis* et le journal *CG&A* reliant les sommets initiaux de la requête).

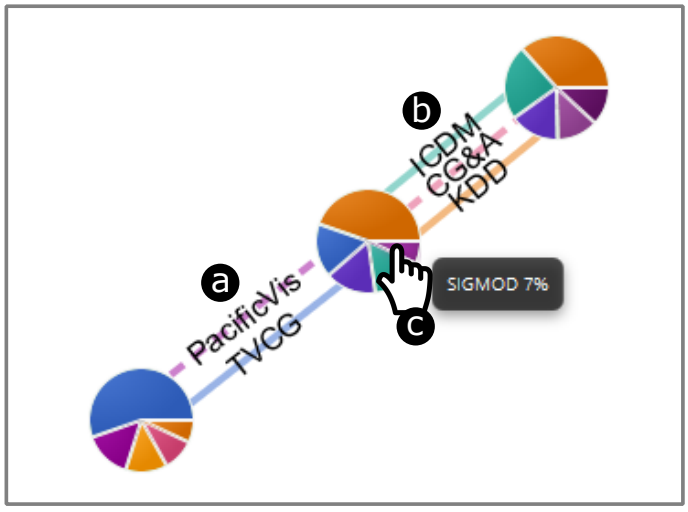


FIGURE 3.8 – Un exemple de suggestion d'une requête. Des éléments visuels affichent la suggestion des arêtes.

Comme les arêtes externes étendent la requête vers un sommet non représenté, et dans le but de représenter k suggestions d'arêtes, nous utilisons un camembert où le nombre de tranches est égal à k (Cf. Figure 3.8c). Le pourcentage affiché sur une tranche fournit le nombre d'arêtes dans le graphe supportant l'extension d'un type d'arête particulier depuis un sommet de la requête. Par exemple, la Figure 3.8c indique que 7% des résultats déjà trouvés peuvent être étendus avec le type d'arêtes *SIGMOD*. Lorsque la souris est déplacée sur une tranche, une info-bulle affiche le pourcentage et l'étiquette du type d'arête auquel elle appartient.

La Figure 3.9 montre la requête après avoir ajouté les suggestions mentionnées précédemment. Afin d'accepter une suggestion, l'utilisateur fait un clic sur l'arête interne (ligne en pointillés) ou externe (tranche du camembert). Dans le premier cas, une nouvelle arête interne est affichée en tant que ligne continue (Cf. Figure 3.9(d,e) les liens *PacificVis* et *CG&A*). Dans le deuxième cas, une nouvelle arête externe est affichée en tant que lien vers un sommet dans la direction de sa tranche (Cf. Figure 3.9f le nouveau lien *SIGMOD*).

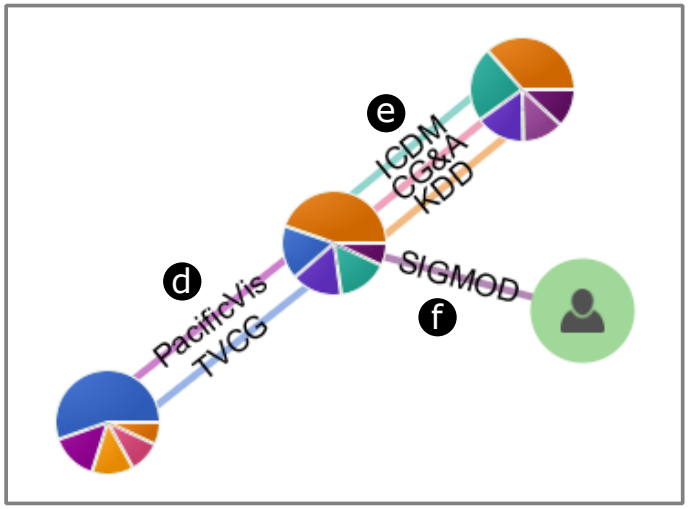


FIGURE 3.9 – La requête de la Figure 3.8 incluant des arêtes suggérées et valides.

Le mécanisme de suggestion de requêtes fournit des éléments visuels pour guider l'utilisateur dans la construction incrémentale de requêtes *[R1, R4]*. Ainsi, chaque fois que la requête est étendue, l'utilisateur peut récupérer les résultats et affiner le processus de recherche. Ce mécanisme de suggestion est activé depuis la barre de conception (Cf. Figure 3.7b).

3.4.3 Vue du graphe

Nous adoptons un diagramme de type nœuds-liens avec un algorithme de force [43] pour produire une visualisation appropriée du graphe. Travailler avec des graphes volumineux implique certains défis : critères esthétiques, passage à l'échelle et temps d'interaction raisonnable. Pour résoudre tous ces problèmes, nous utilisons une technique appelée *Multipole Multilevel Method* (FM^3) [31]. FM^3 est une approche bien connue dans le domaine du dessin de graphes qui permet de traiter de grands graphes avec un bon compromis entre le temps de dessin et la qualité de la visualisation *[R2, R5]*.

Un exemple de visualisation d'un graphe obtenue par VERTIGo est présenté dans la Figure 3.10. Le graphe en entrée correspond à un ensemble de données biologiques [8] modélisé sous la forme d'un graphe multicouches. Ce graphe est composé de 38 936 sommets, de 310 664 arêtes et de 7 types d'arêtes (les couches du graphe). Nous notons qu'une telle visualisation a l'avantage de distinguer des groupes de sommets qui forment clairement des communautés. Dans cet exemple, nous pouvons facilement mettre en évidence quatre groupes principaux (clusters) de sommets.

Un diagramme nœuds-liens souffre d'un problème d'encombrement visuel (cluttering) dû aux nombreux croisements d'arêtes dans les graphes volumineux. Dans notre approche, les arêtes du graphe ne sont pas affichées pour simplifier la visualisation.

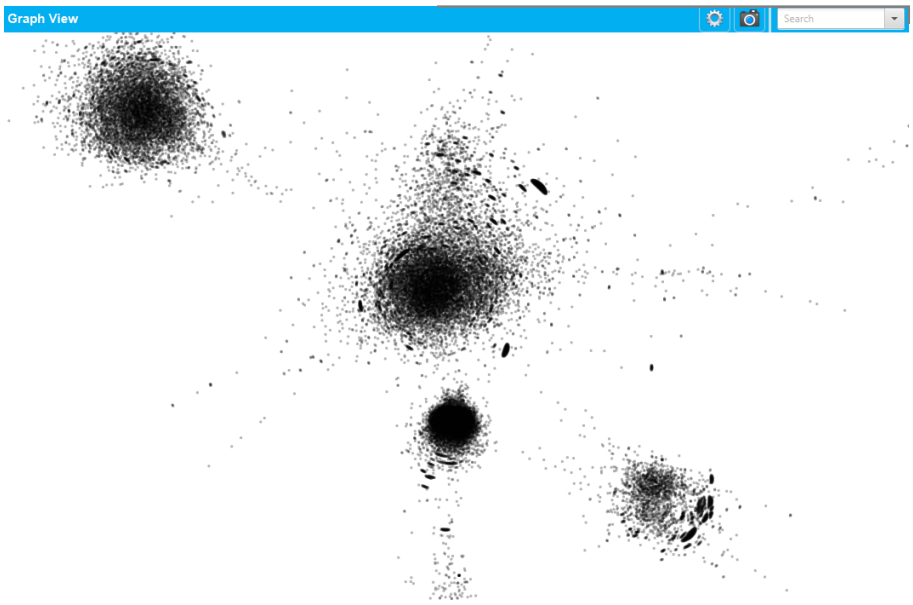


FIGURE 3.10 – La *vue du graphe* affichant un jeu de données biologiques. La visualisation obtenue permet d'identifier facilement quatre communautés.

3.4.4 Vue des résultats

Pour afficher un grand nombre de résultats, Pienta *et al.* [57] les ont réduit en points. Ce genre d'abstraction évite d'encombrer la visualisation. Cependant, la distribution spatiale des éléments (*i.e.* les sommets) est ainsi perdue. Dans le but d'afficher en détail la distribution spatiale et la topologie des résultats (*i.e.* sommets et arêtes) *[R2]*, nous proposons la *vue des résultats* (Cf. Figure 3.11). Dans cette vue, nous affichons en détail les résultats (Cf. Figure 3.11b) contenant un ensemble donné de sommets (Cf. Figure 3.11a) (les interactions pour sélectionner ces sommets sont décrites dans la section 3.5).

L'objectif de la vue des résultats est, à partir d'une sélection de sommets dans le graphe, d'afficher les résultats associés. Par exemple, la Figure 3.11b affiche la partie de la liste des résultats d'une requête contenant les sommets *S. Liu* et *Y. Song*. Ces sommets ont été sélectionnés par l'utilisateur (Cf. Figure 3.11a). Cette fonctionnalité améliore la tâche d'exploration *[R2]* car elle affiche non seulement ces résultats mais, surtout elles les regroupe.

Nous décrivons à présent comment cette vue prend en compte l'information topologique et spatiale des résultats.

Structure topologique. Dans le but de décrire la structure topologique des résultats, nous devons prendre en compte le fait que certains d'entre eux contiennent le même ensemble de sommets reliés avec des arêtes différentes. Sur la base de cette caractéristique, nous regroupons les résultats ayant le même ensemble de sommets dans un seul *résultat fusionné*.

La Figure 3.12 montre un exemple où le résultat (c) correspond à la fusion des autres résultats (a) et (b). Le nombre d'agrégations est représenté par l'épaisseur des arêtes. Sur la Figure 3.12c, nous pouvons remarquer l'épaisseur de l'arête bleue entre les sommets B et C.

Distribution spatiale. Afin de montrer la distribution spatiale des résultats, nous utilisons la valeur de la boîte englobante (MBR) des sommets des résultats dans la vue du graphe. Par exemple, une faible valeur de MBR montre la proximité de ces sommets, tandis qu'une valeur élevée de MBR signifie que les sommets sont éloignés les uns des autres (éventuellement dans des clusters différents).

La liste des résultats fusionnés (Cf. Figure 3.11b) est présentée avec leurs métadonnées (*i.e.* la valeur de MBR et le nombre d'agrégations). Ces métadonnées peuvent être particulièrement utiles pour explorer les résultats en fonction de leurs caractéristiques spatiales/topologiques.

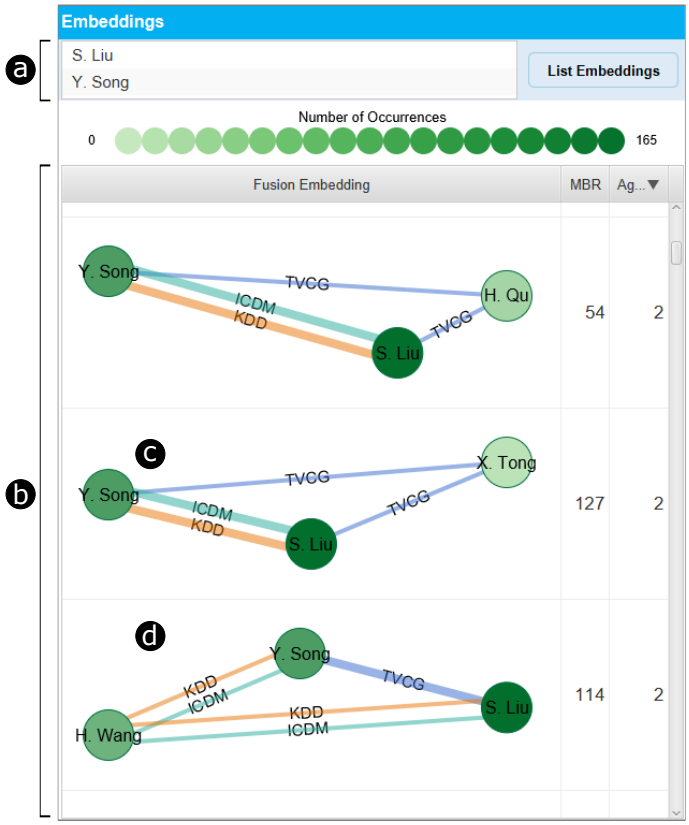


FIGURE 3.11 – La *vue des résultats* affiche une liste de résultats (b) pour un ensemble de sommets (a). (c)(d) Exemples de résultats fusionnés. La saturation des couleurs des sommets indique le nombre de résultats auxquels le sommet appartient. L'épaisseur des arêtes représente le nombre d'agrégations.

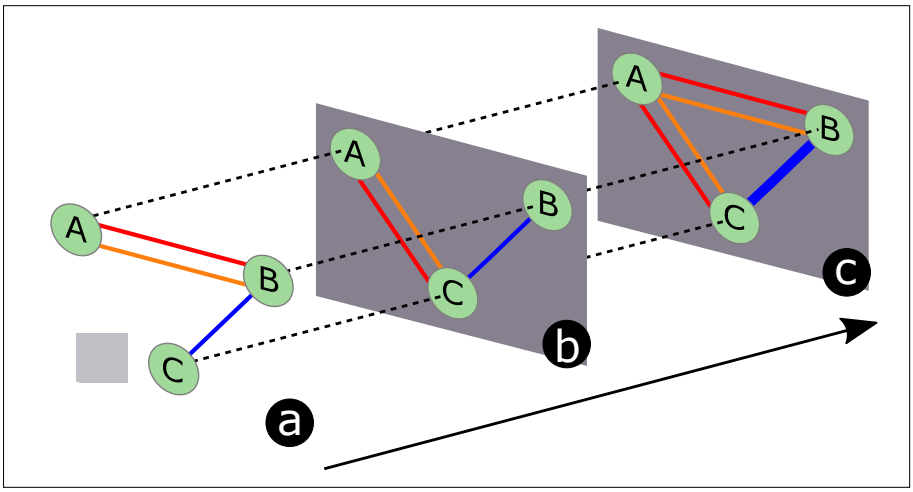


FIGURE 3.12 – Construction d'un résultat fusionné (c) à partir de la fusion des occurrences (a) et (b). L'épaisseur des arêtes montre le nombre d'agrégations.

Par exemple, dans un réseau de co-auteurs, étant donné un ensemble d'auteurs (Cf. Figure 3.11a), si l'utilisateur veut trouver d'autres auteurs qui collaborent souvent avec eux, il/elle peut ordonner la liste des résultats (Cf. Figure 3.11b) et récupérer les résultats avec le plus grand nombre d'agrégations (la dernière colonne à droite de la Figure 3.11b).

La Figure 3.11(c,d) montre des exemples de résultats fusionnés. La saturation des couleurs des sommets révèle le nombre de résultats auxquels le sommet appartient. Par exemple, la structure fusionnée sur la Figure 3.11c révèle que *S. Liu* apparaît dans plus de résultats que *X. Tong*. La Figure 3.11d révèle une corrélation plus élevée entre Y. Song et S. Liu (épaisseur de la liaison *TVCG*) plutôt qu'avec *H. Wang* (épaisseur des liaisons *KDD* et *ICMD*). Comme dans cet exemple la fusion correspond à l'agrégation de deux résultats, en tenant compte de l'épaisseur des liens, nous pouvons déduire : i) *Y. Song* et *S. Liu* ont publié deux fois ensemble un article dans *TVCG*, ii) *H. Wang* et *Y. Song* ont publié une fois ensemble à *KDD* et *ICDM* iii) *H. Wang* et *S. Liu* ont également publié ensemble une fois à *KDD* et *ICDM*.

Dans cette section, nous avons présenté les trois principaux composants visuels de VERTIGO : la *vue de la requête*, la *vue du graphe* et la *vue des résultats*. Chaque vue permet d'effectuer une tâche spécifique du processus de requêtage. La *vue de la requête* permet le dessin/suggestion interactive d'une requête. La *vue du graphe* affiche la structure du graphe et permet de naviguer/explorer les résultats à différents niveaux de détail. La *vue des résultats* permet à l'utilisateur de visualiser une liste de résultats pour d'autres analyses (*e.g.* spatiales ou topologiques). Dans la section suivante, nous décrivons les interactions entre ces composants visuels.

3.5 Interactions

Dans cette section, nous décrivons les interactions entre les composants visuels de VERTIGO. Les interactions visent à aider l'utilisateur dans les différents tâches : construction de requêtes, visualisation des résultats, exploration des résultats et suggestion de requêtes. La Figure 3.13 montre l'architecture générale de VERTIGO où les flèches représentent les interactions entre les composants.

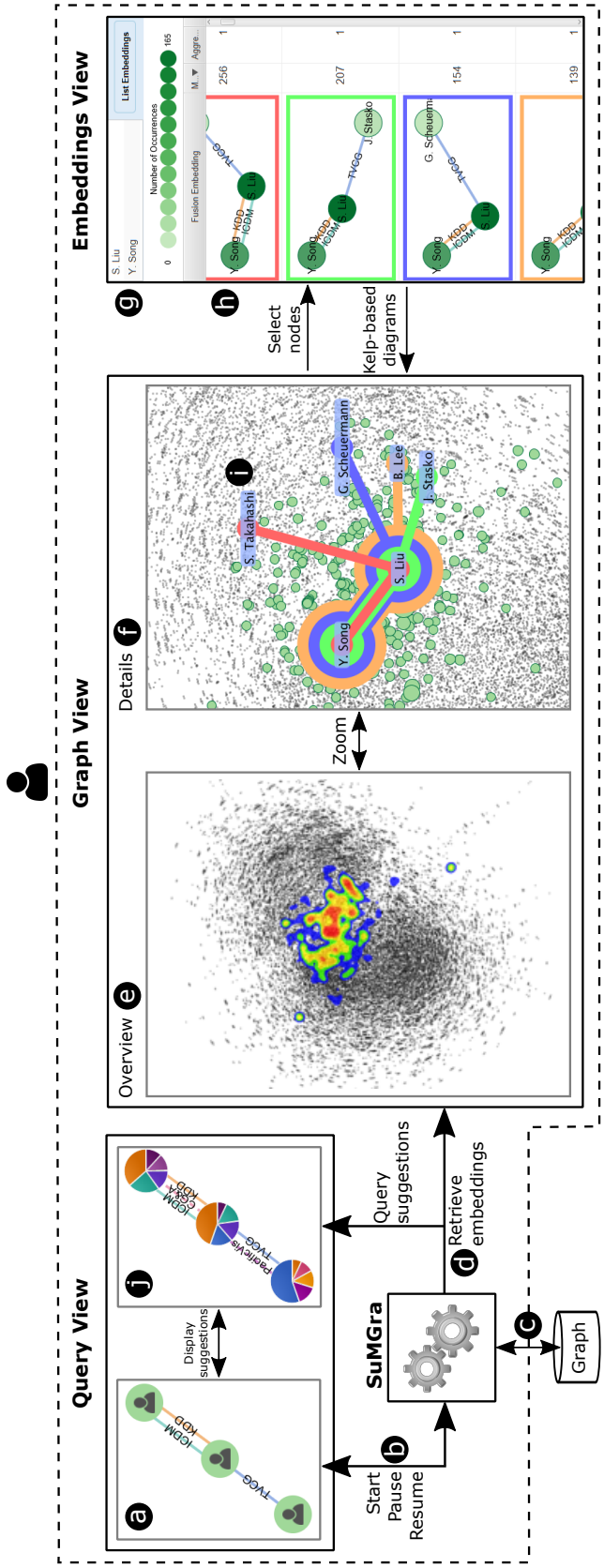


FIGURE 3.13 – L'architecture du système VERTIGO. Les flèches montrent les interactions entre les composants.

3.5.1 Visualiser les résultats

L'utilisateur démarre le processus de requêtage visuel en dessinant une requête (Cf. Figure 3.13a). Une fois la construction de la requête terminée, l'utilisateur envoie la requête au moteur de recherche (Cf. Figure 3.13b). Ensuite, le moteur recherche les résultats (Cf. Figure 3.13(c,d)) et les renvoie dans la vue du graphe. Sur la base des résultats récupérés et de la structure du graphe, VERTIGo suggère k arêtes afin d'affiner la requête précédente et de la ré-exécuter (Cf. Figure 3.13j).

Comme le nombre de résultats peut être important, leur visualisation est un défi. Pour surmonter ce problème, VERTIGo fournit deux niveaux d'analyse différents : i) un niveau global dans lequel il résume les résultats en considérant le graphe (Cf. Figure 3.13e) et ii) un niveau détaillé dans lequel il permet à l'utilisateur d'inspecter des sommets particuliers contenus dans les résultats (Cf. Figure 3.13f). L'utilisateur navigue entre ces deux niveaux en effectuant un zoom sémantique (*i.e.* en modifiant les niveaux de détail du graphe et des résultats).

3.5.1.1 Niveau global

Pour le niveau global, nous utilisons une carte de chaleur [56] qui montre la partie du graphe où les résultats sont situés. La Figure 3.14 montre un exemple d'une telle carte où la distribution des résultats est représentée par un gradient de couleur. Ce gradient varie selon une palette de couleurs du bleu (zones moins denses) au rouge (zones denses). Dans l'exemple de la Figure 3.14, nous pouvons remarquer que la vue du graphe nous révèle différentes communautés (l'algorithme de force met en évidence différents groupes de sommets). Dans cet exemple, les résultats ne sont pas répartis équitablement entre ces communautés. En particulier, nous pouvons noter une densité élevée (forte concentration de résultats en rouge) au centre de l'image alors qu'une plus faible concentration de résultats apparaît ailleurs.

Cette carte de chaleur peut être utilisée pour comprendre visuellement si une requête est spécifique à une région du graphe ou non et fournit ainsi à l'utilisateur un point de départ pour la tâche d'exploration *[R2]*.

Nous rappelons que VERTIGo permet à l'utilisateur de mettre en pause et de relancer le moteur de recherche (Cf. Figure 3.13b). Cela peut être particulièrement utile si le nombre de résultats augmente rapidement *[R5]*. Dans ce scénario, le processus peut être suspendu à tout moment, les résultats sont récupérés en considérant l'état actuel des résultats trouvés par le moteur de recherche et la carte de chaleur correspondante (Cf. Figure 3.13e) est produite ou mise à jour. Enfin, l'utilisateur peut reprendre le processus de requête à partir de l'endroit où il s'est arrêté.

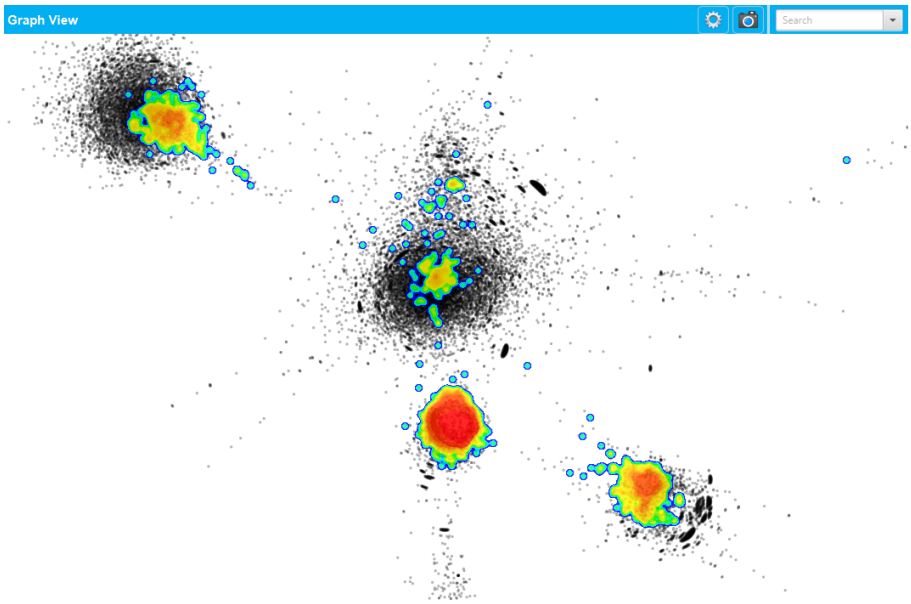


FIGURE 3.14 – La *vue du graphe* affichant le jeu de données biologique de la Figure 3.10. Une carte de chaleur affiche les emplacements des résultats récupérés sur le graphe.

Cependant, l'abstraction fournie par la carte de chaleur peut rendre difficile la compréhension du nombre exact et de la distribution des résultats dans une région du graphe. Pour résoudre ce problème, VERTIGO permet de changer les niveaux de détails en se concentrant sur une zone particulière *[R2]* (Cf. Figure 3.13f).

3.5.1.2 Niveau détaillé

Guidé par la carte de chaleur, l'utilisateur peut zoomer sur une zone particulière du graphe pour effectuer une analyse plus approfondie *[R2]*. La Figure 3.15 affiche les résultats détaillés de la requête de la Figure 3.7. Les sommets des résultats sont représentés en vert. Nous utilisons la taille du sommet pour représenter le nombre de résultats auxquels il appartient : les sommets les plus grands correspondent aux sommets contenus dans le plus grand nombre de résultats.

Étiquetage des sommets des résultats. Afin d'aider l'exploration *[R2]*, les étiquettes des sommets des résultats sont également affichées. L'étiquetage des sommets n'est pas une tâche triviale, car l'affichage de toutes les étiquettes peut entraîner des problèmes de lisibilité. Pour surmonter ce problème, nous les affichons selon le poids des sommets. Le poids d'un sommet est donné par le nombre de résultats auxquels il appartient et la taille de la police est liée à ce poids. Basé sur cette caractéristique, nous proposons un algorithme qui affiche les étiquettes des sommets avec le plus de poids.

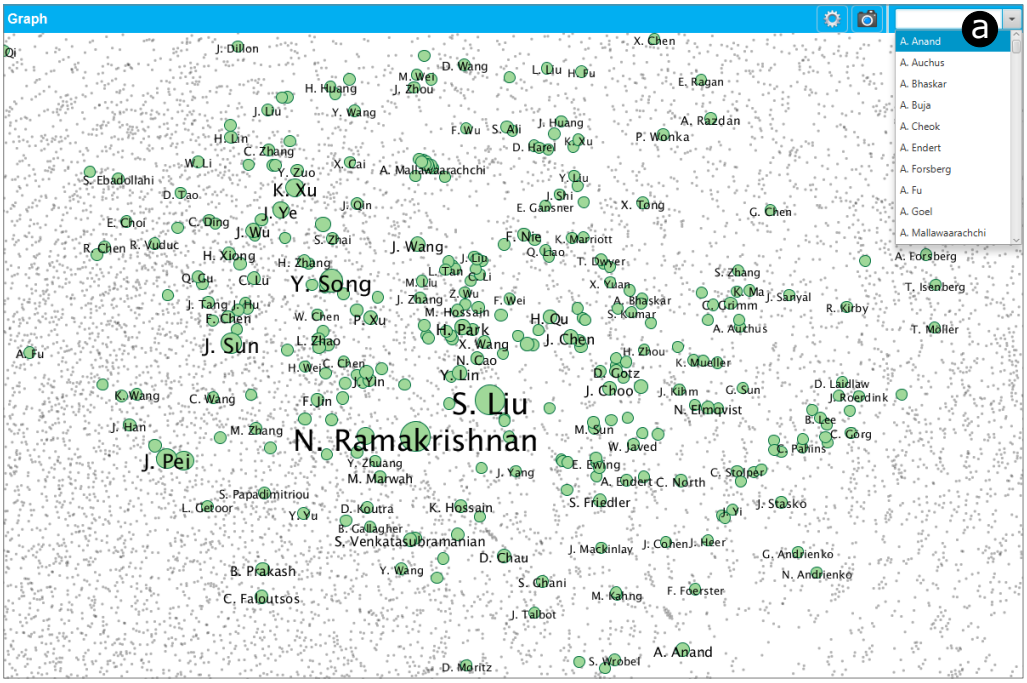


FIGURE 3.15 – La vue du graphe affichant les sommets des résultats en surbrillance. La taille des sommets représente le nombre de résultats auxquels ils appartiennent. (a) Un champ de texte permet de rechercher un sommet spécifique à partir de son nom.

L'algorithme trie d'abord l'ensemble des sommets selon leur poids. Ensuite, pour chaque sommet, il calcule la boîte englobante de l'étiquette et vérifie s'il n'y a pas de chevauchement avec les sommets étiquetés précédemment. Si aucun chevauchement n'existe, l'étiquette est affichée, sinon elle ne l'est pas. La complexité de cet algorithme est $O(n^2)$, où n est l'ensemble des sommets des résultats. Il dépend du niveau de zoom. Ainsi, lorsque l'utilisateur effectue un zoom avant/arrière, l'algorithme calcule à nouveau les étiquettes affichées. La Figure 3.15 en donne un exemple. Nous pouvons constater que certaines étiquettes sont plus grandes (*e.g. N. Ramakrishnan*) que leurs voisins.

L'algorithme d'étiquetage affiche clairement les étiquettes des sommets les plus grands. Cependant, en fonction du niveau de zoom, certains sommets avec un poids faible peuvent être cachés et donc difficile à localiser. Pour résoudre ce problème, nous avons muni notre système d'une fonctionnalité de recherche par champ de texte pour récupérer les sommets en fonction leur nom. La Figure 3.15a montre cette option. Une fois que l'utilisateur a saisi un nom de sommet, le centre de la vue du graphe est automatiquement redirigé vers l'emplacement du sommet et une animation indique son emplacement exact.

À partir du niveau détaillé, l'utilisateur peut sélectionner un ensemble de sommets pour une analyse plus approfondie. En cliquant dessus, leur couleur devient rouge et ils sont automatiquement ajoutés à la vue des résultats (Cf. Figure 3.13g). L'utilisateur peut ainsi visualiser la liste des résultats qui lui sont associés (Cf. Figure 3.13h). Dans la section suivante, nous décrivons la tâche d'exploration en utilisant à la fois les vues des résultats et celle du graphe.

3.5.2 Explorer les résultats

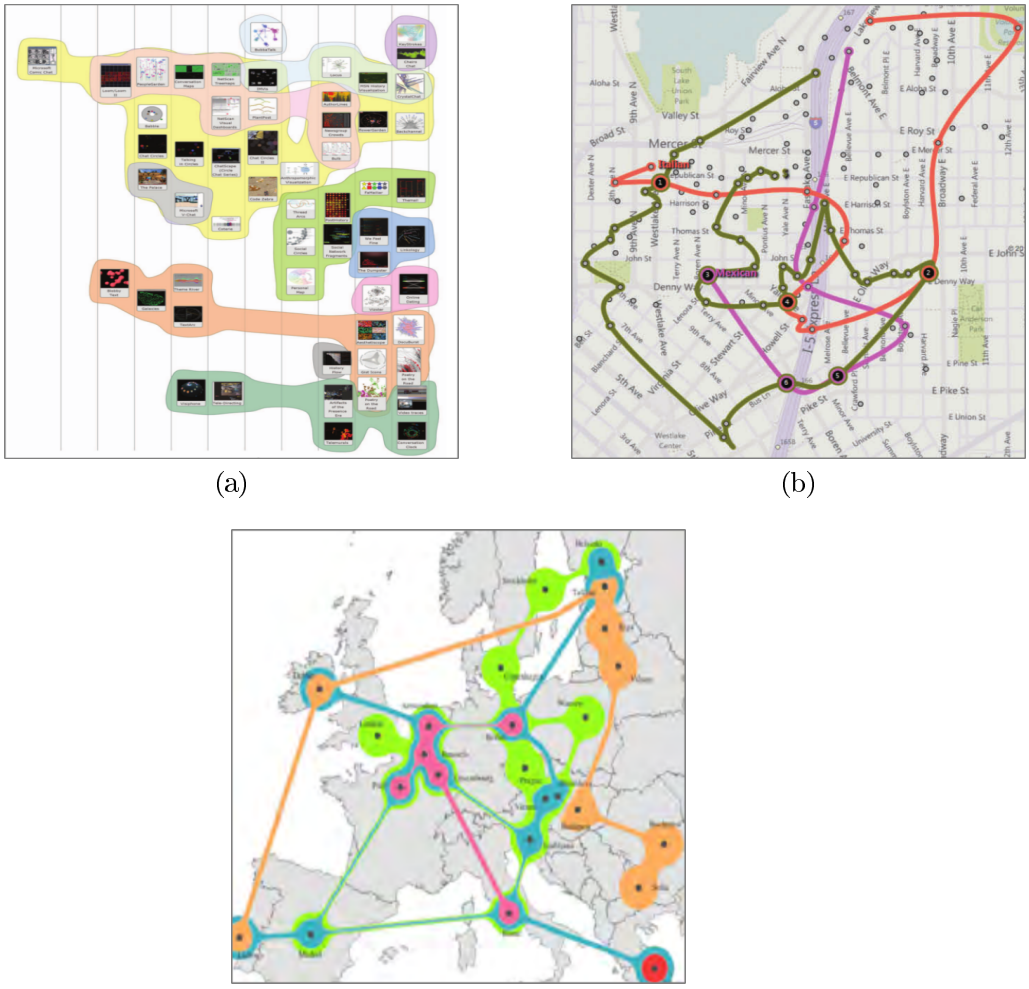
La vue des résultats montre la liste des occurrences fusionnées (Cf. Figure 3.13h) contenant un ensemble de sommets sélectionnés (Cf. Figure 3.13g). Cette liste permet à l'utilisateur d'explorer les résultats en fonction de leur étendue spatiale (valeur de la boîte englobante) et de leur topologie (nombre d'agrégations). Cependant, l'emplacement spécifique sur le graphe n'est pas indiqué. Puisque le graphe est représenté entièrement dans la vue du graphe (Cf. Figure 3.13(e,f)), nous avons décidé d'afficher les emplacements des résultats dans cette vue. Pour y parvenir, nous devons considérer certains aspects: i) la position des sommets en fonction du niveau de zoom actuel, ii) la position des sommets prédéfinie dans le graphe, et iii) la présence des mêmes sommets dans certains résultats.

Compte tenu de ces aspects, plusieurs méthodes existantes telles que *BubbleSets* [15] (Cf. Figure 3.16a), *LineSets* [2](Cf. Figure 3.16b) ou les *diagrammes Kelp* [23, 45, 51](Cf. Figure 3.16c) semblent appropriées. Toutes ces approches fonctionnent lorsque les positions des sommets sont fixes. Nous avons choisi les diagrammes Kelp en raison de critères esthétiques et de la possibilité de les adapter pour faire face au chevauchement des éléments (voir ci-dessous pourquoi).

Généralement, les diagrammes Kelp sont appliqués à des ensembles de sommets statiques (*e.g.* des cartes) sans interaction avec l'utilisateur. Étant donné que dans notre contexte il existe de nombreuses interactions, comme le zoom qui peut par exemple changer la position des sommets, nous devons adapter les diagrammes Kelp (cette approche est décrite dans la section 3.6).

À partir de la vue des résultats (Cf. Figure 3.13h), l'utilisateur peut sélectionner jusqu'à cinq occurrences qui seront mises en surbrillance dans la vue du graphe en utilisant des diagrammes Kelp (Cf. Figure 3.13i). Lorsque l'utilisateur sélectionne une occurrence dans la liste, la bordure de son cadre est colorée avec l'une des couleurs prédéfinies : rouge, vert, bleu, orange ou rose. La même couleur est utilisée dans le diagramme Kelp associé dans la vue du graphe (Cf. Figure 3.13i).

VERTIGo fournit aussi un histogramme pour explorer les résultats en fonction de la valeur de leur boîte englobante (MBR) *[R2]*. Les valeurs de MBR sont regroupées en n intervalles uniformes définis par l'utilisateur ($1 \leq n \leq 10$).



(a) (b)

(c)

FIGURE 3.16 – Approches pour visualiser un ensemble de sommets : (a) BubbleSets (b) LineSets et (c) Diagrammes Kelp.

La Figure 1.2c illustre un histogramme avec cinq intervalles. Dans cet exemple, nous pouvons souligner qu'il existe une concentration de résultats avec une faible valeur de MBR, ce qui reflète leur proximité dans le graphe (peut-être dans le même cluster). D'un autre côté, les deux derniers intervalles montrent une faible concentration de résultats avec une valeur élevée de MBR (les résultats correspondants peuvent être dans des clusters différents). Cet histogramme permet à l'utilisateur de filtrer les résultats en cliquant sur la barre désirée. Grâce à cette fonctionnalité, les utilisateurs peuvent concentrer leur attention sur un groupe spécifique ou découvrir des résultats aberrants (Cf. section 3.7).

3.6 Une approche basée sur les diagrammes Kelp

Lors d'un zoom dans la vue du graphe, les positions des sommets changent en fonction du point focal (qui correspond au pointeur de la souris). Ainsi, à un certain niveau de zoom, certains sommets peuvent se chevaucher. Malheureusement, même si les diagrammes Kelp sont très attrayants pour mettre en évidence un ensemble de sommets, ils ne sont pas adaptés au traitement de sommets dont les positions sont variables. Aussi, le temps de calcul élevé pénalise son utilisation en temps réel. Pour résoudre ce problème, nous étendons la technique Kelp avec une stratégie de gestion des positions de sommets variables.

Nous considérons chaque résultat comme un *sous-graphe* E du graphe initial. $\mathbb{E}$ désigne l'ensemble des résultats et $\mathbb{N}$ désigne l'ensemble des sommets dans $\mathbb{E}$. Notre approche comporte deux étapes: i) trouver un arrangement spatial des sommets ($\mathbb{N}$) et des résultats ($\mathbb{E}$) au niveau de zoom actuel et ii) générer une visualisation imbriquée des résultats ($\mathbb{E}$).

3.6.1 Positionner les sommets et les résultats

La première étape de notre approche vise à localiser l'ensemble des sommets $\mathbb{N}$ et l'ensemble des résultats $\mathbb{E}$ en évitant le chevauchement lié au niveau de zoom actuel.

Chaque sommet $n \in \mathbb{N}$ est affiché comme un cercle de rayon r_n ($r_n \geq 1$) correspondant au nombre de résultats $E \in \mathbb{E}$ auquel il appartient. En prenant en compte les cercles et les positions des sommets au niveau de zoom actuel, certains sommets peuvent se chevaucher. Pour résoudre ce problème, nous supprimons le chevauchement en utilisant l'approche décrite dans [62]. Cette approche propose de calculer une *triangulation de Delaunay* à partir de la disposition des sommets (Cf. Figure 3.17a), puis d'éliminer les chevauchements progressivement en utilisant un modèle de *stress majorization* [28] (Cf. Figure 3.17b).

Le résultat de cette méthode fournit un graphe sans chevauchement qui permet de conserver la carte mentale de la mise en page initiale. En outre, la méthode conserve également une complexité de calcul raisonnable pour permettre de zoomer/dézoomer en temps réel.

Après avoir supprimé les chevauchements, l'objectif est de relier les sommets de chaque résultat E. Puisque la structure topologique des résultats est déjà montrée dans la vue des résultats (Cf. Figure 3.13h), notre but ici est juste de connecter visuellement les sommets. Pour ce faire, nous appliquons l'algorithme de Dijkstra pour chaque résultat $E \in \mathbb{E}$ afin d'obtenir un arbre recouvrant de longueur minimale $T_E \subseteq E$. La distance euclidienne entre sommets est utilisée comme poids des arêtes. Nous obtenons ainsi un ensemble de liens $\mathbb{L}$ correspondant à l'union des arêtes des

arbres couvrants. Chaque lien $l \in \mathbb{L}$ contient une épaisseur notée r_l ($r_l \geq 1$) qui est égale au nombre d'arbre couvrants auxquels appartient l'arête correspondante. La Figure 3.17c montre ces liens en orange.

Cependant, il arrive que certains liens chevauchent les sommets, voir la boîte rouge sur la Figure 3.17c. Pour résoudre ce problème nous introduisons des *sommets de contrôle fixes* autour du barycentre des régions de chevauchement. Ici, le rayon est donné par la distance du barycentre au lien l. La Figure 3.17d montre deux sommets de contrôle, Cl et C2 sur les régions de chevauchement représentées sur la Figure 3.17c. L'ensemble des *sommets de contrôle fixes* est ajouté à l'ensemble des sommets $\mathbb{N}$ afin d'effectuer à nouveau le processus de suppression des chevauchements décrit ci-dessus. La position des sommets de contrôle reste fixe pendant le processus. Ainsi, les positions des liens sont préservées et seuls les sommets qui se chevauchent sont réorganisés. Une fois le processus de chevauchement de sommets-liens terminé, les sommets de contrôle sont retirés de l'ensemble des sommets $\mathbb{N}$. La Figure 3.17e illustre le résultat final de ce processus, avec la triangulation de Delaunay en gris et le barycentre des sommets de contrôle en rouge. Une fois que nous obtenons un emplacement approprié des sommets et des liens, ces dernières sont dessinées.

3.6.2 Génération d'une visualisation imbriquée

La deuxième étape de notre approche consiste à visualiser l'ensemble des sommets et des arêtes des arbres couvrants. Nous proposons deux styles imbriqués différents : un diagramme incluant les arêtes et un autre sans arêtes (Cf. Figure 3.18).

Pour générer ces visualisations, chaque arbre couvrant T_E est représenté par une couche colorée. Ensuite, chaque T_E est empilé sur les autres en modifiant le rayon des sommets et l'épaisseur des liens. La couleur de T_E correspond à la bordure du résultat E sélectionné dans la vue des résultats (Cf. Figure 3.13h).

La Figure 3.18 illustre les deux visualisations imbriquées proposées. Dans la Figure 3.18a, les arêtes sont visibles tandis que dans la Figure 3.18b, les arêtes sont masquées. Dans ces exemples, les diagrammes montrent quatre résultats sélectionnées où *Y. Song* et *S. Liu* apparaissent (Cf. Figure 3.13h). Nous avons étiqueté les sommets afin de faciliter leur identification. Nous pouvons remarquer que lorsque les sommets sont impliqués dans de nombreux résultats, ils deviennent visuellement dominants (*e.g.* les sommets représentant *Y. Song* et *S. Liu*).

VERTIGo fournit également à l'utilisateur un panneau de contrôle permettant de passer d'un style à l'autre à la volée, ou pour modifier certains paramètres visuels tels que le rayon initial des sommets, l'épaisseur des arêtes, la taille de la police de l'étiquette, etc.

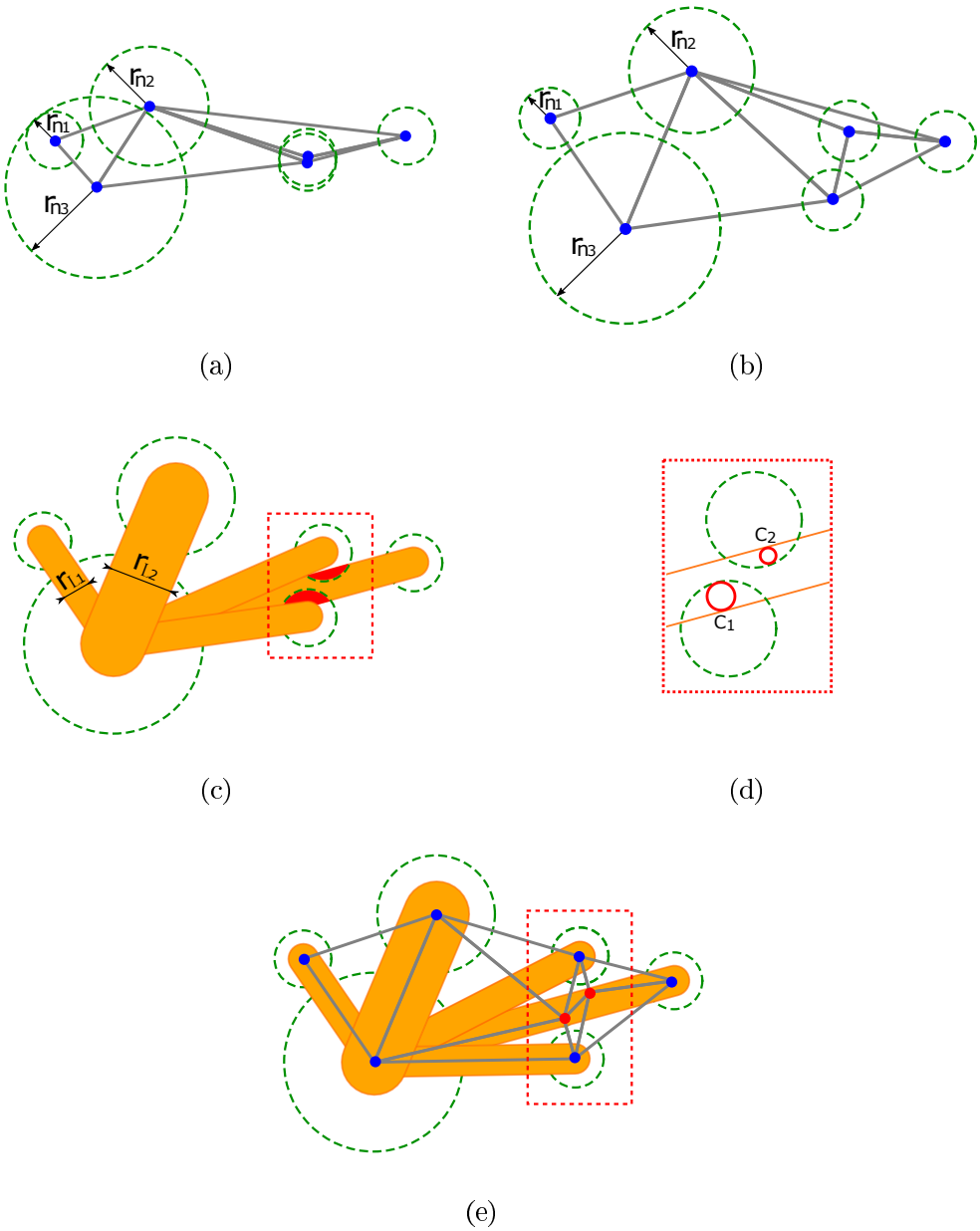
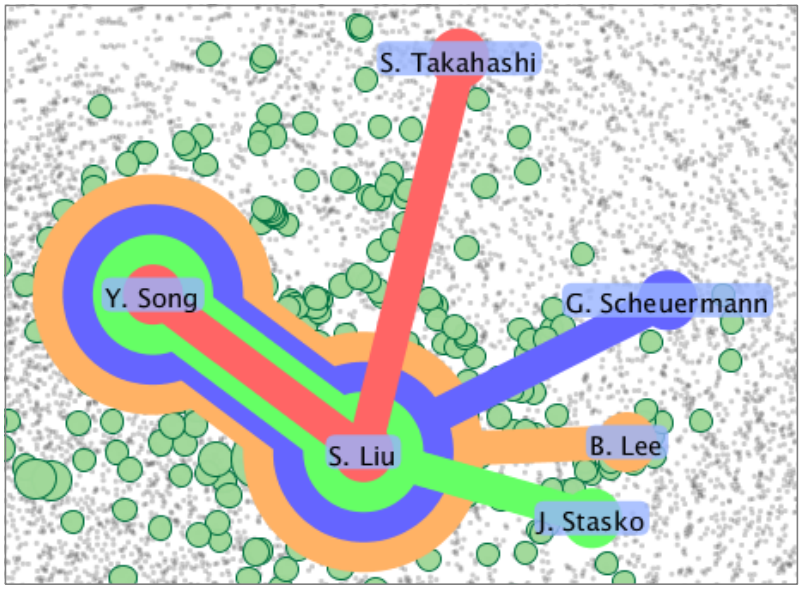
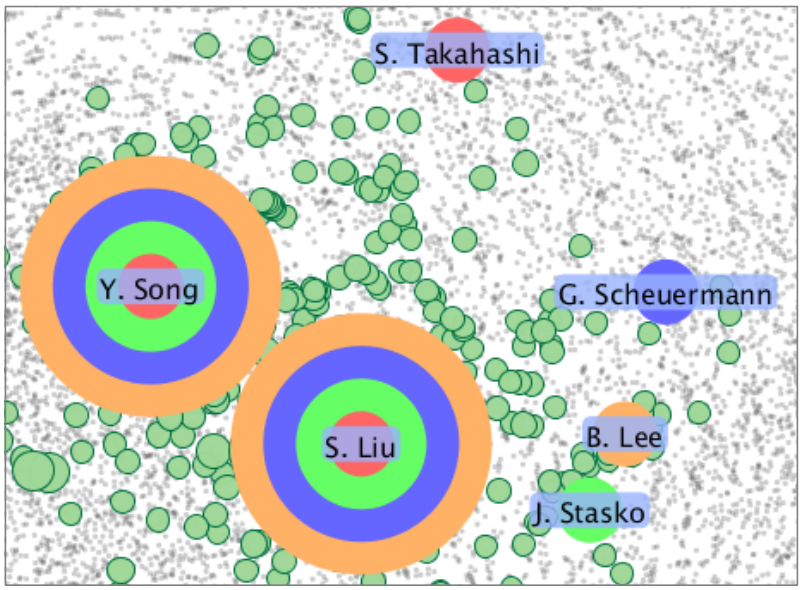


FIGURE 3.17 – Processus de suppression de chevauchements de sommets-liens : (a) Détection des chevauchements de sommets (b) Suppression des chevauchements des sommets en utilisant une approche basée sur une triangulation de Delaunay. (c) Ajout d'arêtes entre les sommets (orange) et détection de chevauchements possibles de ces arêtes avec les sommets (boîte rouge) (d) Détermination de la position des sommets de contrôle fixes (en rouge) (e) Les sommets de contrôle fixes sont utilisés pour recalculer la position des sommets de façon à supprimer les chevauchements sommets-liens.



(a)



(b)

FIGURE 3.18 – Styles imbriqués basés sur les diagrammes Kelp: (a) les arêtes sont colorées et visibles (b) les arêtes ne sont pas affichées.

3.6.3 Mise en œuvre et équipement

VERTIGo est développé en utilisant les technologies Java (JavaFX, Processing [5]) pour l'interface et l'interaction entre les différents composants visuels. Nous utilisons l'implémentation de *Multipole Multilevel Method* (FM^3) fournie par l'Open Graph Drawing Framework (OGDF) [6] pour le graphe. Le moteur de recherche SuMGra [37] est également implémenté en Java et intégré au système. Le scénario de démonstration est réalisé sur un poste de travail équipé d'un processeur Intel Xeon à 3,00 GHz avec 16 Go de RAM et un système d'exploitation Windows 64 bits.

3.7 Exemple

Cette section présente un exemple d'utilisation de VERTIGo. Nous considérons comme jeu de données un réseau de co-auteurs obtenu à partir de la base de données bibliographiques *DBLP* [7]. Le réseau est composé de 37 878 sommets (les auteurs) et 129 983 arêtes (relations entre les auteurs). Le réseau comporte 18 couches (types d'arêtes) correspondant chacune à une conférence ou un journal dans lequel les auteurs ont publié. Deux sommets sont reliés par une arête de type X si les auteurs correspondants ont co-écrit au moins un article dans la conférence ou le journal X. Le graphe contient des chercheurs issus de deux domaines de recherche : la Visualization (VIS) et le Data Mining/Data Base (DMDB). En ce qui concerne le domaine de la VIS, nous avons sélectionné les articles parus dans : *TVCG*, *InfoVis*, *CGF*, *EuroVis*, *PacificVis*, *Graph Drawing*, *CG&A*, et *IV*. En ce qui concerne le domaine de DMDB, nous avons sélectionné les articles parus dans : *DASFAA*, *EDBT*, *ICDE*, *ICDM*, *ICDT*, *KDD*, *SDM*, *SSDBM*, *SIGMOD*, et *VLDB*. Chaque conférence ou journal correspond à une couche et est représenté par une couleur différente dans le graphe multicouches.

La Figure 3.19 montre le graphe affiché dans la vue du graphe. Compte tenu de la structure du graphe, les sommets (auteurs) sont affichés comme des cercles noirs avec des tailles identiques et les arêtes ne sont pas représentées pour éviter d'encombrement la visualisation. Nous pouvons noter que la disposition du graphe issue de l'algorithme de force permet de reconnaître visuellement deux zones denses. Ce résultat était attendu car le jeu de données correspond à deux communautés différentes.

5. Processing Website: https://processing.org/
6. OGDF Website: http://www.ogdf.net/
7. DBLP Website: http://dblp.uni-trier.de/

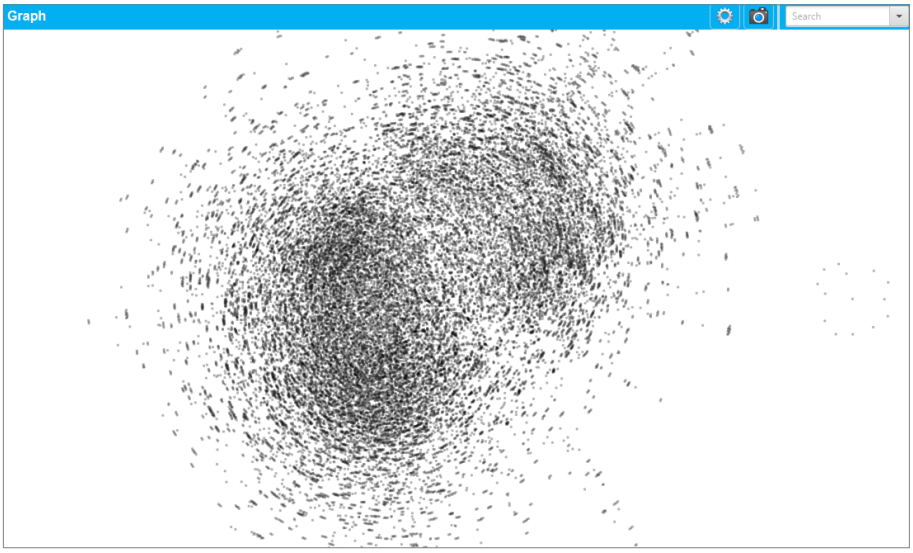


FIGURE 3.19 – Visualisation du jeu de données DBLP dans la vue du graphe. Nous pouvons constater deux zones denses dans la disposition des sommets du graphe correspondant aux deux communautés de recherche représentées.

Dans cet exemple, nous cherchons à récupérer le *réseau de collaborateurs* d'un auteur donné en utilisant les k arêtes suggérées fournies par le mécanisme de suggestion de requête. Le haut de la Figure 3.20a décrit la requête initiale. La requête contient deux auteurs (sommets) qui ont co-écrit un papier dans *TVCG*. L'un des deux sommets est identifié comme étant *Kwan-Liu Ma*. Une fois la requête exécutée, la vue du graphe affiche les résultats récupérés via une carte de chaleur afin de les localiser sur le graphe d'origine (voir le bas de la Figure 3.20a). La carte montre une concentration de résultats dans la partie centrale droite du réseau, ce qui nous amène à penser que la zone dense sur la droite correspond à la communauté VIS.

Dans le but d'analyser visuellement le réseau de collaboration des co-auteurs de *Kwan-Liu Ma*, nous utilisons le mécanisme de suggestion de requêtes fourni par VERTIGO. Le haut de la Figure 3.20b affiche des suggestions d'arêtes pour la requête. En concentrant l'attention sur la suggestion d'arêtes externes (*i.e.* les camemberts) du co-auteur, le pourcentage affiché sur les tranches nous informe que 50% des résultats supportent l'extension obtenue via un nouveau lien *TVCG* entre un co-auteur de *Kwan-Liu Ma* et un autre auteur. Ainsi, en ajoutant l'arête suggérée (*TVCG*) à la requête et en relançant à nouveau le moteur, nous obtenons de nouveaux résultats. Le bas de la Figure 3.20b illustre les résultats de cette nouvelle requête. Nous pouvons facilement noter que la concentration des résultats (et aussi le volume) augmente par rapport à celui de la requête précédente. Ceci est souligné par la nouvelle carte de chaleur générée (le bas de la Figure 3.20b). De plus, ces résultats sont concentrés sur la partie droite du réseau DBLP. Ceci confirme que cette zone du graphe correspond bien à la communauté VIS. Une fois cette communauté

détectée dans le réseau, il est possible d'ajouter des arêtes suggérées afin d'observer si d'autres connexions font le lien avec des auteurs de la communauté DMDB (zone dense à gauche du réseau DBLP).

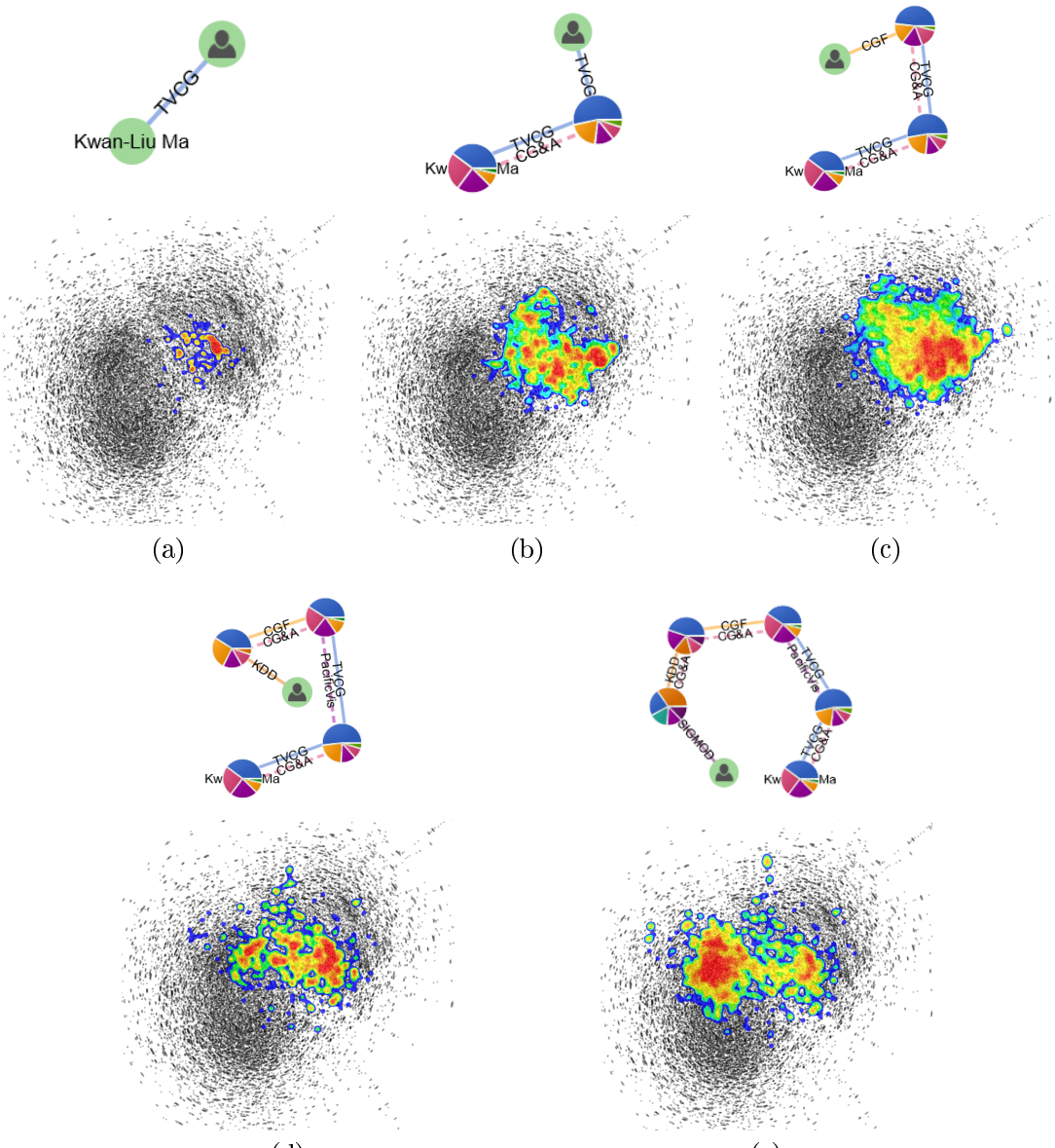


FIGURE 3.20 – Processus d'affinement de requête grâce aux suggestions fournies par VERTIGO

.

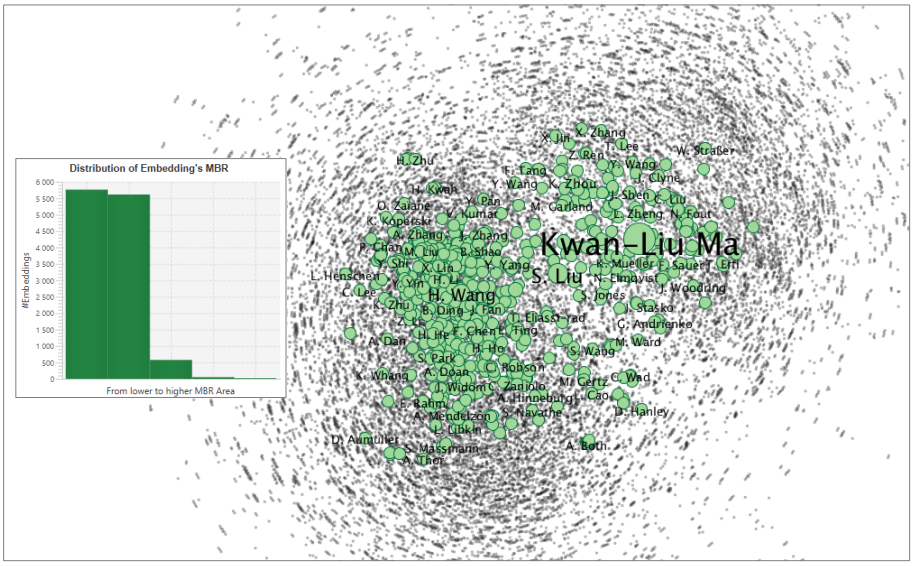
Le haut de la Figure 3.20c montre la requête après l'ajout d'une arête *CGF* vers un nouvel auteur. Les résultats pour cette nouvelle requête montrent encore plus de concentration des résultats dans la communauté VIS. Ceci est dû au fait que

jusqu'à présent la requête a été affinée en ne considérant que des journaux (types d'arêtes) qui n'appartiennent qu'au domaine VIS (*i.e.* des liens *TVCG* et *CGF*). Cependant, le sommet en haut à gauche de la Figure 3.20d montre une suggestion d'arêtes correspondant à la conférence *KDD*. Le bas de la Figure 3.20d affiche les résultats lors de l'ajout de cette arête. Il est possible d'observer que, cette fois, les résultats commencent à se propager vers la communauté DMDB (partie gauche du réseau DBLP).

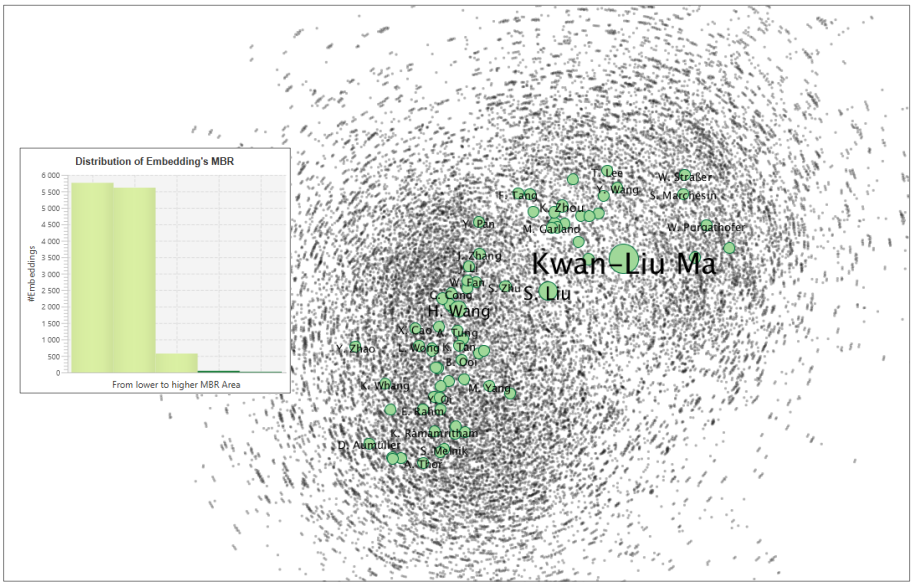
Après trois étapes de raffinement, nous pouvons déduire que le réseau de collaborations de co-auteurs de *Kwan-Liu Ma* contient principalement des chercheurs de la communauté VIS mais, à partir de la requête actuelle (haut de la Figure 3.20d), nous notons qu'il n'y a que *4 degrés de séparation* (*TVCG-TVCG-CGF-KDD*) de *Kwan-Liu Ma* à un auteur du domaine DMDB (article *KDD*). Sur la Figure 3.20e, un arête *SIGMOD* est ajoutée à la requête pour considérer des auteurs appartenant à la communauté DMDB. La carte de chaleur illustrée sur la Figure 3.20e montre que la requête affinée récupère des résultats qui contiennent clairement des auteurs appartenant aux deux communautés. Ceci est soutenu par le fait que, en observant la carte de chaleur, la concentration des résultats s'est décalée du côté droit vers le côté gauche du réseau.

Guidés par la carte de chaleur de la Figure 3.20e, nous nous concentrons sur le centre des deux communautés. La Figure 3.21a présente en détail les sommets des résultats. La taille de la police des étiquettes permet de localiser facilement les auteurs en fonction de leur nom. Par exemple, la taille de la police de *Kwan-Liu Ma* est plus grande que celle des autres noms d'auteurs puisque cet auteur apparaît dans tous les résultats de la requête.

Nous pouvons constater qu'il est situé au centre de la communauté VIS. Dans l'histogramme qui affiche les résultats repartis sur 5 intervalles, nous pouvons remarquer qu'il y a beaucoup de résultats avec une faible valeur de MBR, ce qui signifie une forte collaboration entre les auteurs de la même communauté. Sur la Figure 3.21b, en filtrant les trois premiers intervalles de l'histogramme afin de se concentrer sur les résultats avec une valeur élevée de MBR (auteurs qui peuvent être dans des communautés différentes), la vue du graphe montre moins de résultats, ce qui indique une moindre collaboration entre les auteurs des deux communautés. Une partie de la liste des résultats sur le côté droit de la Figure 3.22 correspond à la sélection de *Kwan-Liu Ma* afin de voir l'ensemble des résultats auxquels il appartient.



(a)



(b)

FIGURE 3.21 – (a) L'histogramme montre qu'un grand nombre de résultats ont une faible valeur de MBR. (b) Résultats après filtrage des valeurs faibles de MBR.

La Figure 3.22 illustre les diagrammes Kelp après avoir sélectionné quatre résultats. Chaque résultat sélectionné est encadré par une couleur différente dans la liste, la même couleur est utilisée pour représenter le diagramme Kelp sur le graphe DBLP. Cette image met en évidence que l'auteur *N. Magnenat-Thalmann* constitue une sorte de pont entre les deux communautés depuis qu'elle a publié des articles dans des conférences DBDM (*KDD*) et dans des revues du VIS (*CGF*).

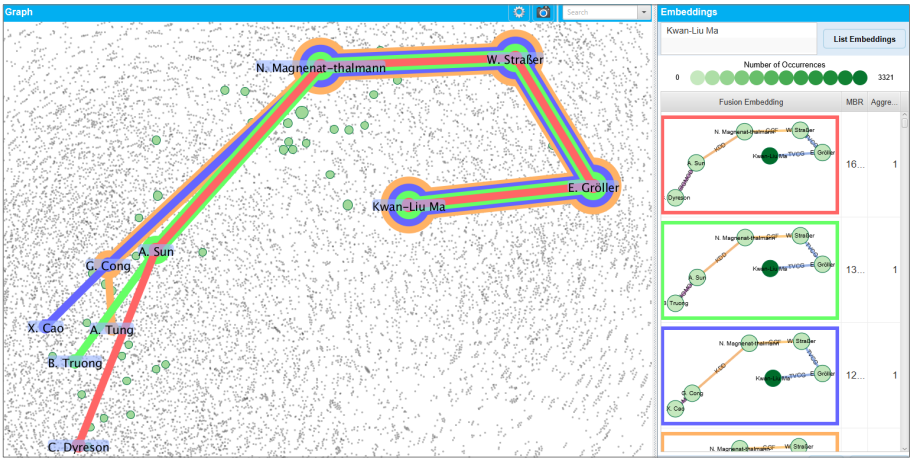


FIGURE 3.22 – Les diagrammes Kelp associés aux résultats sélectionnés dans la liste de droite.

3.8 Conclusion

Dans ce chapitre, nous avons présenté VERTIGO, une nouvelle plateforme visuelle permettant de requêter des grands graphes multicouches, de visualiser/explorer les résultat obtenus et de suggérer de nouvelles extensions des requêtes basées sur la structure du graphe et les résultats de la requête actuelle.

Les résultats peuvent être analysés à différents niveaux de détail et les composants visuels conçus aident l'utilisateur à se concentrer sur des zones particulièrement intéressantes du graphe pour une inspection plus en détail. Le système permet également à l'utilisateur d'interagir avec le moteur de recherche, ainsi, l'utilisateur peut démarrer, mettre en pause et reprendre la recherche de résultats avec la possibilité de naviguer/explorer les résultats partiellement collectés.

Un exemple montre que VERTIGO peut être utilisé pour inspecter et explorer un réseau de co-auteurs modélisé à l'aide d'un graphe multicouches. Les composants visuels et les techniques d'interaction permettent à l'utilisateur de gérer les tâches couramment impliquées dans le processus de requêtage visuel : construction de requêtes, visualisation des résultats, exploration des résultats et suggestion de requêtes.

CHAPITRE

4 Conclusions générales et perspectives

Sommaire

4.1 Résumé des contributions 90
4.1.1 Visualisation de séries temporelles hiérarchiques 90
4.1.2 Visualisation de graphes multicouches 90
4.2 Perspectives 91
4.2.1 Agrégation temporelle des séries 91
4.2.2 Fouille visuelle de graphes 92
4.2.3 Traitement de graphes dynamiques 92

4.1 Résumé des contributions

Dans ce mémoire, nous avons présenté deux nouvelles approches visuelles pour traiter des problématiques liées à l'analyse de données dynamiques et complexes : MULTISTREAM [19, 20, 50] permet de traiter des séries temporelles hiérarchiques et VERTIGO [17,18] permet de traiter des données complexes modélisées sous la forme de graphes multicouches.

4.1.1 Visualisation de séries temporelles hiérarchiques

Dans le chapitre 2, nous avons présenté l'outil MULTISTREAM pour traiter de longues séries temporelles multiples. Cette approche permet la visualisation, l'exploration et la comparaison de l'évolution de séries organisées dans une structure hiérarchique. Elle est basée sur quatre vues interactives : une *vue globale* montre les principales tendances du jeu de données; une *vue multirésolution* montre les détails sur un sous-ensemble du jeu de données; un *contrôleur* permet de sélectionner/filtrer des périodes spécifiques et un *gestionnaire de hiérarchie* permet de naviguer/explorer la structure hiérarchique.

Les interactions mises en place aident à sélectionner un segment temporel et à naviguer dans la hiérarchie de ce segment. En conséquence, la vue multirésolution représente la hiérarchie à différents niveaux de détails dans une période de temps sélectionnée. Enfin deux jeux de données, l'un portant sur les émotions exprimées dans des médias sociaux, l'autre sur l'évolution des genres musicaux, montrent l'utilité de MULTISTREAM pour mettre en évidence des pics saisonniers et des tendances au fil du temps. Récemment l'approche a également été mise en œuvre pour analyser et caractériser des échanges entre utilisateurs dans des forums de santé [50].

4.1.2 Visualisation de graphes multicouches

Dans le chapitre 3 nous avons présenté VERTIGO, un système permettant le requêtage visuel de grands graphes multicouches. Il permet de visualiser et d'explorer les résultats obtenus à différents niveaux de détails. De plus, Il permet aussi de suggérer de nouvelles extensions de requêtes basées sur la structure du graphe et les résultats de la requête courante. VERTIGO est basé sur trois vues principales : une *vue de la requête* qui permet la construction et la suggestion interactive de requêtes; une *vue du graphe* qui affiche le graphe et permet d'explorer les résultats à différents niveaux de granularité; une *vue des résultats* qui montre la liste de résultats pour d'autres analyses (*e.g.* spatiales ou topologiques).

Ces trois vues et les modes d'interactions proposés par VERTIGo aident l'utilisateur à se concentrer sur des zones particulièrement intéressantes du graphe et ainsi à les analyser plus en détail. VERTIGo interagit aussi avec le moteur de recherche pour permettre à l'utilisateur de démarrer, mettre en pause et reprendre la recherche de résultats avec la possibilité d'explorer des résultats partiellement collectés. Enfin, un réseau de co-auteurs modélisé comme un graphe multicouches montre l'utilité de cette approche pour gérer les tâches couramment impliquées dans le processus de requêtage visuel : construction de requêtes, visualisation des résultats, exploration des résultats et suggestion de requêtes.

4.2 Perspectives

Les travaux présentés dans ce mémoire offrent de nombreuses perspectives et nous en présentons quelques-unes dans les sections suivantes.

4.2.1 Agrégation temporelle des séries

Comme nous l'avons vu, les différentes vues et interactions proposées dans MultiStream permettent d'analyser de longues séries temporelles. Nous revenons, à présent, sur le fait que les échelles de temps peuvent être très grandes ou que les données peuvent être disponibles en temps réel, *i.e.* des flux de données (*e.g.* RSS, emails, etc.). Dans ce cas, les approches issues des traitements de flots de données pourraient être utilisées. Par exemple, les titled time windows, introduites dans [30], sont basées sur le fait que les personnes sont souvent plus intéressées par les changements récents avec une granularité fine (*e.g.* connaître l'ensemble des émotions exprimées dans des tweets la dernière heure) et à long terme à une granularité moins précise (*e.g.* savoir que l'opinion est globalement positive ou négative les mois précédents). Ce mécanisme agrège progressivement, en fonction de périodes spécifiques (*e.g.* heure, jour, mois, etc.), les données en utilisant la hiérarchie disponible. Une telle approche semble tout à fait appropriée pour étendre MultiStream aux flux de données : la hiérarchie pourrait être couplée, par exemple, à une métaphore visuelle basée sur le processus physique de sédimentation [36]. La vue multirésolution actuelle, qui est basée sur une approche *focus+contexte*, semble aussi adaptée en montrant les segments des séries qui apparaissent dans la zone de contexte avec une agrégation temporelle plus élevée que les segments qui apparaissent dans la zone du focus. Elle soulève toutefois la problématique de la récupération des données lors du focus sachant que ces dernières ne sont peut-être pas conservées.

4.2.2 Fouille visuelle de graphes

L'objectif de VERTIGo est de visualiser les résultats à partir d'une requête exprimée sous la forme d'un graphe. En outre, la visualisation est fortement interconnectée avec le moteur de recherche et permet à l'utilisateur de pouvoir lancer, mettre en pause ou analyser des résultats partiels. L'une des perspectives est de poursuivre ces travaux en considérant non plus la recherche de sous-graphes mais l'extraction de sous-graphes fréquents. De nombreuses approches existent (*e.g.* [38, 42]) et il serait intéressant de proposer un outil visuel pour les piloter. De manière à mieux intégrer l'utilisateur dans le processus d'extraction, de nouvelles fonctionnalités visuelles pourraient être proposées pour permettre de sélectionner des zones de données vers lesquelles l'utilisateur souhaite se focaliser. Par exemple, dans le cas du graphe DBLP (voir chapitre 3), l'utilisateur pourrait vouloir extraire uniquement les graphes pour une seule communauté. D'autres problèmes doivent également être considérés et sont plus liés aux algorithmes d'extraction : comment appréhender la sélection de l'utilisateur dans l'espace de recherche ? Quid des paramètres d'extraction ? Par exemple le support minimal utilisé couramment dans les approches doit-il être adapté à la partie sélectionnée ou encore à l'ensemble des données ? De manière complémentaire, VERTIGo pourrait proposer une nouvelle vue pour interagir avec l'espace de recherche. Elle pourrait mettre en évidence et éventuellement proposer des mécanismes de filtrage d'une région de l'espace de recherche.

4.2.3 Traitement de graphes dynamiques

VERTIGo permet de traiter des graphes statiques. Il existe cependant de nombreuses applications où les graphes sont dynamiques. Alors que les réseaux sociaux induisent, par exemple, des opérations d'ajout et de suppression; d'autres réseaux, comme le graphe des auteurs, n'évoluent que par ajout. L'une des perspectives serait de s'intéresser à ces graphes dynamiques afin de mieux comprendre leur évolution. Ce problème soulève de très nombreuses questions comme : quelle représentation visuelle est la plus adaptée pour représenter le graphe au cours du temps ? Quel type de requête l'utilisateur peut-il formuler ?

Pour visualiser la temporalité du graphe, une vue multirésolution comme celle de MultiStream pourrait être adaptée. La zone de focus peut être utilisée pour afficher les graphes à différents pas de temps. Par contre la représentation des graphes dans les zones de transition et de contexte soulève de nouveaux problèmes. Comment représenter des graphes agrégés dans lesquels les différentes évolutions apparaissent ? Dans le cas de l'apparition ou de la disparition de nœuds ou d'arcs, une solution pourrait être de considérer un support (*e.g.* un nœud doit apparaître à minima x fois dans une fenêtre de temps pour être retenu). Cette solution toutefois ne prend pas en compte réellement les évolutions. Par exemple l'utilisateur pourrait être intéressé

par une représentation mettant en évidence qu'un nœud est utilisé de plus en plus souvent ou a disparu pendant un laps de temps et, a, de nouveau, été inséré. De la même manière, si on ne considère que la possibilité d'ajout, l'utilisateur peut être intéressé par l'évolution et non pas simplement par la valeur finale. Des travaux issues des traitements de flots de données ont proposé des modélisations pour représenter des évolutions et il pourrait être intéressant de les évaluer dans ce contexte.

En ce qui concerne la requête, il existe de nombreuses requêtes qui peuvent être formulées en prenant en compte les évolutions. Par exemple, « *Parmi les auteurs qui ont publié à TVCG et KDD, donner l'ordre d'apparition des auteurs* » ou « *Quels sont les auteurs qui ayant publié dans TVCG et KDD ont publié de plus en plus, au cours du temps, dans TVCG ?* ». Il convient bien entendu de faire une première analyse du type de requête que l'on souhaite aborder pour étudier les modes de représentation à mettre en place. Le second problème est lié au moteur de recherche. En effet, il n'existe pas, à notre connaissance, de moteurs permettant ce type d'interrogations sur les graphes.

Bibliographie

[1] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Springer, 2011. (Cité à la page 12.)

[2] Basak Alper, Nathalie Riche, Gonzalo Ramos, and Mary Czerwinski. Design study of linesets, a novel set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, 2011. (Cité à la page 77.)

[3] Dominikus Baur, Bongshin Lee, and Sheelagh Carpendale. TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs. In *Proceedings of the International Conference on Interactive Tabletops and Surfaces (ITS)*, pages 255–264. ACM, 2012. (Cité aux pages xii, 15, 16, 19, 20, 43 et 44.)

[4] Richard A Becker and William S Cleveland. Brushing Scatterplots. *Technometrics*, 29(2):127–142, 1987. (Cité aux pages 16 et 26.)

[5] Lior Berry and Tamara Munzner. Binx: Dynamic Exploration of Time Series Datasets Across Aggregation Levels. In *Proceedings of the Symposium on Information Visualization (InfoVis)*. IEEE, 2004. (Cité à la page 8.)

[6] Sourav S Bhowmick, Byron Choi, and Shuigeng Zhou. VOGUE: Towards A Visual Interaction-aware Graph Query Processing Framework. In *Proceedings of the Biennial Conference on Innovative Data Systems Research (CIDR)*, 2013. (Cité aux pages xv, 56, 57, 59, 60 et 61.)

[7] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen. Distance oracles in edge-labeled graphs. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 547–558. ACM, 2014. (Cité à la page 56.)

[8] Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Antti Ukkonen. Distance oracles in edge-labeled graphs. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 547–558. ACM, 2014. (Cité à la page 69.)

[9] Romain Bourqui, Dino Ienco, Arnaud Sallaberry, and Pascal Poncelet. Multilayer Graph Edge Bundling. In *Proceedings of the Pacific Visualization Symposium (PacificVis)*, pages 184–188. IEEE, 2016. (Cité à la page 56.)

[10] L. Byron and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008. (Cité aux pages xii, 9, 13, 14 et 25.)

[11] V. Carletti, P. Foggia, A. Saggese, and M. Vento. Introducing VF3: A New Algorithm for Subgraph Isomorphism. In *Proceedings of the International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR)*, pages 128–139. Springer, 2017. (Cité aux pages 57 et 66.)

[12] Marianne Sheelagh Therese Carpendale. *A Framework for Elastic Presentation Space*. Simon Fraser University, 1999. (Cité à la page 18.)

[13] Duen Horng Chau, Christos Faloutsos, Hanghang Tong, Jason I Hong, Brian Gallagher, and Tina Eliassi-Rad. Graphite: A visual query system for large graphs. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 963–966. IEEE, 2008. (Cité aux pages xv, 56, 57, 58 et 61.)

[14] Andy Cockburn, Amy K. Karlson, and Benjamin B. Bederson. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1):2:1–2:31, 2009. (Cité aux pages 11 et 16.)

[15] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009. (Cité à la page 77.)

[16] Luigi Pietro Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. An improved algorithm for matching large graphs. In *Proceedings of the International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR)*, pages 149–159. Springer, 2001. (Cité aux pages 57 et 59.)

[17] Erick Cuenca, Arnaud Sallaberry, Dino Ienco, and Pascal Poncelet. Visual Querying of Large Multilayer Graphs. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 32–34. ACM, 2018. (Cité aux pages 3, 57 et 90.)

[18] Erick Cuenca, Arnaud Sallaberry, Dino Ienco, and Pascal Poncelet. Visual Querying and Exploring of Large Multilayer Graphs. In *Poster session of the IEEE Information Visualization Conference (InfoVis)*. IEEE, (to present), 2018. (Cité aux pages 3, 57 et 90.)

[19] Erick Cuenca, Arnaud Sallaberry, Florence Ying Wang, and Pascal Poncelet. Visualizing Hierarchical Time Series with a Focus+Context Approach. In *Poster session of the IEEE Information Visualization Conference (InfoVis)*. IEEE, 2017. (Cité aux pages 3, 11 et 90.)

[20] Erick Cuenca, Arnaud Sallaberry, Florence Ying Wang, and Pascal Poncelet. MultiStream: A Multiresolution Streamgraph Approach to Explore Hierarchical Time Series. *IEEE Transactions on Visualization and Computer Graphics*, (to appear), 2018. (Cité aux pages 3, 11 et 90.)

[21] Weiwei Cui, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai Gao, Huamin Qu, and Xin Tong. TextFlow: Towards Better Understanding of Evolving Topics in Text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011. (Cité à la page 15.)

[22] Weiwei Cui, Shixia Liu, Zhuofeng Wu, and Hao Wei. How Hierarchical Topics Evolve in Large Text Corpora. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2281–2290, 2014. (Cité aux pages xii, 15 et 18.)

[23] Kasper Dinkla, Marc J Van Kreveld, Bettina Speckmann, and Michel A Westenberg. Kelp Diagrams: Point Set Membership Visualization. *Computer Graphics Forum*, 31(3pt1):875–884, 2012. (Cité à la page 77.)

[24] Wenwen Dou, Li Yu, Xiaoyu Wang, Zhiqiang Ma, and William Ribarsky. HierarchicalTopics: Visually Exploring Large Text Collections Using Topic Hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2002–2011, 2013. (Cité aux pages xii, 15, 17, 19, 20, 43 et 44.)

[25] L Feldman Barrett and James A Russell. Independence and Bipolarity in the Structure of Current Affect. *Journal of Personality and Social Psychology*, 74(4):967–984, 1998. (Cité aux pages 21 et 45.)

[26] Ying-Huey Fua, Matthew O Ward, and Elke A Rundensteiner. Navigating Hierarchies with Structure-Based Brushes. In *Proceedings of the Symposium on Information Visualization (InfoVis)*, pages 58–64. IEEE, 1999. (Cité à la page 19.)

[27] G. W. Furnas. Generalized Fisheye Views. In *Proceedings of the Conference on Human Factors in Computing System (CHI)*, pages 16–23. ACM, 1986. (Cité à la page 16.)

[28] Emden R Gansner and Yifan Hu. Efficient node overlap removal using a proximity stress model. In *Proceedings of the International Symposium in Graph Drawing (GD)*, pages 206–217. Springer, 2008. (Cité à la page 79.)

[29] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky. NewsLab: Exploratory Broadcast News Video Analysis. In *Proceedings of the Symposium on Visual Analytics Science and Technology (VAST)*, pages 123–130. IEEE, 2007. (Cité aux pages xii, 15, 17, 19, 20, 43 et 44.)

[30] Chris Giannella, Jiawei Han, Jian Pei, Xifeng Yan, and Philip S Yu. Mining frequent patterns in data streams at multiple time granularities. *Next generation data mining*, pages 191–212, 2003. (Cité à la page 91.)

[31] S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proceedings of the International Symposium in Graph Drawing (GD)*, pages 285–295. Springer, 2004. (Cité à la page 69.)

[32] W.-S. Han, J. Lee, and J.-H. Lee. $\text{Turbo}_{\text{iso}}$: Towards ultrafast and robust subgraph isomorphism search in large graph databases. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 337–348. ACM, 2013. (Cité aux pages 57 et 66.)

[33] R. L. Harris. *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, 1999. (Cité aux pages 8, 9 et 13.)

[34] S. Havre, E. Hetzler, and L. Nowell. ThemeRiver: Visualizing Theme Changes over Time. In *Proceedings of the Symposium on Information Visualization (InfoVis)*, pages 115–123. IEEE, 2000. (Cité aux pages xii, 13 et 14.)

[35] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002. (Cité aux pages xii, 13 et 14.)

[36] Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. Visual Sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013. (Cité aux pages 13 et 91.)

[37] Vijay Ingalalli, Dino Ienco, and Pascal Poncelet. SuMGra: Querying Multigraphs via Efficient Indexing. In *Proceedings of the International Conference on Database and Expert Systems Applications (DEXA)*, pages 1–15. Springer, 2016. (Cité aux pages 66 et 83.)

[38] Vijay Ingalalli, Dino Ienco, and Pascal Poncelet. Mining Frequent Subgraphs in Multigraphs. *Information Sciences*, 451–452:50–66, 2018. (Cité à la page 92.)

[39] Roberto Interdonato, Andrea Tagarelli, Dino Ienco, Arnaud Sallaberry, and Pascal Poncelet. Local community detection in multilayer networks. *Data Mining and Knowledge Discovery*, 31(5):1444–1479, 2017. (Cité à la page 56.)

[40] Nandish Jayaram, Sidharth Goyal, and Chengkai Li. VIIQ: Auto-Suggestion Enabled Visual Interface for Interactive Graph Query Formulation. *Very Large Data Base*, 8(12):1940–1943, 2015. (Cité aux pages xv, 56, 57, 60, 61, 62 et 67.)

[41] M. A. Jenkins. Algorithm 493: Zeros of a Real Polynomial [c2]. *ACM Transactions on Mathematical Software*, 1(2):178–189, 1975. (Cité à la page 38.)

[42] Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(1):75–105, 2013. (Cité à la page 92.)

[43] M. Kaufmann and D. Wagner. *Drawing graphs: Methods and models*. Springer, 2003. (Cité à la page 69.)

[44] Daniel A Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002. (Cité aux pages 11 et 16.)

[45] Antoine Lambert, François Queyroi, and Romain Bourqui. Visualizing patterns in Node-link Diagrams. In *Proceedings of the International Conference Information Visualization (IV)*, pages 48–53. IEEE, 2012. (Cité à la page 77.)

[46] L. Libkin, J. Reutter, and D. Vrgoč. Trial for rdf: Adapting graph query languages for rdf data. In *Proceedings of the International Symposium on Principles of Database Systems (PODS)*, pages 201–212. ACM, 2013. (Cité à la page 56.)

[47] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. TIARA: Interactive, Topic-Based Visual Text Summarization and Analysis. *ACM Transactions on Intelligent Systems and Technology*, 3(2):25:1–25:28, 2012. (Cité aux pages xii, 13 et 20.)

[48] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of the Conference on Human Factors in Computing System (CHI)*, pages 173–179. ACM, 1991. (Cité à la page 18.)

[49] John D Manning, Beatriz E Marciano, and James J Cimino. Visualizing the Data Using Lifelines2 to Gain Insights from Data Drawn from a Clinical Data Repository. *AMIA Summits on Translational Science Proceedings*, 2013:168, 2013. (Cité à la page 8.)

[50] Yves Mercadier, Jérôme Azé, Sandra Bringay, Viviane Clavier, Erick Cuenca, Céline Paganelli, Pascal Poncelet, and Arnaud Sallaberry. #AIDS Analyse Information Dangers Sexualité : caractériser les discours à propos du VIH dans les médias sociaux. In *Actes des 29es Journées francophones d'Ingénierie des Connaissances (IC)*, 2018. (Cité aux pages 3, 11 et 90.)

[51] Wouter Meulemans, Nathalie Henry Riche, Bettina Speckmann, Basak Alper, and Tim Dwyer. KelpFusion: A hybrid set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1846–1858, 2013. (Cité à la page 77.)

[52] Davide Mottin, Francesco Bonchi, and Francesco Gullo. Graph query reformulation with diversity. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 825–834. ACM, 2015. (Cité à la page 60.)

[53] Kajal Nusratullah, Shoab Ahmad Khan, Asadullah Shah, and Wasi Haider Butt. Detecting Changes in Context using Time Series Analysis of Social Network. In *Proceedings of the Conference on Intelligent Systems (SAI IntelliSys)*, pages 996–1001. IEEE, 2015. (Cité à la page 8.)

[54] Marcus Paradies, Wolfgang Lehner, and Christof Bornhövd. GRAPHITE: An extensible graph traversal framework for relational database management systems. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)*, page 29. ACM, 2015. (Cité à la page 58.)

[55] Marcus Paradies, Michael Rudolf, and Wolfgang Lehner. GraphVista: Interactive Exploration of Large Graphs. *Cornell University Library*, 2015. (Cité aux pages xv, 56, 57, 58, 59, 61 et 66.)

[56] A. Perrot, R. Bourqui, N. Hanusse, F. Lalanne, and D. Auber. Large interactive visualization of density functions on big data infrastructure. In *Proceedings of the International Conference on Large Data Analysis and Visualization (LDAV)*, pages 99–106. IEEE, 2015. (Cité à la page 74.)

[57] Robert Pienta, Fred Hohman, Alex Endert, Acar Tamersoy, Kevin Roundy, Chris Gates, Shamkant Navathe, and Duen Horng Chau. VIGOR: Interactive Visual Exploration of Graph Query Results. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):215–225, 2018. (Cité aux pages xv, 56, 57, 58, 59, 61, 62 et 70.)

[58] Robert Pienta, Fred Hohman, Acar Tamersoy, Alex Endert, Shamkant Navathe, Hanghang Tong, and Duen Horng Chau. VISAGE: Interactive Visual Graph Querying. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI)*, pages 272–279. ACM, 2016. (Cité aux pages xv, 56, 57, 58 et 61.)

[59] Oky Purwantiningsih, Arnaud Sallaberry, Sebastien Andary, Antoine Seilles, et al. Visual Analysis of Body Movement in Serious Games for Healthcare. In *Proceedings of the Pacific Visualization Symposium (PacificVis)*, pages 229–233. IEEE, 2016. (Cité à la page 13.)

[60] Denis Redondo, Arnaud Sallaberry, Dino Ienco, Faraz Zaidi, and Pascal Poncelet. Layer-Centered Approach for Multigraphs Visualization. In *Proceedings of the International Conference Information Visualization (IV)*, pages 50–55. IEEE, 2015. (Cité à la page 56.)

[61] Benjamin Renoust, Guy Melançon, and Tamara Munzner. Detangler: Visual Analytics for Multiplex Networks. *Computer Graphics Forum*, 34(3):321–330, 2015. (Cité à la page 56.)

[62] Arnaud Sallaberry, Faraz Zaidi, Christian Pich, and Guy Melançon. Interactive Visualization and Navigation of Web Search Results Revealing Community Structures and Bridges. In *Proceedings of the Graphics Interface Conference (GI)*, pages 105–112. ACM, 2010. (Cité à la page 79.)

[63] Manojit Sarkar and Marc H Brown. Graphical Fisheye Views. *ACM Communication*, 37(12):73–83, 1994. (Cité aux pages 11 et 18.)

[64] Time Series Analysis in Meteorology and Climatology: An Introduction. *Duchon, Claude and Hale, Robert*, volume 7. John Wiley & Sons, 2012. (Cité à la page 8.)

[65] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the Symposium on Visual Languages (VL)*, pages 336–343. IEEE, 1996. (Cité aux pages 22 et 64.)

[66] Hanghang Tong, Christos Faloutsos, Brian Gallagher, and Tina Eliassi-Rad. Fast best-effort pattern matching in large attributed graphs. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 737–746. ACM, 2007. (Cité aux pages 57 et 58.)

[67] Andrew Urquhart. The inefficiency of Bitcoin. *Economics Letters*, 148:80–82, 2016. (Cité à la page 8.)

[68] Fernanda B Viegas, Martin Wattenberg, Frank Van Ham, Jesse Kriss, and Matt McKeon. ManyEyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007. (Cité aux pages xii, 15, 16, 19, 20, 43 et 44.)

[69] F. Y. Wang, A. Sallaberry, K. Klein, M. Takatsuka, and M. Roche. SentiCompass: Interactive Visualization for Exploring and Comparing the Sentiments of Time-varying Twitter Data. In *Proceedings of the Pacific Visualization Symposium (PacifVis)*, pages 129–133. IEEE, 2015. (Cité à la page 45.)

[70] Colin Ware. *Information Visualization: Perception for Design*. Elsevier, 2012. (Cité à la page 66.)

[71] Martin Wattenberg. Baby Names, Visualization, and Social Data Analysis. In *Proceedings of the Symposium on Information Visualization (InfoVis)*, pages 1–7. IEEE, 2005. (Cité aux pages xii, 13, 14, 20 et 25.)

[72] Martin Wattenberg and Jesse Kriss. Designing for Social Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):549–557, 2006. (Cité aux pages xii, 15, 19, 20, 43 et 44.)

[73] Xifeng Yan and Jiawei Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 721–724. IEEE, 2002. (Cité aux pages 57 et 59.)

[74] Peipei Yi, Byron Choi, Sourav S Bhowmick, and Jianliang Xu. AutoG: A Visual Query Autocompletion Framework for Graph Databases. *Very Large Data Base*, 9(13):1505–1508, 2016. (Cité à la page 60.)

[75] A. Zhang. Protein interaction networks: Computational analysis, 2009. (Cité à la page 56.)