# THÈSE

**présentée à**

**L'UNIVERSITÉ BORDEAUX 1**

**ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE**

**par**

**Benjamin MARTIN**

**pour obtenir le grade de Docteur en Informatique**

---

## Analyse de structures répétitives dans les séquences musicales

---

**Soutenue le 12/12/12**

**Devant la commission d'examen composée de**

| | | |
|---|---|---|
| Président | Cédric Chauve | Professeur |
| Rapporteurs | Frédéric Bimbot | Directeur de Recherche |
| | Thierry Lecroq | Professeur |
| Directeurs | Myriam Desainte-Catherine | Professeur |
| | Pascal Ferraro | Maître de Conférences |
| Examinateurs | Jean-Julien Aucouturier | Maître de Conférences |
| | Pierre Hanna | Maître de Conférences |
| | Matthias Robine | Maître de Conférences |

# Résumé

Cette thèse rend compte de travaux portant sur l'inférence de structures répétitives à partir du signal audio à l'aide d'algorithmes du texte. Son objectif principal est de proposer et d'évaluer des algorithmes d'inférence à partir d'une étude formelle des notions de similarité et de répétition musicale.

Nous présentons d'abord une méthode permettant d'obtenir une représentation séquentielle à partir du signal audio. Nous introduisons des outils d'alignement permettant d'estimer la similarité entre de telles séquences musicales, et évaluons l'application de ces outils pour l'identification automatique de reprises. Nous adaptons alors une technique d'indexation de séquences biologiques permettant une estimation efficace de la similarité musicale au sein de bases de données conséquentes.

Nous introduisons ensuite plusieurs répétitions musicales caractéristiques et employons les outils d'alignement pour identifier ces répétitions. Une première structure, la répétition d'un segment choisi, est analysée et évaluée dans le cadre de la reconstruction de données manquantes. Une deuxième structure, la répétition majeure, est définie, analysée et évaluée par rapport à un ensemble d'annotations d'experts, puis en tant qu'alternative d'indexation pour l'identification de reprises.

Nous présentons enfin la problématique d'inférence de structures répétitives telle qu'elle est traitée dans la littérature, et proposons notre propre formalisation du problème. Nous exposons alors notre modélisation et proposons un algorithme permettant d'identifier une hiérarchie de répétitions. Nous montrons la pertinence de notre méthode à travers plusieurs exemples et en l'évaluant par rapport à l'état de l'art.

**Mots-clés : Recherche d'Informations Musicales, Structure musicale, Répétitions musicales, Algorithmique du texte**

# Abstract

The work presented in this thesis deals with repetitive structure inference from audio signal using string matching techniques. It aims at proposing and evaluating inference algorithms from a formal study of notions of similarity and repetition in music.

We first present a method for representing audio signals by symbolic strings. We introduce alignment tools enabling similarity estimation between such musical strings, and evaluate the application of these tools for automatic cover song identification. We further adapt a bioinformatics indexing technique to allow efficient assessments of music similarity in large-scale datasets.

We then introduce several specific repetitive structures and use alignment tools to analyse these repetitions. A first structure, namely the repetition of a chosen segment, is retrieved and evaluated in the context of automatic assignment of missing audio data. A second structure, namely the major repetition, is defined, retrieved and evaluated regarding expert annotations, and as an alternative indexing method for cover song identification.

We finally present the problem of repetitive structure inference as addressed in literature, and propose our own problem statement. We further describe our model and propose an algorithm enabling the identification of a hierarchical music structure. We emphasize the relevance of our method through several examples and by comparing it to the state of the art.

**Title: Repetitive structure analysis in music sequences**
**Keywords: Music Information Retrieval, Music structure, Musical repetitions, String matching**

# Préface

Cette thèse de doctorat a été effectuée au Laboratoire Bordelais de Recherche en Informatique (LaBRI) de l'université Bordeaux 1, de 2010 à 2012. Je tiens à remercier dans un premier temps mon équipe d'encadrement Pascal Ferraro, Pierre Hanna, Matthias Robine et Myriam Desainte-Catherine pour toute l'aide qu'ils m'ont apportée et pour m'avoir laissé une certaine liberté dans mes recherches, indispensable à mon sens pour mener à bien des travaux scientifiques, tout en assurant un encadrement de qualité. J'adresse ensuite des remerciements particuliers à mes rapporteurs, Frédéric Bimbot et Thierry Lecroq, qui m'ont fait l'honneur de relire mon manuscrit ainsi qu'aux autres membres extérieurs de mon jury de thèse, Jean-Julien Aucouturier et Cédric Chauve, qui ont accepté d'évaluer mon travail.

Un grand merci également à toute l'équipe de chercheurs du LaBRI sur le son et l'analyse de la musique avec qui j'ai pu échanger et collaborer, ainsi qu'aux co-auteurs des publications auxquelles j'ai participé : Pierre, Pascal, Matthias, Myriam, Julien Allali, Thomas Rocher, Vinh-Thong Ta, Dan Brown, David Janin, Florent Berthaut. Plus généralement, je remercie tous les doctorants et collègues du laboratoire avec qui j'ai pu échanger durant ces quelques années, notamment à travers les associations AFoDIB ou Labruit : Olivier, Thomas M., Florent, Julien, Allyx, Simon, Noémie-Fleur, Vincent, Stanislaw, Dominique, Thomas R., Antoine, Xavier, Sylvain, Gabriel, Ève, Anne-Laure, Nicolas, Rémi, Émilie, Anaïs, Pierre, Lætitia... et j'en oublie probablement beaucoup. Ce sont aussi ces échanges, scientifiques, associatifs, musicaux ou autres, qui rythment le travail de doctorant et le rendent passionnant. Je souhaite également remercier les différents responsables de modules d'enseignement avec lesquels j'ai travaillé à l'ENSEIRB-MATMECA pour leur soutien pédagogique, et principalement Aymeric Vincent, Georges Eyrolles, Floréal Morandat, Toufik Ahmed et Denis Lapoire.

En outre, je tiens à remercier chaleureusement tous les doctorants de diverses disciplines avec lesquels j'ai échangé, notamment à travers l'association AquiDoc des jeunes chercheurs d'Aquitaine. M'inpliquer dans cette dernière a été particulièrement enrichissant à la fois d'un point de vue humain et pour mon expérience professionnelle, en me donnant notamment un certain recul sur le doctorat et sur la condition des jeunes chercheurs. Je souhaite inciter tout doctorant lisant ces quelques lignes à s'intéresser de près et à s'impliquer, autant qu'il le peut, dans cette association.

Mes derniers remerciements, et non les moindres, vont à tous mes amis qui se reconnaîtront et qui m'ont soutenu tout au long de ma thèse, ainsi qu'à ma famille et mes parents qui m'ont toujours poussé à poursuivre les sujets qui me passionnent, dans mes études comme dans ma vie professionnelle. C'est en grande partie grâce à eux que j'ai réussi dans ces travaux à réunir mes passions pour l'informatique et la musique, et c'est avec une certaine fierté que je leur dédie aujourd'hui ce manuscrit.

# Table des matières

# Introduction

Depuis une quinzaine d'années, la musique est au cœur d'un changement radical d'accès à l'information. Conséquence du développement fulgurant d'internet, des technologies numériques et de la multiplication des plateformes de distribution dématérialisée, la musique n'a jamais été aussi largement accessible qu'à l'heure actuelle. La quantité considérable de morceaux disponibles, de l'ordre de plusieurs millions de titres, a entraîné un effort conséquent des acteurs de l'industrie musicale pour développer des méthodes de parcours efficace des bases de données musicales, permettant à tout utilisateur de retrouver, de trier ou de découvrir les titres qui lui plaisent. Il est cependant impossible pour un humain de garder une trace de l'ensemble des morceaux existants et des relations entre ceux-ci. Ce constat est l'une des raisons qui a poussé les chercheurs et fournisseurs de contenu à utiliser l'outil informatique pour l'analyse et la comparaison automatiques des données musicales. En particulier, un enjeu majeur de l'analyse de l'information musicale consiste à utiliser les éléments acoustiques des morceaux de musique pour déduire certaines propriétés perçues par l'oreille humaine, et ainsi comparer ou représenter les œuvres par leur contenu même et sans l'aide d'annotations supplémentaires.

La *similarité* est une notion clé en analyse du contenu musical. Lorsqu'elle est constatée entre des morceaux distincts, elle peut témoigner d'une volonté de la part du compositeur ou de l'interprète de ressemblance à une œuvre, en proposant une reprise ou une nouvelle interprétation par exemple. Elle peut également être relevée entre morceaux d'un même style musical, d'un même auteur, d'un même interprète, d'une même époque *etc.*, et joue ainsi un rôle essentiel dans l'accès à la donnée musicale basé sur le contenu. En outre, lorsqu'elle est constatée entre différents extraits à l'intérieur même d'un morceau, la similarité est susceptible de mettre en évidence des répétitions musicales. Ces *structures répétitives*, présentes à différentes échelles, caractérisent une certaine organisation temporelle du morceau de musique. Cette dernière est un aspect fondamental dans la culture musicale occidentale. Ainsi, certains styles musicaux sont construits autour d'un agencement caractéristique de structures répétitives, composé par exemple de couplets et refrains en musique populaire, ou encore d'une forme répétitive prédéfinie comme dans les fugues, sonates ou menuets en musique classique. Plus généralement, la répétition musicale se produit à de nombreuses échelles temporelles, allant par exemple dans le cas de la musique classique depuis la note jusqu'au mouvement en passant par la phrase, le thème ou encore le motif. C'est pourquoi l'analyse des structures répétitives d'un morceau est une source d'informations riche sur le contenu musical qui ouvre un large éventail d'applications, dont certaines sont exposées dans ce document.

La musique a ceci de spécifique que l'auditeur humain est généralement capable dans sa perception de combiner et prendre en compte simultanément de nombreux critères acoustiques, quelle que soit sa culture musicale. La tâche de modélisation mathématique de la musique polyphonique, en revanche, présente une complexité importante. Ainsi, à l'heure actuelle, l'analyse automatique de la musique pour identifier des notes, des rythmes ou des instruments par exemple n'atteint pas les niveaux de performance de la perception humaine. En particulier, la détection au-

tomatique de similarités musicales est un problème difficile et lié à de nombreux facteurs, alors même que le cerveau humain est souvent en mesure de comparer implicitement et de qualifier aisément les ressemblances et les répétitions musicales perçues.

Ce document rend compte des travaux de thèse effectués sur le thème de l'analyse du contenu musical pour l'*inférence de structures répétitives*. L'objectif principal de cette thèse est ainsi de proposer des méthodes d'analyse de répétitions caractéristiques dans les morceaux de musique occidentale. L'aspect non trivial des notions de similarité et de répétition musicale pour l'analyse nous amène à utiliser des techniques d'identification adaptées aux nombreuses variations musicales susceptibles d'apparaître entre motifs perceptivement jugés comme similaires. En particulier, les travaux présentés dans cette thèse se basent à de nombreuses reprises sur des outils d'*algorithmique du texte* et de segmentation des séquences biologiques. Le bienfondé d'une telle adaptation pour l'analyse musicale est justifié plus précisément dans la suite de ce document par la richesse de la représentation séquentielle et de la structuration répétitive dans les deux cadres applicatifs, biologique et musical. Ces recherches s'inscrivent dans une approche scientifique validée par des évaluations concrètes. Ainsi, pour l'ensemble des problèmes que nous définissons, une solution algorithmique est apportée et évaluée sur des données musicales. En particulier, la concordance de l'analyse effectuée avec des éléments liés à la *perception* musicale est cruciale. Dans les travaux exposés dans cette thèse, la mise en application n'est pas systématiquement effectuée sur des bases de données à l'échelle du contenu disponible dans l'industrie musicale, pour des raisons pratiques d'obtention des données ou de droits d'auteur. Cependant, l'efficacité calculatoire est discutée dans les différents systèmes proposés, et plusieurs techniques de réduction significative du temps de calcul sont étudiées.

Les travaux présentés dans ce document sont organisés de la manière suivante. Le Chapitre 1 introduit plus précisément le contexte des travaux présentés. Les chapitres suivants décrivent et évaluent les différentes étapes menant à l'extraction de structures répétitives depuis le signal audio. Le Chapitre 2 décrit et justifie la méthode permettant de représenter le signal audio par une séquence de symboles représentative d'un critère musical. Le Chapitre 3 présente les outils de comparaison adaptés à de telles séquences permettant d'identifier des similarités musicales. Le Chapitre 4 emploie ces techniques de comparaison pour identifier des structures répétitives simples. Le Chapitre 5 propose alors un algorithme d'inférence de la structuration d'un morceau de musique à partir d'une combinaison de structures répétitives simples. Les conclusions, travaux en cours et perspectives sont enfin présentés en Chapitre 5.4. Les paragraphes suivants décrivent plus en détail la composition de chaque chapitre.

**Chapitre 1**
Ce chapitre introduit d'abord les enjeux majeurs de la problématique de recherche dans laquelle ces travaux s'inscrivent. Nous décrivons le contexte actuel de distribution et d'accès à la donnée musicale, et présentons les approches relatives à nos travaux pour l'analyse automatique d'informations musicales en les illustrant par plusieurs cas d'utilisation. Ce chapitre introduit ensuite la notion de répétition

en musique, et présente différentes applications à son analyse automatique. Nous justifions le choix d'une représentation séquentielle de la musique, et exposons les principes algorithmiques utiles à la détection efficace de structures répétitives.

**Chapitre 2**
Ce chapitre décrit une méthode de représentation de l'information musicale sous la forme de séquences. Nous discutons et justifions notre choix du critère musical de représentation dans le cadre de la recherche de répétitions. La suite du chapitre décrit les outils de traitement du signal audio permettant de le représenter sous forme de séquence de descripteurs musicaux. Afin de permettre d'estimer des similarités musicales, nous détaillons alors des techniques issues de l'état de l'art permettant de comparer ces descripteurs entre eux.

**Chapitre 3**
Ce chapitre introduit les notations et techniques algorithmiques nécessaires à la comparaison de séquences de symboles, inspirées notamment par les outils d'analyse de séquences biologiques. Nous détaillons une application à l'estimation de la similarité musicale et l'évaluons dans le cadre de la recherche de reprises. Nous introduisons ensuite une technique d'identification à l'efficacité calculatoire élevée en adaptant une méthode bio-informatique d'indexation heuristique, BLAST, à l'information musicale, que nous testons sur plusieurs bases de données de taille conséquente.

**Chapitre 4**
Ce chapitre étudie plusieurs types de répétitions dans les séquences musicales. Pour un extrait donné au sein d'un morceau de musique, nous présentons d'abord une méthode d'identification de la meilleure répétition de celui-ci dans le même morceau, que nous évaluons par un test perceptif dans le cadre applicatif de la reconstruction de données audio manquantes. Nous étudions ensuite la répétition musicale dans un cas plus général et sans fixer l'une de ses occurrences. Nous introduisons alors une répétition particulière, dite *majeure*, et proposons un algorithme d'identification de celle-ci. Nous comparons cette répétition analysée à des éléments de perception de la structure musicale, puis l'utilisons pour l'indexation d'un système de recherche de reprises.

**Chapitre 5**
Ce chapitre décrit des méthodes d'inférence des structures répétitives contenues dans des morceaux de musique. Nous étudions tout d'abord les travaux existants et mettons en valeur plusieurs limitations dans la position du problème que nous proposons de redéfinir. Nous proposons ensuite une modélisation pour la structuration répétitive d'un morceau de musique sans pré-connaissance particulière sur ses répétitions. Nous présentons alors un algorithme d'inférence et le confrontons aux techniques de l'état de l'art dans le cadre d'une comparaison avec des annotations perceptives des structures musicales.

Chapitre 1

# Contexte

La démocratisation de l'informatique et les avancées technologiques en termes de réseau, de puissance de calcul ou encore de capacité de stockage de données, associées à l'avènement de formats de compression audionumérique, ont précipité ces dernières années le développement de bases de données musicales de l'ordre de plusieurs millions de titres. D'abord stocké sur les supports personnels tels l'ordinateur, le lecteur nomade ou le téléphone mobile, le contenu musical a été au centre de changements majeurs tant dans son mode de distribution que dans les habitudes d'écoute [Don09]. En particulier, la multiplication ces dernières années de solutions de stockage à distance, souvent désignées par le terme anglais *cloud*, repoussent les limites du stockage personnel et permettent à un utilisateur de bénéficier de bibliothèques musicales numériques de plusieurs dizaines de milliers de titres depuis n'importe quel appareil de lecture compatible.

En outre, faisant écho au développement de connexions internet dites "haut-débit" et à une popularité croissante des services internet, de nombreux serveurs et plateformes de téléchargement ont été mis en place, proposant des millions de contenus musicaux accessibles à tout utilisateur de manière quasi-instantanée. Dans cette configuration, l'utilisateur n'est plus nécessairement propriétaire de la ressource numérique mais a accès à un catalogue complet d'œuvres accessible via des plateformes de téléchargement ou de diffusion (*streaming*). Ainsi, l'Observatoire de la Musique [1] recense notamment en février 2012 [Nic12] plus de 78 sites et portails accessibles depuis la France, classés en quatre catégories : les plateformes de téléchargement, les services de radios et de diffusion, les sites communautaires et les sites éditoriaux, de création ou innovants.

Ces récentes avancées dans la dématérialisation du contenu musical et dans l'accès à celui-ci ont provoqué une explosion du nombre de titres disponibles. Par exemple, en juillet 2012, on compte plus de 23 millions de titres [Spo12] sur une plateforme de *streaming* telle que *Spotify* [2], et plus de 28 millions de titres [Dig12] pour les plateformes de téléchargement telles que l'*Itunes Store* [3]. La mise à disposition d'une telle quantité de morceaux a ceci de paradoxal qu'elle est décorrélée du temps pratique d'écoute, et peut même dépasser en temps cumulé l'échelle de plusieurs vies humaines [CVG+08].

## 1.1 Recherche d'Informations Musicales

Le constat d'une telle profusion de l'information musicale a incité le développement de méthodes pour permettre d'organiser, de trier et de parcourir ces bases de données suivant des critères musicaux suffisamment pertinents pour permettre de

1. http://rmd.cite-musique.fr/observatoire/
2. http://www.spotify.com
3. http://www.apple.com/fr/itunes/whats-on/#music

| | |
|---|---|
| (i) | Trouver toutes les œuvres d'un compositeur donné |
| (ii) | Trouver tous les enregistrements d'un interprète donné |
| (iii) | Trouver le titre d'une chanson à partir de quelques paroles, ou vice-versa |

FIG. 1.1 – *Exemples de requêtes adaptées à l'approche par métadonnées (selon [Dow03]).*

discriminer les quelques titres qui intéresseront un utilisateur donné. La Recherche d'Informations Musicales, ou *Music Information Retrieval*, rassemble ces méthodes.

### 1.1.1 Recherche basée contexte

À l'heure actuelle, la grande majorité des dispositifs de lecture permet d'organiser et de chercher à travers des collections musicales à partir d'informations *éditoriales*, telles que le titre, l'artiste, l'album, le numéro de piste ou encore la date de parution [CVG+08, Dow03, Par06b]. Ces informations sont parfois enrichies de données complémentaires comme la localisation de l'enregistrement ou le lien hypertexte menant au site internet du compositeur. Ces *métadonnées*, juxtaposées à l'information musicale, fournissent des descriptions annexes au contenu musical lui-même, et sont destinées à indexer facilement les titres afin de les retrouver aisément sur les plateformes internet, en magasin ou encore au sein de sa propre bibliothèque musicale. Ainsi, leur spécification permet par exemple de répondre aux requêtes indiquées en Figure 1.1.

Dans le cas des plateformes de diffusion, par exemple, et contrairement aux collections personnelles, il est très probable qu'un utilisateur donné ne connaisse qu'une petite partie des morceaux accessibles. Or, bien que l'étiquetage par métadonnées présente un intérêt évident pour retrouver directement un titre dont on connaît précisément les caractéristiques, l'approche tombe vite en défaut lorsqu'il s'agit, par exemple, de découvrir de nouveaux morceaux. Dans un système de recherche contextuel, on formule en effet l'hypothèse qu'un utilisateur connaît et est capable d'expliciter la métainformation recherchée. Les requêtes citées en Figure 1.1 s'avèrent non pertinentes si le compositeur, le titre et l'album sont inconnus de l'utilisateur. Ainsi, les possibilités offertes à l'utilisateur pour identifier des titres musicalement proches s'avèrent fortement limitées en n'employant que cette simple métainformation.

L'apparition de bases de données de l'ordre de centaines de milliers, voire de millions de titres a donc amené le besoin de développer de nouvelles technologies afin de permettre une interaction plus aisée et plus significative avec une telle quantité d'information, en particulier pour découvrir de nouveaux titres. L'approche la plus utilisée à l'heure actuelle consiste à enrichir l'ensemble des métadonnées caractérisant les morceaux de musique, en fournissant par exemple des étiquettes *culturelles*, subjectives, permettant de rapprocher les morceaux en fonction de traits musicaux communs. Ainsi, des outils tels que *All Music Guide*[1], *Gracenote*[2] ou encore *MusicBrainz*[3] sont en mesure de fournir une catégorisation des morceaux de musique

1. http://www.allmusic.com
2. http://www.gracenote.com
3. http://www.musicbrainz.com

(i) Identifier un morceau pouvant intéresser un utilisateur à partir des préférences de son entourage
(ii) Identifier un ensemble de morceaux du même genre musical à partir des habitudes d'un groupe de personnes
(iii) Identifier des artistes similaires à partir de statistiques d'écoute

FIG. 1.2 – *Exemples de requêtes adaptées à l'approche sociale (adapté de [SK09]).*

à partir de centaines d'étiquettes subjectives, ou *tags*, attribuées par des experts et/ou des utilisateurs. Ces étiquettes peuvent concerner différents aspects liés au contexte de l'œuvre, à des éléments culturels ou encore à l'émotion ressentie à son écoute, comme le style, l'humeur, le contexte d'écoute favori, *etc.* [KSM+10, SK09]. Elles constituent également des *métadonnées* qui sont adjointes au contenu musical.

Bien que très largement utilisée à l'heure actuelle, cette approche souffre de plusieurs limitations qui la rendent difficile à mettre en pratique. D'abord, une telle prolifération du contenu musical suppose un effort considérable d'annotation manuelle dans le cas de l'annotation de métadonnées, qu'il s'agisse de fournir des informations éditoriales ou des étiquettes subjectives. Ainsi, pour le service de web-radios automatique *Pandora*[1] qui se base sur une large source de métadonnées développée dans le cadre du *Music Genome Project*, la durée d'annotation d'un nouveau morceau par un expert est estimée entre 20 et 30 minutes [CVG+08]. À l'échelle des plus grandes bases de données audionumériques, une telle annotation doit alors être effectuée par de nombreux individus, rendant l'approche extrêmement coûteuse et risquant de générer une certaine variabilité de la qualité des annotations subjectives [CVG+08, Fre06].

Pour contourner ce problème et permettre un accès plus large aux bases de données musicales, une approche récente consiste à considérer l'information musicale dans son contexte social d'utilisation [SK09]. Dans cette approche, les préférences musicales de la masse sont appliquées à l'individu. Par exemple, en analysant automatiquement les statistiques d'écoute et en les combinant à des métadonnées indiquées par la communauté d'utilisateurs, le système *LastFM*[2] est capable d'extrapoler les morceaux favoris d'un utilisateur et ainsi de découvrir de nouveaux titres. L'utilisation de contenu extrait de moteurs de recherche [WL02], l'analyse de listes de lecture [PWL01] ou encore la prise en compte des statistiques d'utilisation de réseaux pair-à-pair [EWBL02] sont d'autres exemples de stratégies basées sur le contexte pour établir des mesures de similarité au sein de grandes bases de données musicales. La Figure 1.2 illustre plusieurs exemples de requêtes pouvant être satisfaites par une approche considérant le contexte social d'écoute.

Cette approche est particulièrement développée à travers une intégration sur des réseaux sociaux spécifiques tels que *SoundCloud*[3] ou *Exfm*[4], notamment, où l'échange et le partage de musique permettent d'enrichir le parcours de grandes bases de données audionumériques. Cependant, plusieurs limites sont liées à cette logique : ici, le parcours de bases de données est motivé par imitation d'un compor-

1. http://www.pandora.com (disponible uniquement depuis les États-Unis)
2. http://www.lastfm.com
3. http://www.soundcloud.com
4. http://www.ex.fm

(i) Trouver toutes les reprises, ou enregistrements par différents artistes, d'une même chanson d'origine
(ii) Identifier tous les morceaux dont l'orchestration comporte un ensemble donné d'instruments de musique
(iii) Identifier tous les titres de même style, ou de même forme qu'un morceau donné
(iv) Étant donné une mélodie, l'air d'une chanson ou le thème d'une symphonie par exemple, identifier le morceau d'origine
(v) Ne pas jouer le refrain à la lecture d'un morceau
(vi) Masquer la voix du chanteur lors de l'écoute d'un morceau

FIG. 1.3 – *Exemples de requêtes adaptées à l'approche basée sur le contenu (selon [Dow03] et [Par06a]).*

tement social, et non par la donnée ou métadonnée du contenu numérique. L'aspect social de la découverte de nouveaux morceaux est ainsi intrinsèquement lié à la popularité des morceaux de musique, donnant par exemple plus de crédit à un morceau présent dans la base de données depuis une longue durée qu'à un morceau qui vient d'être publié (effet de démarrage à froid, ou *cold start*). Cette approche fait également l'hypothèse d'une certaine uniformité du goût musical à travers les utilisateurs. De ce fait, la connaissance musicale d'un groupe est partagée entre ses membres, la découverte d'œuvres se limitant à cette connaissance commune.

Plus généralement, les approches basées sur le contexte des œuvres musicales présentent plusieurs limites conceptuelles qui ont motivé le développement de méthodes alternatives de recherche d'information. L'inconvénient majeur est une fréquente inadéquation des métadonnées, liées au contexte, et à la recherche d'information souhaitée, souvent liée au contenu. Les données contextuelles peuvent ainsi s'avérer non pertinentes, non fiables ou incomplètes [Ori06]. L'approche basée sur le contenu musical permet de contourner ces problèmes liés à l'insuffisance des métadonnées en permettant un parcours de la donnée musicale indépendamment de son contexte.

### 1.1.2 Recherche basée contenu

Une solution au problème de parcours de grandes bases de données musicales consiste à décrire la musique par son contenu musical même, et non plus par des informations relatives au contexte. Une telle approche possède l'avantage crucial de donner une description automatique pertinente sur le plan musical. La recherche basée contenu s'intéresse à l'analyse automatique du contenu audionumérique selon des critères musicaux, tels que les instruments, les notes ou les rythmes présents dans un enregistrement. Elle se concentre en particulier sur la détermination des structures musicales abstraites encodées dans les signaux [Par06a], et permet ainsi de nouvelles interactions avec l'information musicale. La Figure 1.3 présente un ensemble de requêtes pouvant être satisfaites en employant une approche basée sur le contenu musical.

La recherche basée sur le contenu musical s'effectue sur des critères de caractérisation des morceaux de musique déduits du support de codage musical, tel que

le signal audionumérique. Par conséquent, elle ne requiert pas d'effort d'annotation manuelle et reste indépendante de la popularité des morceaux de musique. De ce fait, l'approche par le contenu élargit le champ d'interaction et de parcours avec des grandes bases de données musicales en ouvrant la recherche d'information à différentes familles d'applications. Les paragraphes qui suivent décrivent brièvement plusieurs cas d'utilisation pratique d'une approche basée contenu.

### Identification et requête par l'exemple

Les techniques dites d'identification audionumérique, ou *fingerprinting*, cherchent à associer au contenu audionumérique d'une œuvre une empreinte (ou signature) liée au contenu musical, qui permet de l'identifier de manière unique au sein d'une base de données [CBKH05, GMS12]. Par exemple, le système *Shazam*[1] [Wan03] permet à un utilisateur équipé d'un téléphone mobile d'obtenir le titre d'un morceau en cours de lecture au bout de quelques secondes d'enregistrement, même en présence d'un environnement bruyant ou d'un système d'enregistrement peu performant. Les systèmes d'identification audionumérique commencent par isoler un ensemble de caractéristiques du signal avant de comparer celles-ci à une base de données située sur un serveur central. Même s'ils peuvent permettre certaines variations dans la qualité de l'enregistrement, les techniques de *fingerprinting* à proprement parler ne sont en mesure d'identifier un extrait joué que si l'exacte copie est présente dans la base de données centralisée. La technologie d'identification est aujourd'hui particulièrement développée dans une optique de gestion des droits numériques, sur des plateformes de lecture de flux vidéo par exemple [CE10].

La requête par l'exemple permet à un utilisateur de rechercher une œuvre musicale à partir d'un extrait audionumérique qui présente un nombre significatif de ressemblances avec l'œuvre originale. Par exemple, les systèmes de requête par fredonnement proposent à un utilisateur de chanter ou fredonner quelques secondes d'une mélodie, puis essaient d'identifier le morceau d'où provient celle-ci [DBP+07]. Ainsi, le programme *SoundHound*[2] permet d'effectuer une telle recherche par fredonnement à partir d'un appareil mobile en enregistrant et comparant les extraits fredonnés par les différents utilisateurs.

### Classification

La classification de morceaux de musique intègre des techniques d'identification de similarité musicale afin de former un nombre fini de groupes ou classes de similarité. Au niveau applicatif, les algorithmes d'estimation de similarité musicale sont presque systématiquement adjoints à des techniques de classification pour identifier une caractéristique de haut niveau de l'œuvre musicale. Par exemple, les travaux du projet CUIDADO [VHP02] visent notamment à qualifier de manière automatique des échantillons sonores en fonction de leur instrumentation (voir par exemple [HBPD03]), afin de fournir des structures de description pertinentes permettant d'assister la production sonore. L'identification du compositeur d'une œuvre [PS01], le regroupement par style [TC02] ou encore l'inférence de concepts

1. http://www.shazam.com
2. http://www.soundhound.com

affectifs tels que l'humeur [LLZ06] ou l'émotion [KSM+10] sont d'autres exemples d'applications à la classification basée sur le contenu musical.

#### Recherche par similarité musicale

D'une manière générale, la perception humaine de la musique nous permet souvent d'estimer facilement et implicitement si deux morceaux de musique sont similaires. Un tel jugement est basé sur de nombreuses facettes de l'information musicale [Dow03], liées à la fois à des critères éditoriaux (titre, artiste), culturels (style, langue), musicologiques (mélodie, harmonie, structure), perceptifs (énergie, texture) ou encore cognitifs (expérience, mémoire) [Jeh05a, AP02, HAE03]. La similarité musicale consiste à estimer la ressemblance entre plusieurs œuvres à partir de critères musicaux. Face à la complexité de la perception de similarité, la recherche basée contenu n'utilise que le signal audionumérique pour analyser les ressemblances. La recommandation d'œuvres similaires ou encore la détection de reprises ou de plagiats sont des exemples d'applications de l'estimation de la similarité musicale.

## 1.2 Répétition musicale

La répétition musicale est une notion centrale pour la recherche d'informations musicales basée sur le contenu. La musique est une information structurée autour de caractéristiques perçues par l'oreille humaine. En fonction de son expertise musicale, un auditeur est en mesure de percevoir, par exemple, une mélodie chantée, une ligne d'accompagnement instrumentale, une partie rythmique, un accord *etc.* Les ressemblances et contrastes entre ces différentes entités sonores forment un ensemble de structures répétitives dont l'apparition, la transformation ou l'évolution décrivent tout au long d'un morceau une architecture spécifique à l'œuvre musicale [Ong07]. Middleton [Mid99] constate cet aspect fondamental de la répétition en musique :

> *« La répétition joue un rôle particulièrement important en musique - quel que soit le style musical considéré. Dans le cas de la musique populaire, les procédés de répétitions sont particulièrement développés. »*

À l'image d'un bâtiment, un morceau de musique est construit selon une certaine *architecture*. Que celle-ci soit explicite ou implicite, elle est présente dans la quasi-totalité des styles de musique. Ainsi, Levitin [Lev08, p.167] explique que la mémorisation des motifs musicaux et leur agencement jouent un rôle prépondérant dans notre perception de la musique :

> *« La musique est basée sur la répétition. La musique fonctionne parce que nous retenons les tons que l'on vient d'entendre et les mettons en relation avec ceux qui sont en train d'être joués. Ces groupes de tons – phrases – peuvent survenir plus tard dans le morceau dans une variation ou une transposition qui excite notre système de mémoire en même temps qu'il active nos centres émotionnels. [...] La répétition, lorsqu'elle est effectuée avec talent par un compositeur exercé, est émotionnellement satisfaisante pour notre cerveau et rend l'expérience d'écoute aussi plaisante qu'elle peut l'être. »*

Dans cet extrait, Levitin [Lev08] souligne également l'aspect *approché* de la répétition de motifs musicaux, qui peuvent dans son exemple être altérés par des

*variations* ou *transpositions*. Les répétitions musicales ne sont pas nécessairement effectuées de manière exacte. Un motif répété peut ainsi avoir subi une série d'altérations musicales telle qu'une transposition de ses notes, une modification de sa durée, la suppression ou l'ajout de notes *etc.* L'altération peut également être liée à des paramètres acoustiques, dus par exemple à la grande difficulté à reproduire des sons exactement identiques sur les instruments de musique acoustiques. Les variations résultantes dans les motifs musicaux peuvent être volontaires, c'est-à-dire souhaitées par le compositeur ou l'interprète du morceau de musique, ou involontaires, dues à des limites physiques ou des éléments sonores imprévus. Dans les deux cas, cette grande variété susceptible d'apparaître au sein de structures répétitives implique de considérer des techniques d'analyse robustes à de telles inexactitudes.

La musique occidentale, en particulier, est composée de nombreuses structures répétitives. Par exemple, les termes de "couplet" ou "refrain" désignent des sections fréquemment répétées dans les morceaux de musique populaire. La musique classique compte également un large vocabulaire de sections répétées pouvant être caractéristiques de méthodes de composition, sous les noms de "motif", "thème", "phrase", "cadence" ou "mouvement", par exemple [Ste79]. Bien que ces termes ne soient pas forcément définis uniquement par leur caractère répétitif, leur récurrence joue un rôle central dans la perception de l'œuvre musicale [Esc88].

### 1.2.1 Enjeux et intérêt

Analyser la structuration d'un morceau de musique revient à décrire l'organisation temporelle des éléments musicaux qui la composent. L'analyse des répétitions d'un morceau de musique fournit ainsi une sorte de *"dé-composition"*, abstraite, des processus musicaux mis en œuvre au long de ce morceau. Par exemple, l'aspect répétitif du *Boléro* de Ravel relève notamment d'un choix de composition, l'auteur souhaitant donner une impression de *« Danse d'un mouvement très modéré et constamment uniforme »* [Esc88]. Dans le cas idéal d'une analyse parfaite de la répétition, les structures identifiées correspondent alors, dans une certaine mesure, aux motifs arrangés par le compositeur. Dès lors, les répétitions identifiées dans un morceau prennent une signification musicale forte reflétant sa composition. L'intérêt de l'analyse de celles-ci est d'exploiter directement cette signification, afin par exemple de restructurer le morceau de manière automatique et cohérente, d'identifier une forme caractéristique de celui-ci, de le simplifier ou de simplement enrichir sa description analytique. La suite de cette section décrit plus en détail ces différentes applications.

#### Enrichissement du contenu audio

La possibilité d'analyser et de repérer des structures répétitives ouvre un large éventail d'applications. Détecter les répétitions est d'abord utile à l'enrichissement du parcours de contenu musical, en permettant par exemple à un utilisateur d'écouter un morceau de musique en évitant certaines sections [Got03, Vin05]. Bien que la visualisation des structures répétitives ne soit pas encore proposée sur les plateformes commerciales, quelques outils développés par des laboratoires de recherche ont été proposés à cet effet, dont *SemanticHIFI* [Vin05], *SmartMusi-*

*cKIOSK* [Got03], ou encore plus récemment *Songle*[1] [GYF+11]. Pour une revue des supports qui existent pour la visualisation de structures répétitives, le lecteur est invité à consulter [BPMW12].

Une telle analyse peut également apporter un support à la manipulation du son dans différents contextes, en fournissant par exemple à l'ingénieur du son un outil de traitement permettant d'assurer une cohérence musicale, à l'image de [TC99], ou encore en aidant le concepteur lumière à synchroniser avec précision des effets visuels [KAC05].

#### Restructuration du signal

La connaissance de structures répétitives dans un morceau de musique peut être utilisée dans un objectif de recomposition du morceau. Bien que peu considérée pour le signal audio musical, cette application est fréquemment utilisée dans le cas de l'image ou de la vidéo, et fait appel à la notion de *textures* [SSSE00, Ash01]. En analysant les structures répétitives d'un extrait vidéo [SSSE00] ou d'une image [Ash01, EL99], les textures identifiées permettent de reconstruire des sections de l'objet considéré, de le prolonger (dans le temps pour la vidéo ou dans l'espace pour l'image), ou encore de synthétiser de nouveaux objets à partir de ces structures. Bien que la notion de texture soit également introduite dans le cadre des objets sonores [LWZ04b], elle est rarement utilisée dans un but de recomposition d'œuvres musicales. L'analyse des structures répétitives peut cependant permettre de restaurer un signal audio ou encore de le prolonger en identifiant les répétitions les plus attendues par un auditeur [Hur06], comme proposé par exemple par Jehan [Jeh05a]. Une utilisation des structures répétitives pour cette application est proposée en Section 4.1 du Chapitre 4.

#### Identification de formes musicales

Certains morceaux de musique sont caractérisés par l'agencement de segments structurels caractéristiques, ou *forme* musicale. C'est le cas par exemple des œuvres de formes *sonate*, *rondo* ou *menuet*, dont la composition suit généralement une série de règles structurelles précises [Eme98]. L'analyse des structures répétitives peut ainsi être appliquée à la classification automatique de tels morceaux en fonction de l'agencement de leurs répétitions [MK07]. De plus, la structuration répétitive des morceaux de musique peut être employée afin de comparer plusieurs morceaux en ne tenant compte que de leur aspect répétitif. Cette approche, proposée dans plusieurs études récentes [MRH09, Bel11, GSMA12], s'avère pertinente pour identifier des ressemblances de composition de morceaux dont les répétitions sont proches alors que le contenu musical-même peut fortement varier.

#### Résumé musical et *thumbnailing*

L'analyse de la répétition permet également d'identifier des sections jugées représentatives d'une œuvre musicale, appelées *thumbnails* [LC00, BW05, AS02]. Les extraits audio sont utilisés par les plateformes de téléchargement, telles qu'*Amazon*[2],

1. http://www.songle.jp
2. http://www.amazon.fr

qui proposent aux utilisateurs d'écouter une portion audio avant l'achat. La mise en place d'une analyse des répétitions permettrait, pour un tel système, de sélectionner le segment le plus caractéristique du morceau souhaité, et ainsi donner un aperçu représentatif de l'œuvre à l'acheteur potentiel. De tels extraits audio peuvent aussi être utilisés afin d'identifier rapidement des similarités entre morceaux de musique [GOH06], comme expliqué plus précisément en Section 4.2.

En outre, cette analyse permet en identifiant les processus compositionnels mis en jeu de générer un résumé musical de l'œuvre [Pee04]. Dans le cas de répétitions très similaires et particulièrement récurrentes, l'analyse structurelle peut également être utilisée pour factoriser des sections de morceaux de musique et ainsi compresser sensiblement leur représentation [Rao04].

#### Amélioration de méthodes d'analyse

La connaissance de structures répétitives au sein d'un morceau de musique peut également être employée comme technique d'amélioration de méthodes d'analyse annexes. Par exemple, Mauch *et al.* [MND09] parviennent à améliorer de manière significative une méthode d'analyse des accords notamment en prenant en compte la répétition de sections identifiées dans des morceaux de musique. Rocher [Roc11] relève également une augmentation de la précision de son système d'identification des accords en prenant en compte les structures répétitives proposées dans le cadre de cette thèse.

## 1.3 Algorithmique des séquences musicales

L'analyse du signal audio ainsi que sa description musicale automatique sont des procédés complexes débouchant sur de nombreuses applications, telles que celles présentées dans les sections précédentes. Malgré une certaine progression de la recherche en analyse du signal au cours des dernières années, l'efficacité de l'analyse automatique du signal audio n'atteint pas à ce jour la performance de l'oreille humaine [DG09]. Devant l'incapacité à représenter de manière précise un critère musical, il semble inenvisageable de représenter de manière fiable toute abstraction musicale de plus haut niveau, en particulier la structuration d'un morceau de musique.

Cependant, la représentation de la musique sous forme de *séquences* permet de contourner cette limitation conceptuelle pour le cas particulier de l'analyse de structures répétitives. Tout morceau de musique peut être représenté comme une suite de sons organisés dans le temps. En pratique, l'analyse automatique du signal peut fournir une séquence de symboles décrivant un aspect particulier de la musique. Dans cette approche, chaque symbole représente une portion de signal, de durée constante ou variable. La modélisation séquentielle permet alors de modéliser la temporalité de la musique : deux symboles se suivant dans la séquence correspondent à deux instants consécutifs dans le signal. En particulier, si le morceau étudié contient une répétition musicale, alors il est très probable que les séquences correspondant à celle-ci soient similaires sous réserve d'une fiabilité suffisante de la représentation symbolique. Les répétitions musicales peuvent ainsi être analysées même si les symboles calculés à partir du signal ne donnent qu'une description grossière de l'information musicale souhaitée [DG09]. Pour cette raison, la représentation

de la musique sous forme de séquences, particulièrement privilégiée pour l'analyse automatique du contenu musical [CS06], est tout à fait adaptée pour l'analyse de structures répétitives. Le Chapitre 2 détaille notre technique d'obtention d'une séquence représentative d'un critère musical à partir d'un signal audio.

### 1.3.1 Méthodes algorithmiques

Comme détaillé en Section 1.2, la notion de répétition en musique sous-entend une possible variation du contenu musical. Représenter la musique par des séquences pour l'analyse de la répétition suppose donc de considérer des méthodes de comparaison dites *approchées* qui soient robustes à certaines variations. Plus précisément, on sous-entend par le terme *approché* que le système de comparaison permet d'identifier comme similaires des séquences présentant un certain nombre de symboles différents, et tolère ainsi des erreurs qui peuvent figurer entre les séquences comparées.

Les problèmes de comparaison approchée de séquences sont étudiés dans de nombreux contextes applicatifs distincts de l'analyse musicale, tels que l'analyse du texte, la compression de données ou encore la bio-informatique [CIR98]. En effet, de nombreux objets peuvent être représentés sous forme de séquences : ainsi, un fichier texte est une séquence de symboles sur un alphabet ASCII, un fichier binaire est une séquence sur un alphabet d'un nombre fini de valeurs possibles et un code ADN est une séquence sur un alphabet de quelques symboles représentant des molécules organiques.

En particulier, les systèmes de comparaison approchée de séquences sont très étudiés dans le contexte bio-informatique. D'une part, les bio-informaticiens manipulent des données issues d'expérimentations et comportant un certain nombre d'erreurs. Ainsi, les séquences moléculaires (l'ADN, l'ARN ou les séquences d'acides aminés pour les plus connues) sont susceptibles de comporter des incohérences locales qui conduisent les techniques de comparaison à évaluer la similarité avec une certaine tolérance [Gus97]. D'autre part, ces séquences biologiques sont par leur nature issues de processus de mutation que les techniques de comparaison cherchent à révéler et à modéliser [Gus97]. L'évolution se base sur la réutilisation, la duplication et la modification des structures biologiques existantes [Gus97] ; l'objectif de la comparaison de séquences en bio-informatique est ainsi d'identifier ces structures répétitives afin de caractériser des phénomènes biologiques liés à de fortes similarités fonctionnelles ou structurelles.

Bien que les séquences moléculaires comparées en bio-informatique et les motifs comparés en analyse musicale n'aient aucun point de comparaison sur leur nature, l'analogie entre les deux domaines apparaît lorsque l'on considère la pertinence de leur représentation séquentielle et de leur structuration répétitive. L'aspect séquentiel en bio-informatique est lié à la configuration spatiale des molécules, tandis que l'aspect séquentiel en musique est lié à la configuration temporelle des événements musicaux. La répétition approchée, quant à elle, est liée à des phénomènes de transcription et de mutation mis en jeu dans les processus biologiques, alors qu'elle est utilisée en musique pour correspondre à l'organisation implicite par ressemblance de notre perception musicale [LJ96, Bre94]. Par conséquent, les méthodes développées pour la comparaison approchée de séquences biologiques ont un grand intérêt pour l'analyse des structures répétitives dans les séquences musicales. Les algorithmes

adaptés de comparaison de séquences sont décrits précisément dans le Chapitre 3.

### 1.3.2 Efficacité calculatoire

L'approche basée sur le contenu musical demande d'analyser automatiquement des informations musicales de haut niveau à partir du signal. Elle implique alors une capacité calculatoire importante, qui peut s'avérer problématique pour certains traitements complexes sur de grandes bases de données.

La *requête par fredonnement*, ou *query by humming*, par exemple, est une application consistant à retrouver automatiquement un morceau de musique dans une base de données à partir d'un extrait chanté, sifflé ou fredonné de ce morceau [DBP+07]. D'importantes variations peuvent exister entre le morceau original et l'extrait fourni : des différences dans la hauteur des notes chantées, dans le rythme ou même des erreurs de chant ajoutant des notes ou en supprimant. La prise en compte d'un tel faisceau d'altérations possibles a un coût calculatoire élevé, l'algorithme d'analyse devant alors travailler sur de nombreux paramètres interdépendants. Or, le *passage à l'échelle* des méthodes d'analyse, c'est-à-dire l'application d'une telle requête sur une base de données importante, de l'ordre de milliers voire de millions de titres, représente un enjeu crucial pour une utilisation pratique sur le matériel audio disponible à l'heure actuelle.

Pour cette raison, l'efficacité calculatoire est discutée pour plusieurs des méthodes de recherche d'informations basée sur le contenu présentées dans cette thèse. Sur une perspective à long terme, nos méthodes ont pour objectif d'être utilisées sur un grand nombre de données. Plusieurs études d'optimisation d'algorithmes ou d'améliorations basées sur des heuristiques proposées dans les Chapitres 3 et 4 permettent ainsi d'améliorer sensiblement le temps d'analyse des informations musicales.

Outre la multiplicité des données à traiter, le *calcul en temps réel* est également une contrainte calculatoire susceptible d'être examinée dans le cadre de certaines méthodes d'analyse. Dans ce cas, on impose que l'analyse des données soit effectuée dans des contraintes temporelles strictes, correspondant par exemple au temps de lecture du matériel audio. Un exemple de tel système, proposé par Cont [Con08] sous le nom d'*Antescofo*[1], permet à un musicien de bénéficier d'un accompagnement orchestral synchronisé sur son propre jeu. Ainsi, à chaque instant, le système permet d'identifier les notes jouées par le musicien, de repérer la position du jeu dans un morceau et d'adapter l'accompagnement de manière immédiate.

1. http://repmus.ircam.fr/antescofo

Chapitre 2

# Représentation séquentielle de l'information musicale

Les œuvres musicales sont composées d'un ensemble de sons organisés dans le temps. Pour la vaste majorité des études musicologiques [Tem04, Eri75] ou cognitives [LJ96, Hur06, Kru01, Eme98, Bre94] traitant de la structuration musicale, la perception des entités sonores de la musique est intrinsèquement liée à son déroulement temporel. Par exemple, Tenney et Polansky [TP80] désignent comme *« les facteurs primaires de cohésion »* de la perception musicale les notions de *proximité temporelle* et de *simultanéité* des éléments musicaux. Afin d'analyser automatiquement à partir du signal audio des structures musicales, comme les répétitions approchées, il est donc primordial de représenter l'information musicale en respectant l'ordre temporel inhérent à la musique [CVG+08]. Comme introduit dans le chapitre précédent, la modélisation *séquentielle* de l'information est une technique d'analyse efficace et souvent employée afin d'estimer les similarités dans la musique en respectant cette temporalité [CVG+08].

Dans ce chapitre, nous décrivons une méthode de représentation d'un signal audio sous la forme d'une séquence de symboles comparables décrivant une information musicale. L'objectif du modèle exposé ici est de capturer l'évolution d'une information musicale riche afin de permettre une analyse ultérieure des structures répétitives perçues. Par conséquent, chaque symbole, ou *descripteur*, doit représenter une information liée à un aspect de la perception primordial dans la répétition musicale. La Section 2.1 justifie le choix d'un critère de représentation de l'information musicale, puis la Section 2.2 détaille une méthode permettant de représenter ce critère, et ainsi d'estimer une information musicale pertinente sur une portion de signal. La Section 2.3 présente ensuite l'obtention de la séquence décrivant tout le signal audio, et définit une mesure de similarité appropriée permettant de comparer des éléments de cette séquence.

## 2.1 Critères de description musicale

En raison de la complexité et de la diversité de la donnée musicale, l'analyse automatique du signal audio musical est liée à de nombreux critères. Cette section présente brièvement les trois principales familles de critères de description musicale et justifie les choix de représentation effectués dans cette thèse.

### 2.1.1 Tempo, rythme et métrique

Les aspects musicaux du tempo, de la pulsation et du rythme jouent un rôle fondamental dans la perception et l'interaction avec la musique [MEKR11, Par94].

La *pulsation* peut être décrite comme un accent qui intervient de manière cyclique au début de chaque temps. Sa perception est intuitive pour l'oreille humaine,

entraînée ou non [LJ96], et se traduit par exemple par un battement de pied accompagnant l'écoute d'un morceau. Le terme *tempo* désigne la vitesse de référence de la pulsation d'un morceau et se mesure en pulsations par minute (ou *beat per minute, BPM*). En fonction du style et de la volonté de composition, les valeurs habituelles de tempo se situent entre 40 et 260 pulsations par minute [Jeh05a, p.55–57].

En musique populaire occidentale, les notes sont organisées dans le temps autour de la pulsation. Le placement de ces notes en fonction de la celle-ci, communément appelé *rythme*, conduit à la perception de structures rythmiques. La présence de motifs réguliers (percussions, ligne de basse *etc.*) permet une distinction intuitive entre des pulsations fortes (ou temps forts) et faibles. De tels motifs induisent une structure rythmique récurrente, appelée *métrique*. Cette métrique est fondamentale pour l'interprétation musicale, permettant notamment aux musiciens de synchroniser leur jeu. La structure métrique est hiérarchique. Par exemple, une métrique 4/4 divise un morceau en mesures de 4 temps, constituant le niveau du temps musical. Le premier et le troisième temps sont généralement perçus comme plus forts que les deuxième et quatrième, et forment un autre niveau métrique. Enfin, le premier temps est perçu comme plus fort que le troisième, et constitue également un autre niveau métrique.

Comme souligné par Lerdahl et Jackendoff [LJ96], d'autres structures non nécessairement liées à la métrique sont fondamentales dans la perception humaine du rythme. Basée sur des éléments cognitifs, leur théorie souligne que notre cerveau effectue naturellement un découpage des événements selon des groupes en se basant sur de nombreux critères liés aux éléments percussifs, à la dynamique, à la hauteur des notes ou encore au timbre du son. La difficulté d'un tel découpage dépend de la correspondance entre l'organisation des événements dans la musique, d'une part, et des intuitions et connaissances personnelles pour le regroupement de ceux-ci, d'autre part [LJ96]. Par conséquent, l'analyse des informations rythmiques est une opération complexe et liée à de nombreux autres paramètres musicaux.

### 2.1.2 Timbre et instrumentation

La notion de *timbre* est fortement dépendante du contexte d'application. Le timbre est défini par [Ter60] comme *« l'attribut de la sensation auditive qui permet à un auditeur de juger comme dissimilaires deux sons présentés dans les mêmes conditions et possédant la même dynamique sonore et la même hauteur »*. Cet énoncé, qui définit le timbre par ce qu'il n'est pas, est fortement lié à la notion de reconnaissance de sources sonores et d'instruments [Auc06]. Par exemple, la distinction entre le son d'un piano et d'une guitare jouant la même note au même volume peut aisément être effectuée par n'importe quel auditeur humain, quelle que soit son expérience musicale, grâce à la différence de timbre perçue. De plus, le musicien à l'oreille expérimentée peut être capable d'identifier chacun des instruments intervenant dans un mélange musical, et ainsi d'isoler chaque source musicale à partir de ses caractéristiques timbrales. Cette perception "analytique" des caractéristiques timbrales est complétée par une perception plus globale, dans laquelle l'oreille considère le signal audionumérique comme un ensemble cohérent et homogène. Le *timbre polyphonique*, également appelé timbre en cas de non ambiguïté, correspond au mélange timbral d'un signal musical [MEKR11]. Cette fois, le timbre correspond à une perception générale de l'instrumentation d'un morceau de musique plutôt qu'à

une unique source. Cette notion du timbre est fréquemment utilisée pour déduire une texture globale sur un morceau de musique dans un but d'identification du style musical, de l'émotion ressentie à l'écoute, ou encore pour la reconnaissance automatique d'étiquettes, ou *tags* [MEKR11].

L'espace de description du timbre est intrinsèquement multidimensionnel. Contrairement à la sensation de hauteur tonale ou à la sensation de volume sonore, la perception du timbre est liée à de nombreux facteurs acoustiques [Auc06]. Par conséquent, la mesure du timbre dans le signal audionumérique s'avère complexe à représenter sur un espace continu de description des instruments ou des textures sonores.

### 2.1.3 Information tonale

La perception humaine de la musique est fortement liée à la notion de *hauteur* et de *ton musical*.

La plupart des instruments de musique, incluant les instruments à cordes pincées (guitare, harpe...), frappées (piano...), frottées (violon, contrebasse...), les instruments à vent (bois, cuivres) ou encore la voix humaine émettent des sons à une ou plusieurs *hauteurs* caractéristiques [KD06]. En théorie musicale, la hauteur d'un son est identifiée par une *note*, qui définit un niveau standard, du plus grave au plus aigu. Cet aspect *tonal* est essentiel dans la musique occidentale [GÓ6]. Bien que la dénomination explicite de notes ou d'une gamme ne semble pertinent que pour les utilisateurs exercés à la musique, la perception des tons musicaux, implicite, est naturellement développée chez toute personne, musicienne ou non [TB02]. De plus, la hauteur est une notion indépendante du timbre musical : deux instruments de timbres très distincts peuvent jouer des notes facilement identifiables comme ayant la même hauteur.

La *mélodie* correspond à une succession temporelle de hauteurs [KD06]. La mélodie est monophonique, c'est-à-dire que toute mélodie ne comporte qu'une note à un instant donné. Dans la culture occidentale, la mélodie est généralement perçue comme le critère musical le plus caractéristique d'un morceau de musique : elle est souvent facile à identifier, et c'est en fredonnant sa mélodie que l'on peut se référer à une œuvre musicale.

La combinaison de plusieurs hauteurs distinctes à un même instant est appelée *accord* [KD06]. La notion d'*harmonie* fait référence à la formation des accords et aux relations entre eux [KD06]. Plus précisément, le terme fait référence à la fois aux hauteurs et accords en jeu, mais également aux principes structurels qui régissent leur combinaison [GÓ6, p.15]. Distinguer la mélodie et l'harmonie dans un morceau de musique occidentale est une tâche complexe, car les deux concepts font référence à une combinaison de hauteurs et dépendent l'un de l'autre [Kru04].

La notion d'harmonie est fondamentale dans la culture musicale occidentale. Pendant plusieurs siècles, les compositeurs et théoriciens de la musique ont codifié les simultanéités les plus courantes en systèmes exhaustifs de représentation [Dow03]. Selon Temperley [Tem04, p.117], le jeu de l'harmonie dans un morceau donne à la musique *« une impression de dimension spatiale et de mouvement dans l'espace qui représente une partie indispensable de l'expérience musicale, et contribue grandement au pouvoir expressif et dramatique de la musique tonale »*. La très grande majorité des œuvres de musique occidentale est composée autour de sys-

tèmes harmoniques obéissant à des règles de composition musicale qui caractérisent les progressions d'accords de chaque morceau. Ainsi, les processus de répétitions mélodiques et harmoniques sont au centre de la perception d'une structuration de la musique [LJ96, Hur06, Ste79]. L'harmonie en musique occidentale encode ainsi une structure riche et bien-fondée d'un point de vue musical théorique.

Les algorithmes d'analyse présentés dans cette thèse sont axés sur une description tonale de la musique. Pour résumer, trois raisons principales motivent ce choix de l'information tonale :

- Elle correspond à un critère perceptif prépondérant pour la musique ;
- Elle encode des structures musicales riches ;
- Elle est soutenue par une théorie musicale bien connue et étudiée.

## 2.2 Représentation tonale du signal

Nous décrivons dans cette section un modèle permettant de représenter l'information tonale d'un morceau de musique à partir de son signal audio.

### 2.2.1 Signal audio

D'un point de vue physique, un signal audio est généré par une entité vibrante telle que les cordes vocales d'un chanteur, la corde d'un violon ou encore la membrane d'une enceinte. Associées au reste de l'instrument, ces entités vibrantes engendrent des déplacements et oscillations des particules d'air, résultant en un phénomène ondulatoire, qui se propage dans l'air et se mesure par une variation de la *pression acoustique*. L'onde sonore qui arrive à l'oreille humaine provoque alors la vibration du tympan qui, avec l'aide de l'oreille interne, transforme l'onde perçue en signal nerveux pour interprétation par le cerveau.

À l'instant où elle arrive à l'oreille humaine, l'onde sonore peut être représentée par la variation de la pression acoustique en fonction du temps, représentation également appelée *forme d'onde* du signal audio. La Figure 2.1-haut montre un exemple de forme d'onde correspondant à 7 secondes d'un morceau joué à la clarinette. Les variations de pression acoustique dessinent des zones montantes ou descendantes, indiquant grossièrement les modifications d'*amplitude* du signal.

Mathématiquement, une forme d'onde est caractérisée par une fonction continue, définie en tout point du temps. Lorsque la variation de pression acoustique se fait sous forme de motifs réguliers, la forme d'onde est dite *périodique*. La *fréquence* $f$ (en Hertz) est alors définie comme le nombre de répétitions d'un motif par seconde, et la *période* $T$ comme l'inverse de cette fréquence : $T = 1/f$. Un exemple simple de signal audio, le *son pur*, est caractérisé par la formule :

$$x(t) = a \cdot \sin(2\pi f t + \Phi), \tag{1}$$

où $a$ désigne l'amplitude du signal, $f$ sa fréquence et $\Phi$ la phase initiale du signal. La Figure 2.1-bas représente un tel signal sur une seconde, pour une amplitude fixe de 1, une phase initiale nulle et une fréquence de 4 Hz.

La plupart des signaux audibles dans la musique occidentale sont des formes d'ondes beaucoup plus complexes. En particulier, les sons qualifiés d'harmoniques sont en général composés de sons purs de fréquences dites *harmoniques*, multiples

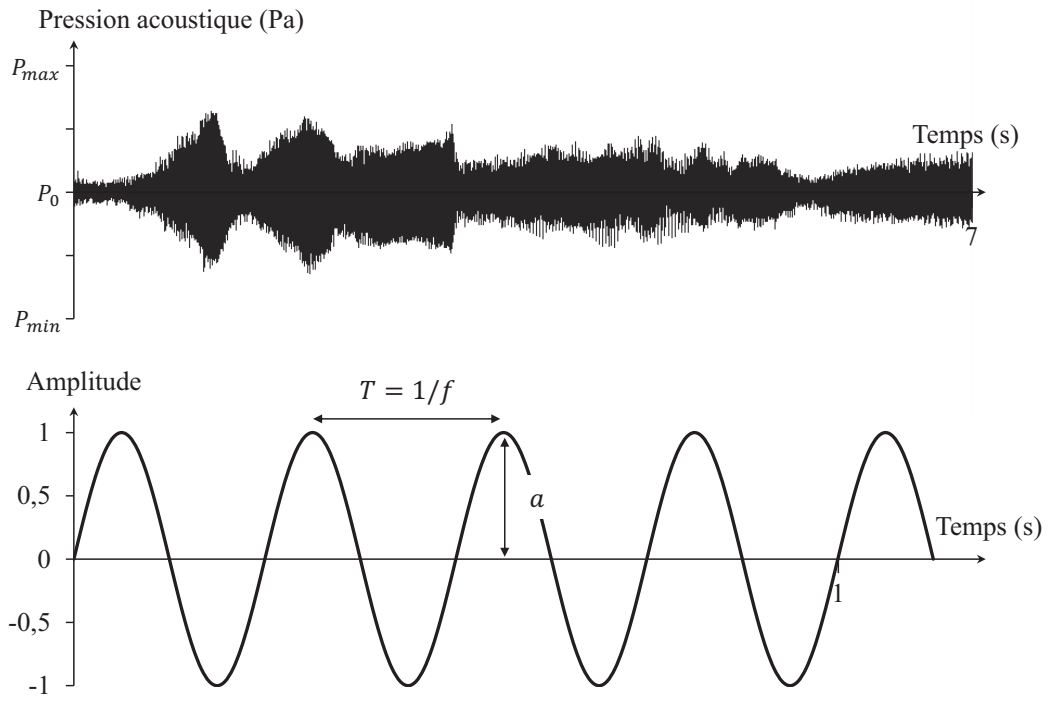


FIG. 2.1 – *Haut : Forme d'onde d'un extrait audio. Bas : Son pur d'amplitude 1, de phase initiale nulle et de fréquence 4 Hz.*

d'une fréquence $f_0$, appelée *fréquence fondamentale*. Les sons naturels, tels que produits par exemple par un instrument de musique ou une voix humaine, correspondent à des signaux plus complexes, souvent formés de composantes harmoniques et de composantes non périodiques. En outre, les paramètres des composantes périodiques de tels signaux sont susceptibles d'évoluer dans le temps, les rendant potentiellement non-stationnaires.

### 2.2.2 Représentation numérique

Le signal sonore peut être représenté sous la forme d'une fonction mathématique continue, définie à chaque instant. Cependant, dans le cadre d'une analyse informatique du son, il est nécessaire de représenter numériquement le signal audio afin de permettre sa manipulation par l'unité logique d'un ordinateur, d'un microcontrôleur ou de tout autre calculateur numérique. Cette opération, appelée *numérisation*, est réalisée par un composant appelé *convertisseur analogique-numérique*, CAN (ou *analog-to-digital converter*, ADC). La numérisation consiste en deux étapes successives : l'*échantillonnage* et la *quantification*. L'échantillonnage est une discrétisation de la dimension temporelle du signal, et correspond généralement à l'enregistrement d'un échantillon du signal continu à intervalles de temps réguliers. Il est caractérisé par une *fréquence d'échantillonnage*, notée $F_e$. La quantification correspond à l'opération de discrétisation de la valeur d'amplitude du signal pour un échantillon donné. Cette quantification est généralement effectuée sur un nombre fini $q$ de bits, qui limitent la résolution du signal.

Une telle méthode de codage a notamment été mise au point dans le cadre

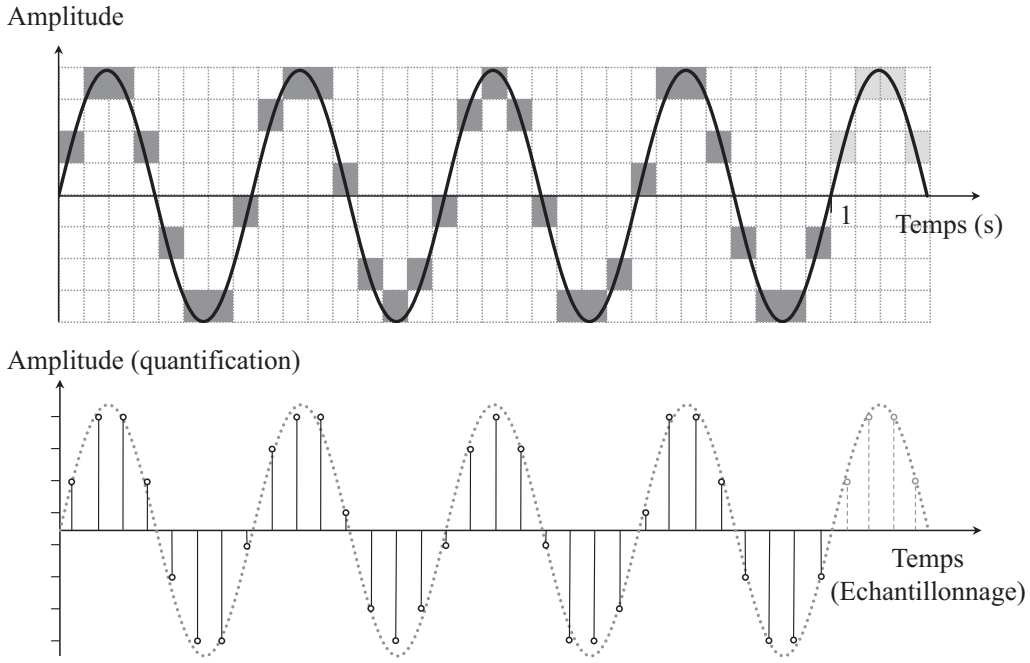


FIG. 2.2 – *Numérisation d'une seconde d'un signal audio. Haut : Le signal continu est représenté en traits pleins. Le quadrillage pointillé représente l'espace des valeurs numériques qui peuvent être codées, pour une fréquence d'échantillonnage* $F_e = 31Hz$ *et une quantification sur* 8 *valeurs possibles. Les cases grisées montrent les échantillons représentant le signal sous ces conditions. Bas : Signal numérisé représenté sous la forme d'un peigne de valeurs discrètes. La représentation du signal original en pointillés souligne l'erreur commise par les opérations d'échantillonnage et de quantification.*

d'une technique appelée *Pulse Code Modulation* (PCM) [BE47]. Bien que d'autres techniques de codage, telles que *Direct Stream Digital* (DSD) [RN01] proposent une représentation différente, PCM reste aujourd'hui la technique de codage audio la plus largement utilisée. Par exemple, le format CD audio comprend des données enregistrées en codage PCM à la fréquence d'échantillonnage de 44100 Hz et selon une quantification sur 16 bits. Ainsi, 44100 mesures définissent le signal à chaque seconde, chacune étant égale à l'une des 65536 valeurs possibles pour l'amplitude du signal. En combinant les opérations d'échantillonnage et de quantification, le signal continu est ainsi représenté par un ensemble fini de valeurs, matérialisé par un quadrillage dans l'exemple de la Figure 2.2-haut. La seconde figure présente le signal discret obtenu sous la forme d'un peigne de valeurs discrètes.

Mathématiquement, l'opération de numérisation est une discrétisation du signal faisant apparaître deux paramètres :

- $F_e$, la fréquence d'échantillonnage qui définit la résolution temporelle ;
- $q$, le nombre de bits utilisés pour la quantification, qui définit la résolution d'amplitude.

Bien que les paramètres $F_e$ et $q$ soient généralement définis comme des constantes, ils peuvent prendre des valeurs variables dans le cadre de systèmes de codage plus

perfectionnés. Par commodité, on supposera dans la suite ces deux paramètres constants.

Il convient de noter que l'opération de numérisation peut être destructrice d'information, et qu'il n'est pas forcément possible de reconstruire la forme d'onde initiale à partir de la représentation numérique [Mül07, p.24]. Les erreurs introduites à la reconstruction sont appelées *effets d'aliasing*, ou encore *erreurs de quantification*, et peuvent introduire des effets audibles dans le signal reconstruit, comme des claquements ou des bruits indésirables. Pour cette raison, le choix des paramètres de numérisation est capital pour une représentation numérique fidèle du signal audio. En particulier, le théorème de Shannon-Nyquist indique que la fréquence d'échantillonnage $F_e$ doit être au moins deux fois supérieure à la fréquence maximale composant un signal afin de représenter numériquement celui-ci de manière correcte et de permettre une reconstruction correcte du signal continu.

Le son pur décrit dans le domaine continu par l'équation 1 est caractérisé sous sa forme numérisée par la formule :

$$x[n] = \lfloor 2^{q-1} \cdot a \cdot \sin(2\pi f \frac{n}{F_e} + \Phi) \rfloor, \tag{2}$$

où $a$ désigne l'amplitude du signal, $f$ sa fréquence, $n$ le numéro d'échantillon numérique considéré, $\Phi$ la phase initiale du signal, $F_e$ la fréquence d'échantillonnage et $q$ le nombre de bits de quantification.

### 2.2.3 Relation hauteur/fréquence

Afin d'identifier les caractéristiques tonales d'un signal audio, il convient de caractériser mathématiquement la notion de hauteur du son. La hauteur d'un son pur est liée à sa fréquence : plus sa fréquence est faible, plus il est perçu comme grave ; à l'inverse, plus sa fréquence est élevée, plus il est perçu comme aigu. La hauteur perçue pour un son harmonique est fortement liée à sa fréquence fondamentale. L'écart perçu entre deux hauteurs est appelé *intervalle*. Plus précisément, la hauteur est perçue de manière logarithmique par rapport à la fréquence des signaux [Kru01] ; par exemple, la variation de hauteur perçue entre deux signaux purs de fréquences 220 Hz et 440 Hz est identique à celle perçue entre deux signaux de fréquences 880 Hz et 1760 Hz. L'intervalle entre un son de fréquence $f$ et un son de fréquence $2 \times f$ est appelé *octave*. La musique occidentale divise chaque octave en un ensemble de 12 intervalles, appelés *demi-tons*, le *ton* musical correspondant à un intervalle de deux demi-tons. Ainsi, 12 notes sont définies pour chaque octave, l'ensemble formant la *gamme chromatique*. La répartition des différents intervalles sur une octave est appelée *tempérament*. Ainsi, le *tempérament égal*, le plus utilisé à l'heure actuelle en musique occidentale, divise chaque octave en 12 demi-tons de ratio égal. Plus précisément, deux fréquences $f_1$ et $f_2$ sont séparées de $n$ demi-tons si et seulement si :

$$\frac{f_2}{f_1} = 2^{\frac{n}{12}}. \tag{3}$$

Par conséquent, l'intervalle $I(f_1,f_2)$ en demi-tons entre deux fréquences $f_1$ et $f_2$ est défini par :

$$I(f_1,f_2) = 12 \cdot \log_2(\frac{f_2}{f_1}). \tag{4}$$

| Nom français | Nom anglo-saxon | Fréquence (Hz - Référence La4-440Hz) |
|---|---|---|
| Do | C | 261.63 |
| Do♯/Ré♭ | C♯/D♭ | 277.18 |
| Ré | D | 293.66 |
| Ré♯/Mi♭ | D♯/E♭ | 311.13 |
| Mi | E | 329.63 |
| Fa | F | 349.23 |
| Fa♯/Sol♭ | F♯/G♭ | 369.99 |
| Sol | G | 392.00 |
| Sol♯/La♭ | G♯/A♭ | 415.30 |
| La | A | 440.00 |
| La♯/Si♭ | A♯/B♭ | 466.16 |
| Si | B | 493.88 |

TAB. 2.1 – *Dénominations usuelles des notes de la gamme tempérée, et correspondances fréquentielles sur la quatrième octave du piano, avec pour référence la note La4 à 440Hz. La notation "/" indique que les deux dénominations de part et d'autre du symbole peuvent être utilisées, le choix dépendant de règles de notation musicale.*

Le Tableau 2.1 précise la dénomination des notes de la gamme tempérée ainsi que les valeurs fréquentielles standard sur une octave.

Dans la culture occidentale, la perception de hauteur est fortement liée à la gamme chromatique. Ainsi, une légère variation de la fréquence d'une forme d'onde par rapport à une fréquence standard ne provoque pas de changement dans la perception de hauteur. Par exemple, en considérant 440Hz comme fréquence de référence pour le cinquième La du piano, un son pur de fréquence 443 Hz est intuitivement perçu comme la note La, malgré cette légère variation. Pour la plupart des auditeurs, cette variation par rapport à la fréquence de référence est inaudible ; mais si les oreilles les plus exercées sont capables de la détecter, la note de La associée reste naturellement identifiée. Il est donc commun d'associer à une note de la gamme tempérée une plage de fréquences, centrée sur la fréquence de référence.

### 2.2.4 Analyse spectrale

La perception musicale d'un signal audio est fortement corrélée aux fréquences des composants de celui-ci. La transformation de Fourier est une technique très répandue permettant d'analyser de manière précise le contenu fréquentiel du signal.

#### 2.2.4.1 Transformée de Fourier Discrète

Le théorème de Fourier indique que toute fonction périodique peut être représentée comme la somme de composantes sinusoïdales d'amplitudes, de phases et de fréquences déterminées, ces dernières étant en relation harmonique. La *Transformée de Fourier* permet ainsi d'estimer les composantes fréquentielles d'un signal audio. En particulier, la *Transformée de Fourier Discrète* (TFD)[SPA07, p.22–23] $X(k)$

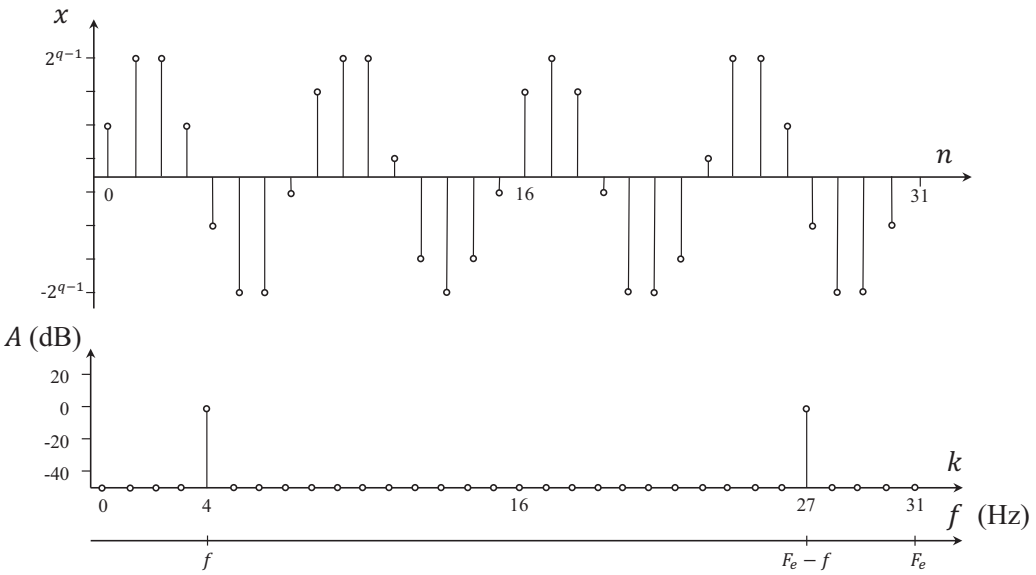


FIG. 2.3 – *Haut : une seconde d'un son pur d'amplitude* 1*, de phase initiale nulle et de fréquence* 4*Hz, numérisé à la fréquence d'échantillonnage 31Hz. Bas : spectre d'amplitude correspondant tel que calculé par transformée de Fourier discrète.*

d'un signal numérique $x[n]$ fini est donnée par la formule :

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \exp^{-j \cdot 2 \cdot \pi \cdot n \cdot k/N} \text{ pour } k \in [\![0,N-1]\!], \tag{5}$$

où $N$ correspond au nombre d'échantillons du signal numérique à analyser.

La complexité temporelle de l'algorithme de calcul de la TFD est quadratique : $\mathcal{O}(N^2)$. Cependant, le calcul pratique de la TFD peut être accéléré par des méthodes optimisant le nombre d'opérations effectuées, comme l'algorithme de Cooley et Tukey [CT65]. Ces techniques de calcul optimisé de la TFD sont appelées *Transformées de Fourier Rapide*, ou FFT. La complexité algorithmique de la FFT est linéarithmique : $\mathcal{O}(N \log N)$, où $N$ est le nombre d'échantillons à analyser [GR98, p.34–38].

#### 2.2.4.2 Spectre d'amplitude et spectre de phase

Chaque valeur calculée par transformation de Fourier est un nombre complexe pouvant s'écrire sous la forme $X[k] = X_r[k] + j \cdot X_i[k]$. Le *spectre d'amplitude* $\mathcal{A}$ du signal correspond au module de sa transformée, soit :

$$\mathcal{A}[k] = |X[k]| = \sqrt{X_r[k]^2 + X_i[k]^2} \text{ pour } k \in [\![0,N-1]\!]. \tag{6}$$

Le *spectre de phase* $\varphi$ du signal correspond à l'argument de sa transformée, soit :

$$\varphi[k] = arg(X(k)) = \arctan \frac{X_i[k]}{X_r[k]} \text{ pour } k \in [\![0,N-1]\!]. \tag{7}$$

Le spectre d'amplitude indique la répartition des composantes fréquentielles du signal analysé. La Figure 2.3-bas présente le spectre d'amplitude du son pur

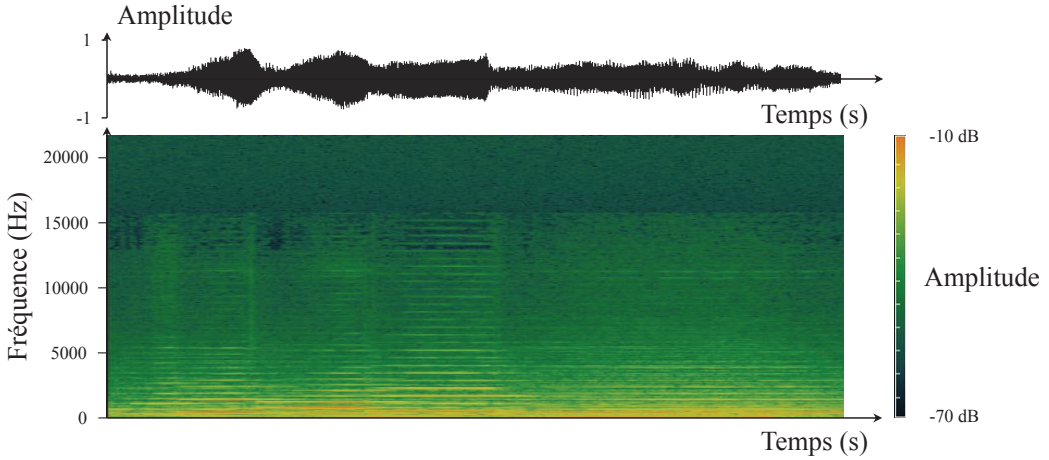


FIG. 2.4 – *Exemple de spectrogramme d'un enregistrement audio.*

représenté au-dessus ($f = 4$Hz, $\phi = 0$, $a = 1$, $F_e = 31$Hz). La FFT des $N$ échantillons de signal fournit $N$ valeurs spectrales. Les indices des composantes du spectre d'amplitude peuvent être représentés sur un axe des abscisses en valeurs fréquentielles avec un pas de $\frac{F_e}{N}$ Hz entre deux points du spectre. En d'autres termes, pour $k \in [\![0,N-1]\!]$, la composante $X[k]$ correspond à une valeur de fréquence de $k\frac{F_e}{N}$. Le spectre d'amplitude du signal présente ainsi une composante non nulle pour la valeur $f = 4$Hz, qui correspond à la fréquence du signal initial. La composante non nulle visible à la valeur $F_e - f$ est due à l'opération de numérisation, qui entraîne une réplique du spectre au-delà de la fréquence $\frac{F_e}{2}$. Le spectre visible entre les fréquences $\frac{F_e}{2}$ et $F_e$ correspond alors à une version renversée du spectre entre 0 et $\frac{F_e}{2}$ ; en d'autres termes, la droite d'équation $x = \frac{F_e}{2}$ est un axe de symétrie du spectre d'amplitude obtenu. Or, sous l'hypothèse que la condition de Shannon-Nyquist est respectée (voir Section 2.2.2), les composantes fréquentielles visibles au-delà de $\frac{F_e}{2}$ ne correspondent pas à des composantes présentes dans le signal. Il est donc possible d'appliquer un filtre sur les fréquences non pertinentes, et ainsi de retirer cette duplication pour ne conserver que la section $[0, \frac{F_e}{2}]$.

Afin de correspondre à des propriétés de la perception humaine, l'amplitude est classiquement donnée en *décibels* (dB) sur une échelle logarithmique donnée par $\mathcal{A}_{dB}[k] = 20\log_{10}(\frac{2\cdot\mathcal{A}[k]}{N})$. Dans cette formule, le facteur $\frac{2}{N}$ permet de normaliser le résultat de la FFT afin de faire correspondre dans le spectre une amplitude de 0 décibels à la fréquence fondamentale d'un son pur (voir Figure 2.3-Bas).

#### 2.2.4.3 Spectrogramme

La transformée de Fourier rapide permet d'obtenir la répartition des composantes fréquentielles sur une portion de signal, en renvoyant une représentation fréquence/amplitude de l'extrait analysé. Le *spectrogramme* est une visualisation temps/fréquence/amplitude qui permet de représenter l'évolution des composantes fréquentielles du signal [Hay95]. Il permet ainsi de visualiser l'évolution du contenu fréquentiel sur plusieurs portions de signal. La Figure 2.4 montre un exemple de spectrogramme obtenu par TFD sur des portions d'environ 23 ms. Le temps est in-

diqué en abscisse, de gauche à droite, tandis que le contenu fréquentiel est indiqué en ordonnée, de bas (basses fréquences) en haut (hautes fréquences). La valeur de l'amplitude des composantes fréquentielles est représentée sur une palette de couleurs, sur une mesure en décibels. Les lignes jaunes parallèles mettent en avant les composantes harmoniques qui composent l'extrait audio.

### 2.2.5 Définition du chroma

Plusieurs types de descripteurs peuvent être utilisés pour représenter l'information tonale d'un signal audio. Ainsi, la *centroïde tonale* [HSG06], les *Profils Q-constants* [Pur05, p.120–123] ou encore les *Profils de Classes de Hauteurs* [Fuj99] sont différentes représentations symboliques possibles de l'information tonale. Une revue détaillée des techniques existantes peut être consultée en [GÓ6, p.35–62].

La représentation par spectre d'amplitude exposée en Section 2.2.4.2 donne accès au contenu fréquentiel de la portion de signal analysée. Le spectre d'amplitude fournit donc une information sur l'ensemble des hauteurs perçues dans l'extrait analysé. Afin de caractériser précisément les informations tonales de l'extrait, il convient alors de projeter ce spectre sur un espace de description des hauteurs musicales. Comme souligné par Shepard [She82], les hauteurs sont perçues de manière cyclique, où un cycle recommence à chaque octave. Dès lors, la hauteur d'un son peut être décomposée par la hauteur de l'octave à laquelle elle appartient, et la classe de la note au sein d'une octave, ou *classe tonale*. Particulièrement développée chez les musiciens entraînés, cette perception par classes tonales est réputée présente chez tout auditeur habitué à la musique occidentale [Deu82, Kru04].

Le *Profil de Classes Tonales* [Fuj99], ou *Chroma* [BW01], propose une représentation des classes tonales perçues dans un extrait audio. Le chroma peut être défini comme un vecteur de nombres réels dont chaque coefficient indique la prépondérance de l'une des divisions de la gamme tempérée dans l'extrait analysé.

La Figure 2.5 présente plusieurs exemples de chromas correspondant à différentes informations tonales. On suppose que ceux-ci caractérisent des extraits musicaux à l'échelle des accords. Ces chromas sont calculés avec une dimension égale au nombre de demi-tons de la gamme chromatique (12) ; par conséquent, chaque coefficient indique la prépondérance d'un demi-ton dans l'extrait. Les trois valeurs les plus élevées correspondant au La, au Do et au Mi dans (i) et (ii) soulignent la prépondérance d'un accord particulier dans l'extrait analysé (accord de La mineur [Bit87]). L'absence de coefficient prépondérant dans (iii) suggère que l'extrait sélectionné ne comporte pas d'harmonie facilement identifiable. Enfin, l'apparition de nouveaux coefficients importants dans (iv) peut correspondre, par exemple, à la présence d'une harmonie complexe à 5 notes (Accord de neuvième de dominante de la gamme de Sol [Bit87]), à la juxtaposition de deux accords (Ré majeur et La mineur [Bit87]) ou encore à une simple série de 5 notes jouées successivement dans l'extrait analysé.

La *résolution* du chroma correspond au nombre de dimensions, ou de valeurs contenues dans le vecteur. Correspondant en Figure 2.5 au nombre de demi-tons présents dans la gamme chromatique, cette résolution peut être ajustée en fonction de l'application afin de régler la robustesse aux problèmes d'accord et aux légères oscillations de fréquence susceptibles d'apparaître dan les signaux acoustiques [GÓ6, p.49]. Ainsi, le choix d'une résolution plus élevée permet par exemple une forte amé-

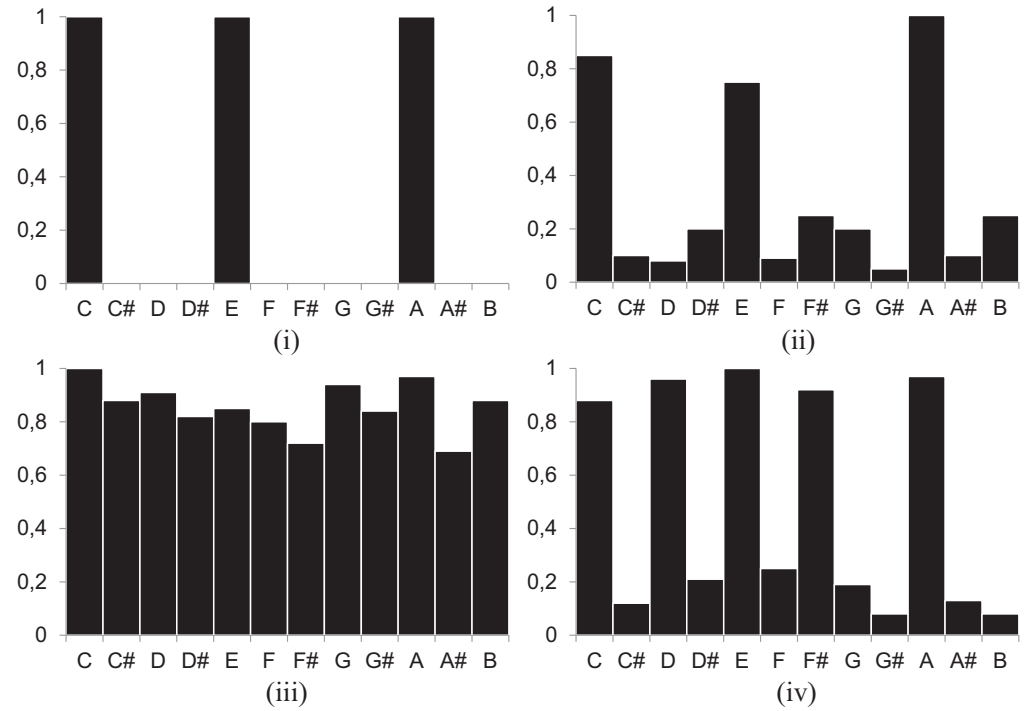


FIG. 2.5 – *Exemples de chromas de dimension 12. (i) : Triade en La mineur, cas idéal. (ii) : Triade en La mineur en présence de bruit harmonique. (iii) : Pas d'harmonie facilement identifiable. (iv) : Superposition de tons dans le chroma, ou harmonie complexe.*

lioration de la qualité de la description harmonique sur un morceau entièrement ou partiellement désaccordé pour l'identification des accords [PBO00, Góm06]. Cette résolution peut facilement être réduite au nombre de demi-tons dans le cas de comparaison avec un modèle tonal, ou par commodité de visualisation. Dans la suite, la résolution du chroma sera abstraite à un entier $B$, classiquement multiple de 12.

Si la nature de l'information contenue dans le chroma reste identique, sa définition mathématique diffère légèrement selon les études. Fujishima [Fuj99] puis, plus tard, Gomez [GÓ6] introduisent le descripteur à partir de fonctions de pondération du spectre d'amplitude obtenu par transformée de Fourier discrète. La définition par banques de filtres [Pee06, Got06], à travers la fréquence instantanée dérivée de la phase [EP07] ou encore à travers une transformée Q-constante remplaçant la transformée de Fourier [ZKG05, BP05, HS05] sont d'autres approches possibles pour construire le chroma. Une comparaison des différentes approches d'obtention du chroma peut être trouvée en [Oud10] ou [GÓ6].

Dans la suite de ce travail, nous utilisons la formulation du chroma telle que proposée par Gómez [GÓ6]. Le choix de ce descripteur est principalement motivé par une performance significativement supérieure aux approches antérieures, relevée par Gómez [GÓ6, p.80–99] en comparant les chromas obtenus selon chacune des approches depuis le signal audio avec les profils de notes obtenus depuis une représentation symbolique (partition).

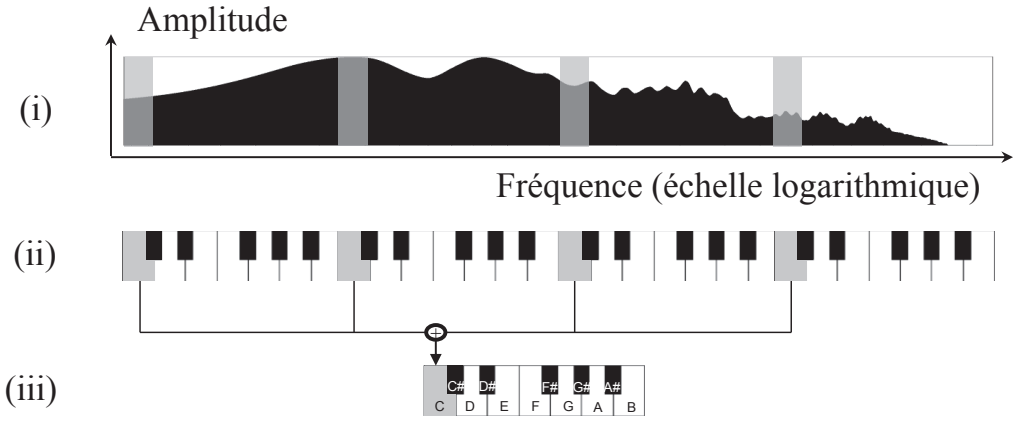


FIG. 2.6 – *Calcul schématique du premier coefficient du chroma de dimension 12. (i) : À partir du spectre d'amplitude d'un signal, chaque bande fréquentielle de la note Do est repérée. (ii) : Cette procédure est effectuée pour toutes les occurrences du Do sur toutes les octaves. (iii) : Le coefficient correspondant dans le chroma est affecté à la somme pondérée des contenus fréquentiels calculés pour les bandes repérées.*

### 2.2.6 Calcul du chroma

Soient $\mathcal{T}$ un signal audio, et $\mathcal{A}[k]$ le spectre d'amplitude de $\mathcal{T}$. On pose $N$ le nombre d'échantillons dans $\mathcal{T}$ ; $\mathcal{A}$ est donc défini pour $k \in [\![0,N]\!]$.

Soit $h = \{h[1],h[2],\ldots,h[B]\}$ le chroma du signal $\mathcal{T}$. Un coefficient $b$ de $h$ est obtenu en sommant toutes les contributions dans le spectre à la note $b$. La Figure 2.6 illustre le principe du calcul de l'un des coefficients du chroma. Sa définition mathématique est décrite ci-dessous.

Soit $b$ une note de la gamme. Pour calculer le coefficient $h[b]$ du chroma correspondant à cette note, il est nécessaire d'identifier dans le spectre d'amplitude toutes les fréquences contribuant à celle-ci.

Soit $f_b$ la fréquence de référence correspondant à la note $b$ de la gamme (par exemple telle qu'indiquée dans le Tableau 2.1). L'intervalle en demi-tons entre $f_b$ et une fréquence quelconque $f$ est $I(f,f_b)$, tel que donné par l'Équation 4. Or, comme décrit en Section 2.2.3, la perception cyclique des hauteurs implique que la note $b$ peut correspondre à une ou plusieurs octaves, donc aux fréquences $f_b$, $2f_b$, $4f_b$, *etc.* Chacune de ces fréquences doit donc être prise en compte pour le calcul du coefficient $b$ du chroma. La taille en demi-tons de l'intervalle entre une fréquence quelconque $f$ et la fréquence de la note $b$ la plus proche est donc donnée par :

$$I_b(f) = 12 \cdot \log_2(\frac{f}{f_b}) \bmod 12, \tag{8}$$

où $x \bmod y$ désigne le reste de la division euclidienne de $x$ par $y$.

Une estimation simple $\hat{h}$ des classes tonales peut alors être calculée en sommant les contributions des différentes occurrences de la note $b$ dans le spectre, selon la

formule :

$$\hat{h}[b] = \sum_{k \text{ t.q. } I_b(f_k)=0} \mathcal{A}[k], \text{ pour } b \in [\![1,B]\!]. \tag{9}$$

Cependant, afin de prendre en compte la perception par plage de fréquences des hauteurs (voir Section 2.2.3) ainsi qu'une possible légère variation des composantes fréquentielles, la contribution dans le coefficient $b$ du chroma est étendue à des bandes de fréquences centrées autour des occurrences de $b$ dans le spectre [GÓ6], représentées par des sections grisées sur la Figure 2.6-(i). Les coefficients ainsi repérés sont pondérés et contribuent ensuite au calcul du coefficient $b$ de $h$. La fenêtre de pondération d'une fréquence $f$ relativement à la note $b$ est donnée par la formule :

$$w(b,f) = \begin{cases} \cos^2(\pi \cdot \frac{I_b(f)}{l}) & \text{si } |I_b(f)| \leq \frac{l}{2} \\ 0 & \text{si } |I_b(f)| > \frac{l}{2} \end{cases}, \tag{10}$$

où $l$ est la largeur de la fenêtre de pondération, paramètre de la méthode. Le réglage de $l$ permet de modifier la taille des bandes fréquentielles considérées pour chaque note.

Le chroma $h$ correspond alors à la somme des contributions des bandes fréquentielles ainsi définies :

$$h[b] = \sum_{k=1}^{N} w(b,f_k) \cdot \mathcal{A}[k]^2 \text{ pour } b \in [\![1,B]\!]. \tag{11}$$

Avec cette formalisation, plusieurs réglages permettent d'adapter le descripteur chroma à l'application considérée.

- La résolution $B$ du chroma. Les valeurs couramment utilisées sont 12 (division de la gamme en demi-tons), 24 (division en quarts de tons) et 36 (division en sixième de tons) [GÓ6]. Les valeurs les plus élevées apportent une plus grande précision des systèmes de comparaison, mais augmentent le temps de calcul de la comparaison de descripteurs [SGHS08].
- La largeur $l$ de la fenêtre de pondération. Gómez [GÓ6] indique un paramétrage empirique à $\frac{2}{3}$ de ton. La modification de ce paramètre permet de faire varier la plage de fréquences correspondant à une hauteur.

### 2.2.7 Traitements possibles

Afin de représenter le plus précisément possible l'information tonale, le calcul du chroma peut compter plusieurs étapes de traitements supplémentaires. En effet, le chroma défini jusqu'ici peut présenter des limitations qui détériorent sa représentativité de l'information tonale. En particulier, la présence de nombreuses composantes harmoniques ou un important problème d'accord peuvent rendre la détection tonale inefficace [GÓ6]. La liste des principales améliorations implémentées est donnée ci-dessous. Nous invitons le lecteur à se référer aux ouvrages cités pour de plus amples détails, ainsi qu'à l'article [MEKR11, p.1091–1092] pour un descriptif général de ces améliorations.

- Réduction de la plage fréquentielle analysée [MEKR11] ;
- Correction de l'accord [EP07] ;
- Prise en compte des fréquences harmoniques [GÓ6, p.77] ;

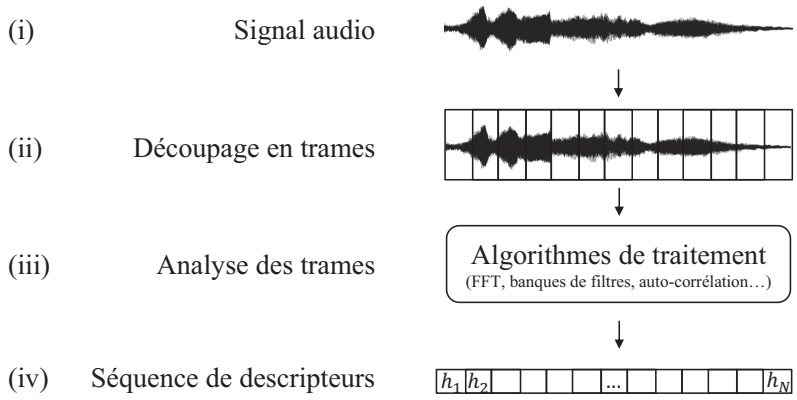


FIG. 2.7 – *Méthode de représentation du signal audio. Le signal (i) est découpé en trames de tailles prédéfinies (ii), chacune étant ensuite analysée (iii) pour produire un descripteur musical. Cette méthode produit une séquence de descripteurs correspondant à l'analyse successive des trames (iv).*

- Blanchiment de spectre et correction de timbre [SR99, ME10] ;
- Normalisation [GÓ6, p.79].

Les chromas utilisés dans les expériences décrites dans ce document ont été calculés implémentant plusieurs de ces améliorations, telles que proposées par Gómez [GÓ6]. L'évaluation de ces chromas a souligné leur robustesse face à des facteurs d'erreurs communs en analyse tonale [GÓ6, p.80–99], tels que la présence de bruit ou de notes désaccordées, le changement d'instrumentation ou encore la modification de la dynamique. En outre, une étude comparative de ce descripteur avec l'état de l'art a mis en avant la qualité de la représentation tonale obtenue pour des applications comme l'analyse de la tonalité [GÓ6] ou l'alignement de séquences [SGHS08].

## 2.3 Représentation séquentielle pour la comparaison

Le descripteur tonal introduit et défini dans la section précédente permet de représenter une information musicale riche depuis une portion de signal audio. Cette analyse peut être combinée à un *traitement par trames* afin d'obtenir une représentation séquentielle de l'information tonale au sein d'un morceau de musique, comme décrit ci-dessous.

### 2.3.1 Traitement par trames

Les algorithmes d'analyse du signal numérique suivent généralement une approche de *traitement par blocs* [KD06, p.68–77]. Cette technique consiste à enregistrer dans un tampon mémoire un nombre prédéterminé d'échantillons du signal audio, puis à analyser l'information reçue à chaque fois que le tampon mémoire est rempli. Cette méthode consiste donc à diviser le signal audio analysé en segments, chaque segment ainsi défini étant appelé *trame*. L'algorithme d'analyse produit alors un descripteur qui caractérise chaque trame selon un paramètre musical, se rattachant aux critères définis en Section 2.1. La Figure 2.7 représente schématiquement le processus d'analyse du signal audio utilisé.

Le choix des durées de trames de signal dépend de multiples facteurs :

- Les limites théoriques des outils mathématiques utilisés, comme la transformée de Fourier décrite en Section 2.2.4.1, imposent des contraintes sur le découpage effectué afin de conserver une analyse de bonne qualité ;
- En outre, la nature du signal analysé, et notamment son caractère stationnaire à l'échelle de temps considérée, induit également des limitations sur le découpage en trames ;
- Enfin, la durée des trames est influencée par l'information musicale à représenter. Par exemple, dans le cas de l'analyse tonale d'un morceau, il convient de choisir une durée de trame supérieure à la note la plus courte du morceau analysé afin de conserver une certaine pertinence musicale. À l'inverse, dans le cas d'une détection des attaques de notes d'un morceau (ou *onsets*), les durées de trames doivent être suffisamment faibles pour permettre l'identification précise des portions de signal correspondant aux débuts de notes, ou *transitoires*.

Selon l'application considérée, les durées des trames peuvent être définies de manière constante ou dépendre d'un paramètre musical, comme les attaques de notes ou le tempo (voir par exemple [EP07, Jeh05a]).

À l'issue de ce traitement, le morceau de musique est représenté par une séquence de descripteurs qui décrivent l'évolution d'un critère musical, les informations tonales dans notre cas.

### 2.3.2 Comparaison entre descripteurs

L'analyse de structures répétitives requiert un mécanisme permettant de comparer les descripteurs entre eux. Il convient donc de définir une mesure permettant d'estimer la ressemblance entre deux chromas. Le choix effectué pour comparer des chromas dans le cadre de ces travaux est issu notamment du travail de Serrà *et al.* [SGHS08], qui met en avant la qualité de ces mesures pour l'estimation de similarités séquentielles par rapport aux autres techniques existantes (voir [SGHS08, p.5] ainsi que l'annexe en ligne [1]).

La mesure utilisée pour comparer deux chromas $h_1$ et $h_2$ de dimension $B$ est le *coefficient de corrélation de Pearson* (voir par exemple [RN88]), défini selon la formule :

$$r(h_1,h_2) = \frac{\sum_{b=1}^{B}(h_1[b]-\overline{h_1})(h_2[b]-\overline{h_2})}{\sqrt{\sum_{b=1}^{B}(h_1[b]-\overline{h_1})^2}\sqrt{\sum_{b=1}^{B}(h_2[b]-\overline{h_2})}}, \tag{12}$$

où $\overline{h_1}$ et $\overline{h_2}$ désignent respectivement la valeur moyenne des coefficients des chromas $h_1$ et $h_2$.

Le coefficient $r$ permet d'estimer la ressemblance entre deux chromas. Cependant, comme souligné par Serrà *et al.* [SGHS08, p.9], il est préférable dans le contexte de comparaisons tonales d'utiliser une mesure de similarité *binaire*, c'est-à-dire prenant une décision de type similaire/dissimilaire. Plusieurs raisons justifient ce choix :

- Comme l'explique Krumhansl [Kru01, p.40–49], il semble erroné de décrire les relations entre accords dans un espace euclidien. Il est donc important de conserver une mesure de similarité non euclidienne ;

1. http://mtg.upf.edu/~jserra/chromabinsimappendix.html, accédé en août 2012

- La définition d'une échelle continue de similarité tonale à partir de considérations musicales et perceptives semble particulièrement complexe à réaliser sur des critères objectifs, et indépendants du style musical considéré [SGHS08] ;
- La réduction à une décision binaire représente un choix simple (similaire ou non similaire) permettant une analogie avec les symboles alphabétiques.

La méthode de comparaison binaire proposée par Serrà *et al.* [SGHS08] consiste à vérifier si deux chromas $h_1$ et $h_2$ correspondent à la même information en comparant $h_1$ avec toutes les *transpositions* possibles de $h_2$.

Pour calculer cette comparaison, il convient donc de définir la notion de *transposition* d'un chroma. Puisque chaque coefficient d'un chroma de dimension 12 correspond à un demi-ton de la gamme tempérée, transposer $h$ de $k$ demi-tons revient à effectuer un décalage circulaire d'ordre $k$ sur les coefficients de $h$. Plus généralement, en notant $B$ la résolution du chroma $h$, on définit la *transposition* de $h$ d'un nombre $k \in [\![1,B]\!]$ de divisions, notée $h^{\uparrow k}$, par la formule :

$$\forall p \in [\![1,B]\!], h^{\uparrow k}[p] = h[(p+k) \bmod B]. \tag{13}$$

Afin d'estimer la similarité entre deux chromas $h_1$ et $h_2$, $h_1$ est comparé aux $B$ transpositions de $h_2$. L'indice de la transposition correspondant à la plus forte corrélation est appelé *Indice de Transposition Optimale* (OTI) [SGH08]. Formellement, l'OTI de deux chromas $h_1$ et $h_2$ est donné par la formule :

$$\mathrm{OTI}(h_1,h_2) = \underset{k \in [\![1,B]\!]}{\operatorname{argmax}} \{ r(h_1, h_2^{\uparrow k}) \}. \tag{14}$$

Si l'indice de transposition optimale correspond à la version non transposée de $h_2$ (soit $k = 0$ si $B = 12$), alors les deux chromas sont déclarés similaires. Dans le cas contraire, ils sont déclarés dissimilaires. De plus, si les chromas sont d'une dimension supérieure aux 12 demi-tons de la gamme tempérée, il est judicieux d'étendre la décision de similarité à toute transposition inférieure au demi-ton. De cette façon, une trame de signal suivant une légère variation des composantes fréquentielles est tout de même considérée comme similaire à la version bien accordée. La mesure binaire de similarité $\lambda_{\mathrm{chr}}$ entre deux chromas $h_1$ et $h_2$ est finalement définie par la formule :

$$\lambda_{\mathrm{chr}}(h_1,h_2) = \begin{cases} \lambda_+ & \text{si} \quad \mathrm{OTI}(h_1,h_2) < \frac{B}{2 \cdot 12} \\ & \text{ou si} \quad |\mathrm{OTI}(h_1,h_2) - B| < \frac{B}{2 \cdot 12} \\ \lambda_- & \text{sinon} \end{cases}, \tag{15}$$

où $\lambda_+$ et $\lambda_-$ sont des paramètres, à définir en fonction de l'application, correspondant aux valeurs associées respectivement à la décision de similarité et à la décision de dissimilarité entre chromas.

## 2.4 Conclusion du chapitre

Ce chapitre présente notre méthode de représentation d'un signal audio sous la forme d'une séquence de descripteurs. Nous justifions le choix de représentation de l'information tonale par la richesse des structures musicales qu'elle possède. Nous présentons alors le descripteur *chroma* tel qu'il est défini dans la littérature et détaillons son obtention à partir du signal audio. Enfin, nous exposons un outil de comparaison entre ces descripteurs en nous appuyant sur une méthode existante et adaptée à la comparaison séquentielle.

Ce chapitre ne décrit qu'un type d'information musicale, l'information tonale. Une perspective majeure de nos travaux est d'examiner une autre information musicale et ainsi d'aboutir à une séquence de descripteurs fondés sur un critère musical distinct. La combinaison de différents critères de description est alors un enjeu important des travaux futurs, dans l'objectif d'extraire depuis le signal audio une représentation encore plus précise et pertinente d'un point de vue musical.

Chapitre 3

# Comparaison de séquences musicales

Analyser les structures répétitives dans la musique exige de pouvoir estimer la *similarité* musicale. Par similarité, nous entendons l'impression *subjective* de percevoir un contenu musical ressemblant, quelles que soient les variations acoustiques des signaux identifiés comme similaires. Cet aspect fondamental de la notion de similarité la rend difficile à définir d'une manière universelle, et sa perception reste propre à la sensibilité de chaque auditeur.

La représentation séquentielle de la musique permet de modéliser la temporalité des œuvres musicales, et constitue pour cette raison une modélisation précise des événements musicaux (voir la Section 1.3). C'est pourquoi de nombreuses techniques de comparaison de séquences sont adaptées et utilisées dans un contexte de recherche en information musicale [Lem00, CI04]. Elles sont par exemple employées pour identifier des requêtes par fredonnement [HFR07, ABSW04], synchroniser des morceaux similaires [DW05, Mül07] ou encore détecter des structures répétitives [CCI+02, DH02, HCC04]. Si les champs d'application et la formalisation des outils peuvent varier entre ces différentes études, elles ont en commun une prise en compte de l'aspect approché des comparaisons effectuées par des techniques dites d'*alignement*.

Nous décrivons dans ce chapitre des outils d'alignement de séquences adaptés à l'estimation de la similarité musicale. Nous montrons que ces outils, d'abord introduits de manière formelle et décorrélée de tout contexte applicatif, permettent alors une évaluation pertinente de la similarité entre séquences musicales. La suite de ce chapitre introduit les notations formelles et les algorithmes de comparaison de séquences, et détaille leurs applications à l'analyse de séquences musicales. La Section 3.1 définit les outils d'algorithmique du texte utilisés pour l'analyse de séquences musicales. À partir de la notion simple d'édition de séquences de symboles, plusieurs techniques d'alignement sont présentées, chaque variante permettant une meilleure robustesse du système face à de possibles variations dans les séquences musicales. Les techniques d'alignement sont ensuite évaluées dans le cadre d'une application musicale en Section 3.2. Dans le contexte de l'estimation de la similarité entre séquences musicales, un cas pratique d'identification de reprises est décrit et évalué. Face au coût calculatoire élevé imposé par les techniques d'alignement, nous examinons une stratégie d'indexation en Section 3.3. Nous proposons une accélération du système de comparaison en nous inspirant d'une méthode heuristique mise au point pour la comparaison efficace de séquences biologiques, que nous adaptons aux spécificités des séquences musicales.

## 3.1 Édition et alignement

Nous introduisons dans cette section les techniques existantes et le vocabulaire d'algorithmique du texte utiles à la comparaison de séquences musicales. Dans la mesure où ces techniques sont largement formalisées et réputées correctes dans la littérature, les preuves d'algorithmes ne font pas l'objet de cette thèse et ne seront pas fournies. Nous invitons le lecteur à se référer par exemple aux ouvrages de Gusfield [Gus97] ou Crochemore *et al.* [CHL07] pour plus d'informations sur les algorithmes et notations détaillés ci-dessous.

### 3.1.1 Notations et définitions

Un *alphabet* $\Sigma$ est un ensemble fini de symboles. Une *séquence de symboles*, ou plus simplement *séquence* $u$ définie sur un alphabet $\Sigma$ est une suite d'éléments de $\Sigma$. La *taille* d'une séquence $u$, notée $|u|$, désigne le nombre de symboles qui constituent $u$. L'ensemble des séquences finies de symboles définis sur $\Sigma$ est noté $\Sigma^*$. On appelle *séquence vide*, et on note $\varepsilon$, la séquence ne comportant aucun symbole (de taille nulle).

On note $u[i]$ le $i$-ème symbole de $u$, $u$ pouvant ainsi être écrite comme une séquence de ses symboles : $u[1]u[2]\dots u[|u|]$. Une séquence $v$ est dite *facteur* de $u$ si et seulement s'il existe deux séquences $w_1$ et $w_2$ telles que $u = w_1vw_2$. Un facteur $v$ d'une séquence $u$ est dit *propre* si $v \neq u$.

Le facteur de $u$ commençant à l'indice $i$ et finissant à l'indice $j$ avec $1 \leq i \leq j \leq |u|$ est noté $u[i\dots j] = u[i]u[i+1]\dots u[j]$. Par convention, si $i > j$, alors $u[i\dots j] = \varepsilon$. Une séquence $v$ est dite *suffixe* (respectivement *préfixe*) de $u$ si et seulement s'il existe une séquence $w$ telle que $u = wv$ (resp. $u = vw$). Tout suffixe ou préfixe $v$ d'une séquence $u$ est dit *propre* si $v \neq u$. En particulier, les séquences $u$ et $\varepsilon$ sont toutes deux facteurs, préfixes et suffixes de la séquence $u$. L'ensemble des facteurs d'une séquence $u$ est noté $\mathcal{S}(u)$.

Deux facteurs non vides $v = u[k\dots l]$ et $w = u[m\dots n]$ de $u$ sont dits *disjoints dans* $u$ si et seulement s'ils ne partagent aucune position de $u$, c'est-à-dire si et seulement si $[k,l] \cap [m,n] = \emptyset$. En outre, on dit que les facteurs $v$ et $w$ sont *chevauchants dans* $u$ si et seulement si un suffixe propre de $v$ est un préfixe propre de $w$, ou si un suffixe propre de $w$ est un préfixe propre de $v$. On désigne par *intersection* de $v$ et $w$, et on note $v \cap_u w$, le facteur de $u$ correspondant aux symboles communs de $v$ et $w$ dans $u$ : $v \cap_u w = u[\max(k,m)\dots\min(l,n)]$. En particulier, l'intersection de facteurs disjoints dans $u$ est $\varepsilon$, l'intersection de $u$ et d'un facteur $v$ de $u$ est $v$ et l'intersection de $u$ et $u$ est $u$. On peut facilement montrer que deux facteurs $v$ et $w$ d'une séquence $u$ se chevauchent dans $u$ si et seulement si (i) ils ne sont pas disjoints dans $u$, (ii) $v \notin \mathcal{S}(w)$ et (iii) $w \notin \mathcal{S}(v)$.

L'exemple suivant illustre la notion de facteurs, préfixes, suffixes et intersections dans la séquence musicalement :

| Séquences | Propriétés |
|---|---|
| $u =$ musicalement | $v_1,v_2,v_3,v_4,u,\varepsilon \in \mathcal{S}(u), v_5 \notin \mathcal{S}(u)$ |
| $v_1 =$ musicale | $v_1$ préfixe de $u$ |
| $v_2 =$ ment | $v_2,v_3$ suffixes de $u$ |
| $v_3 =$ calement | $v_1 \cap_u v_2 = \varepsilon$, $v_1 \cap_u v_3 =$cale, $v_1 \cap_u v_4 =$ sical |
| $v_4 =$ sical | $v_1$ et $v_3$ chevauchants dans $u$, $v_2$ et $v_4$ disjoints dans $u$ |
| $v_5 =$ amusical | $v_1$ et $v_4$ non chevauchants et non disjoints dans $u$ |

| Transcription 1 | | | Transcription 2 | | | Transcription 3 | | |
|---|---|---|---|---|---|---|---|---|
| $u =$ | cabaaab | | $u =$ | cabaaab | | $u =$ | cabaaab | |
| | aabaaab | R | | cabaaab | D | | cabaaab | D |
| | acbaaab | R | | abaaab | C | | abaaab | D |
| | acbaaab | C | | acbaaab | I | | baaab | D |
| | acbaaab | D | | acbaaab | C | | aaab | C |
| | acbbab | R | | acbbaab | R | | acaab | I |
| | acbbaab | C | | acbbaab | C | | acbab | R |
| | acbbaab | D | | acbbaab | C | | acbbb | R |
| | acbbaa | R | | acbbaab | D | | acbba | R |
| | | | | | | | acbbaa | I |
| $v =$ | acbbaa | | $v =$ | acbbaa | | $v =$ | acbbaa | |
| $t_1 =$ RRCDRCDR | | | $t_2 =$ DCICRCCD | | | $t_3 =$ DDDCIRRRI | | |

TAB. 3.1 – *Trois exemples de transcriptions de deux séquences $u$ et $v$.*

### 3.1.2 Distance d'édition et alignement global

L'*édition* [Lev66] est une formalisation simple et fréquemment employée qui permet de calculer une distance entre deux séquences [Gus97]. Éditer une séquence $u$ en une séquence $v$ consiste à transformer $u$ en $v$ en effectuant une série d'opérations élémentaires sur les symboles de ces séquences, appelées *opérations d'édition*. Les opérations usuelles incluent l'*insertion* d'un symbole de $v$, la *suppression* d'un symbole de $u$ et la *substitution* d'un symbole de $u$ par un symbole de $v$. Plus précisément, la substitution d'un symbole de $u$ par le même symbole dans $v$ est appelée *correspondance*, et la substitution d'un symbole de $u$ par un symbole distinct dans $v$ est appelé *remplacement*. Lors de l'édition d'une séquence $u$, on appelle opération d'édition *stricte* toute opération d'édition qui modifie $u$ (telle que l'insertion, la suppression ou le remplacement), et opération d'édition *non stricte* toute opération qui ne modifie pas $u$ (correspondance).

Il convient de noter que d'autres opérations peuvent être considérées pour prendre en compte une édition plus sophistiquée entre des séquences de symboles. Par exemple, les opérations de *consolidation* et de *fragmentation*, telles qu'introduites par Mongeau et Sankoff [MS90], peuvent être employées pour représenter des transformations usuelles sur une description symbolique des notes de musique. Bien que ces différentes opérations ne soient pas considérées dans la suite de nos travaux, leur prise en compte pour l'estimation de similarité constitue une perspective importante de cette thèse.

#### 3.1.2.1 Transcription

Afin de représenter l'édition entre deux séquences $u$ et $v$, une solution simple consiste à décrire la série d'opérations d'édition effectuée pour une transformation de $u$ en $v$. Pour ce faire, on attribue à chaque opération d'édition possible un symbole, et on introduit l'alphabet des opérations d'édition $\Sigma_o$ comme l'ensemble de ces symboles. Par exemple, l'alphabet $\Sigma_o = \{\mathsf{C},\mathsf{R},\mathsf{D},\mathsf{I}\}$ représente les quatre opérations usuelles : la correspondance notée C, le remplacement noté R, la suppression notée D et l'insertion notée I.

**Définition 1 (Transcription)** *On appelle* transcription *de l'édition de $u$ en $v$ toute séquence de symboles définie sur un alphabet d'opérations d'édition $\Sigma_o$ qui décrit une série d'opérations permettant d'éditer une séquence $u$ en une séquence $v$.*

Il existe de nombreuses transcriptions de l'édition de deux séquences $u$ et $v$. Ainsi, dans le Tableau 3.1, $t_1$, $t_2$ et $t_3$ sont des transcriptions possibles de l'édition de $u$ en $v$ sur $\Sigma_o = \{\mathsf{C},\mathsf{R},\mathsf{D},\mathsf{I}\}$.

Pour deux séquences de symboles $u$ et $v$, on note $\mathcal{T}(u,v)$ l'ensemble des transcriptions de l'édition de $u$ en $v$. Il convient de noter que lors d'une transcription de l'édition d'une séquence $u$ en une séquence $v$, on applique une et une seule opération d'édition à chaque position de $u$ et une et une seule opération d'édition à chaque position de $v$ (l'opération de correspondance étant assimilable à la fonction identité). En conséquence, la séquence vide $\varepsilon$ ne peut pas transcrire l'édition de séquences non vides ; formellement, $\varepsilon$ transcrit l'édition de $u$ en $v$ si et seulement si $u = v = \varepsilon$.

#### 3.1.2.2 Alignement global

La transcription d'édition est la représentation de la transformation d'une séquence en une autre. Une alternative pour visualiser l'édition consiste à expliciter les opérations d'édition effectuées dans la transformation en plaçant en correspondance les symboles substitués. Ainsi, l'*alignement global* est défini de la manière suivante :

**Définition 2 (Alignement global)** *Soient $\Sigma_o$ un alphabet d'opérations d'édition, $u$ et $v$ deux séquences. Un alignement global entre $u$ et $v$ est une séquence $z$ sur l'alphabet $(\Sigma_o \cup \{\varepsilon\}) \times (\Sigma_o \cup \{\varepsilon\}) \setminus \{(\varepsilon,\varepsilon)\}$ dont la projection sur la première composante est $u$ et la projection sur la seconde est $v$.*

*Chaque symbole $(a,b)$ d'un alignement de $u$ en $v$ représente une opération de l'édition entre les deux séquences : une substitution si $a$ et $b$ sont des symboles respectifs de $u$ et $v$, une suppression si $b = \varepsilon$ et une insertion si $a = \varepsilon$.*

En pratique, l'alignement global $\mathcal{A}$ de deux séquences de symboles $u$ et $v$ pour une transcription donnée $t$ s'obtient en matérialisant le symbole $\varepsilon$ par un *symbole spécial d'alignement* $\phi$ (un tiret par exemple). Ainsi, on ajoute $\phi$ dans $u$ et dans $v$, puis on place les deux séquences résultantes $u_t$ et $v_t$ l'une en-dessous de l'autre de telle sorte qu'à chaque symbole de $u_t$ est associé un symbole de $v_t$ qui vérifie les opérations de $t$. Plus précisément, dans le cas de l'insertion d'un symbole $s$ de $v$, un symbole spécial $\phi$ est ajouté dans $u_t$ de telle sorte que $s$ et $\phi$ soient alignés l'un en-dessous de l'autre; dans le cas de la suppression d'un symbole $s$ de $u$, un symbole spécial $\phi$ est ajouté dans $v$ de telle sorte que $s$ et $\phi$ soient alignés l'un en-dessous de l'autre; enfin, dans le cas d'une substitution d'un symbole $s_1$ de $u$ par un symbole $s_2$ de $v$, $s_1$ dans $u_t$ et $s_2$ dans $v_t$ sont alignés l'un en-dessous de l'autre.

L'exemple suivant décrit une transcription $t$ des séquences $u = \mathsf{aabcdaaabda}$ et $v = \mathsf{abbccdaacda}$ (ii) ainsi que l'alignement global correspondant (i), avec le tiret comme symbole spécial d'alignement $\phi$ :

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (i) | $u_t$ | a | a | b | c | - | d | a | a | a | b | d | a |
| | $v_t$ | a | b | b | c | c | d | a | a | - | c | d | a |
| (ii) | $t$ | C | R | C | C | I | C | C | C | D | R | C | C |

Ce processus définit une équivalence entre la transcription $t$ et l'alignement global $\mathcal{A}$ : à toute transcription correspond un unique alignement, et réciproquement. Malgré cette équivalence, l'interprétation et l'utilisation de l'alignement et de la transcription diffèrent en pratique. La transcription représente la série de changements nécessaires à la tranformation d'une séquence en une autre, alors que l'alignement explicite la transformation des séquences elles-mêmes. La première représentation est ainsi centrée sur le *processus* de transformation, alors que la seconde met en avant le *produit* de cette transformation.

Comme dans le cas de la transcription, il existe un nombre important d'alignements globaux d'une séquence de symboles sur une autre. Afin de différencier ces alignements, il convient d'introduire une pondération des opérations d'édition.

#### 3.1.2.3 Pondération

À chacune des opérations élémentaires d'édition est associé un poids, qui dépend des symboles comparés. Formellement, on définit une fonction de pondération $\delta : \Sigma \times \Sigma \to \mathbb{R}^+$ qui traduit le *coût* de l'alignement de deux symboles.

Par exemple, pour deux symboles $x$ et $y$ alignés,

- si $x$ est un symbole spécial, alors $\delta(x,y)$ correspond à un coût $\delta_{\mathsf{I}}$ d'insertion de $y$;
- si $y$ est un symbole spécial, alors $\delta(x,y)$ correspond à un coût $\delta_{\mathsf{D}}$ de suppression de $x$;
- si $x$ et $y$ sont identiques, alors $\delta(x,y)$ correspond à un coût de correspondance $\delta_{\mathsf{C}}$;
- si $x$ et $y$ sont différents, alors $\delta(x,y)$ correspond à un coût de remplacement $\delta_{\mathsf{R}}$.

La spécification de la fonction de pondération $\delta$ est appelée *schéma de coûts de pondération*. En pratique, le schéma des coûts de pondération est souvent défini par la donnée des fonctions de pondération de chaque opération élémentaire $\delta_{\mathsf{E}}$ pour $\mathsf{E} \in \Sigma_o$.

**Définition 3 (Coût de transcription)** *Soient $u$ et $v$ deux séquences, et $t$ une transcription non vide de l'édition de $u$ en $v$. On pose $u_t$ et $v_t$ les séquences respectives $u$ et $v$ alignées selon la transcription $t$. On appelle coût de la transcription $t$, et on note $\Delta(t)$, la somme des coûts d'opérations d'édition :*

$$\Delta(t) = \sum_{i=1}^{|t|} \delta(u_t[i],v_t[i]).$$

Par convention, le coût de transcription de la séquence vide $\varepsilon$ est nul : $\Delta(\varepsilon) = 0$.

Le coût de transcription permet ainsi de différencier et de pondérer les éditions possibles de $u$ en $v$.

#### 3.1.2.4 Distance d'édition

La pondération par coûts permet d'introduire un critère optimal pour les transcriptions d'édition. On définit alors une *transcription de coût optimal* entre deux séquences de symboles $u$ et $v$ comme une transcription de l'édition de $u$ en $v$ de coût minimal.

**Définition 4 (Distance d'édition)** *Soient $u$ et $v$ deux séquences. On appelle distance d'édition entre $u$ et $v$, notée $d(u,v)$, le coût d'une transcription optimale de l'édition de $u$ en $v$ :*

$$d(u,v) = \min_{t \in \mathcal{T}(u,v)} \Delta(t).$$

Le coût optimal peut être atteint pour plusieurs transcriptions d'édition, ainsi il n'y a pas forcément unicité des transcriptions optimales de l'édition de $u$ en $v$.

La méthode pratique de calcul de la distance d'édition est généralement attribuée à Needleman et Wunsch [NW70] (ou Levenshtein [Lev66] dans le cas de coûts unitaires). Elle est basée sur un principe de *programmation dynamique* pour calculer cette distance de manière efficace.

#### 3.1.2.5 Calcul pratique de la distance d'édition

Soient $u$ et $v$ deux séquences définies sur un alphabet $\Sigma$ et de longueur respective $n = |u|$ et $m = |v|$. Pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$ , on désigne par $d_{i,j}$ la distance d'édition entre $u[1 \dots i]$ et $v[1 \dots j]$.

Le calcul de $d_{i,j}$ est effectué par récurrence sur $i$ et $j$. Son initialisation se fait selon la proposition :

**Proposition 1 (Initialisation)** *Pour tout $(i,j) \in [\![0,n]\!] \times [\![0,m]\!]$,*

$$\begin{cases} d_{i,0} = i \\ d_{0,j} = j \end{cases}.$$

La récurrence sur $i$ et $j$ est donnée par la proposition :

**Proposition 2 (Récurrence)** *Soient $\phi$ le symbole spécial d'alignement et $\delta$ un schéma de coûts de pondération. Pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$,*

$$d_{i,j} = \min \begin{cases} d_{i-1,j} & + & \delta(u[i],\phi) & \textit{(i)} \\ d_{i,j-1} & + & \delta(\phi,v[j]) & \textit{(ii)} \\ d_{i-1,j-1} & + & \delta(u[i],v[j]) & \textit{(iii)} \end{cases}.$$

Le calcul par programmation dynamique permet ainsi de déduire la distance $d_{i,j}$ à partir d'une optimisation locale entre (i) la distance $d_{i-1,j}$ majorée d'un coût de suppression du symbole $i$ de $u$, (ii) la distance $d_{i,j-1}$ majorée d'un coût d'insertion du symbole $j$ de $v$, ou (iii) la distance $d_{i-1,j-1}$ majorée d'un coût de substitution (correspondance ou remplacement) du symbole $i$ de $u$ par le symbole $j$ de $v$. La preuve de cet algorithme peut être trouvée, par exemple, dans [Gus97, p.218-219].

$d_{i,j}$ correspond à la distance d'édition entre les $i$ premiers symboles de $u$ et les $j$ premiers symboles de $v$. Par conséquent, la distance d'édition recherchée $d(u,v)$ de $u$ et $v$ est simplement donnée par :

$$d(u,v) = d_{n,m}. \tag{16}$$

### 3.1.3 Alignement local

Pour de nombreuses applications en analyse de séquences biologiques comme en analyse musicale, deux séquences peuvent ne pas présenter de forte ressemblance dans leur globalité mais contenir des régions de grande similarité. L'alignement local [SW81] est une variante de la distance d'édition qui permet de localiser dans les deux séquences les régions les plus similaires.

#### 3.1.3.1 Similarité

La similarité est une notion duale de la notion de distance. Les deux mesures diffèrent sur leur logique : à deux symboles identiques est associée une faible distance et une forte similarité, alors qu'à deux symboles distincts est associée une distance élevée et une faible similarité. En pratique, la notion de similarité entre séquences convient mieux à l'évaluation d'alignements locaux, comme décrit ci-après.

Formellement, on définit une fonction de pondération $\lambda : \Sigma \times \Sigma \to \mathbb{R}$ qui représente le *score* de l'alignement de deux symboles. Comme pour la fonction $\delta$ dans le cas de la pondération par coûts, la définition de $\lambda$ peut être divisée en plusieurs définitions de fonctions de score $\lambda_{\mathsf{E}}$ pour chacune des opérations d'édition possibles $\mathsf{E} \in \Sigma_o$. La spécification de $\lambda$, donc de l'ensemble des fonctions de pondération de chaque opération élémentaire, est appelée *schéma de scores de pondération.*

Il convient de noter que, contrairement aux coûts de pondération qui sont définis sur $\mathbb{R}^+$, un score de pondération peut correspondre à n'importe quel réel, positif ou négatif. On introduit alors la notion de bonne formation d'un schéma de scores de pondération :

**Définition 5 (Schéma de scores bien formé)** *Un schéma de scores de pondération est dit bien formé si à toute opération d'édition stricte est associé un score négatif, et à toute opération d'édition non stricte est associé un score positif.*

**Définition 6 (Score de transcription)** *Soit $u$ et $v$ deux séquences, et soit $t$ une transcription non vide de l'édition de $u$ en $v$. On pose $u_t$ et $v_t$ les séquences respectives $u$ et $v$ alignées selon la transcription $t$. On appelle score de la transcription $t$, et on note $\Lambda(t)$, la somme des scores d'opérations d'édition :*

$$\Lambda(t) = \sum_{i=1}^{|t|} \lambda(u_t[i],v_t[i]).$$

Par convention, le score de la transcription vide est nul : $\Lambda(\varepsilon) = 0$.

**Définition 7 (Score de similarité)** *Soient $u$ et $v$ deux séquences. On appelle score de similarité entre $u$ et $v$, noté $s(u,v)$, le score d'une transcription optimale de l'édition de $u$ en $v$ :*

$$s(u,v) = \max_{t \in \mathcal{T}(u,v)} \Lambda(t).$$

Comme dans le cas de la distance d'édition, le score optimal peut être atteint pour plusieurs transcriptions d'édition ; ainsi, il n'y a pas forcément unicité des transcriptions de score optimal de l'édition de $u$ en $v$.

#### 3.1.3.2 Similarité locale et alignement local

Le problème de la *similarité locale* [SW81, Gus97] peut être formulé comme suit :

**Problème 1 (Similarité locale)** *Soient $u$ et $v$ deux séquences définies sur un alphabet $\Sigma$, et $\mathcal{S}(u)$ et $\mathcal{S}(v)$ leurs ensembles de facteurs respectifs. Le problème de la similarité locale consiste à trouver deux séquences $u' \in \mathcal{S}(u)$ et $v' \in \mathcal{S}(v)$ vérifiant :*

$$s(u',v') = \max_{(x,y)\in\mathcal{S}(u)\times\mathcal{S}(v)} s(x,y) \quad .$$

*Ce score optimal est noté $s^*(u,v)$, et appelé score de similarité locale de $u$ et $v$.*

Par convention, on notera $u^*$ et $v^*$ les facteurs respectifs de $u$ et $v$ effectivement alignés, c'est-à-dire tels que $s^*(u,v) = s(u^*,v^*)$. L'alignement global de ces deux facteurs $u^*$ et $v^*$ de score $s^*(u,v)$ est appelé *alignement local optimal* de $u$ et $v$, ou simplement *alignement local* de $u$ et $v$ en l'absence d'ambiguïté.

L'exemple suivant décrit un alignement local optimal de $u =$ babcbaaabcbb et $v =$ ccccbcababccacc (i) ainsi que la transcription des facteurs alignés en noir (ii), avec $\phi =$ "-" comme symbole spécial d'alignement :

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (i) | $u_t^*$ | | | | b | a | b | c | - | b | a | a | a | b | c | - | b | b |
| | $v_t^*$ | c | c | c | c | b | c | a | b | - | - | a | b | c | c | a | c | c |
| (ii) | $t$ | | | | | | C | C | I | C | D | D | C | C | C | | | |

Il convient de noter que les symboles grisés *ne sont pas alignés* dans cet exemple, et seule la similarité locale des facteurs symbolisés en noir est finalement prise en compte.

La méthode pratique de calcul de l'alignement local est attribuée à Smith et Waterman [SW81]. Comme dans le cas de la distance d'édition, elle est basée sur un principe de programmation dynamique.

#### 3.1.3.3 Calcul pratique de la similarité locale

Soient $u$ et $v$ deux séquences définies sur un alphabet $\Sigma$ de longueur respective $n = |u|$ et $m = |v|$. Afin de déterminer les deux facteurs de $u$ et $v$ de similarité optimale, Smith et Waterman [SW81] proposent de parcourir tous les préfixes de $u$ et $v$, et de déterminer pour chaque couple de préfixes, par programmation dynamique, les deux suffixes qui optimisent le score de similarité. Plus précisément, pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$, on désigne par $s^+_{i,j}$ le score de similarité optimal entre un suffixe de $u[1 \dots i]$, potentiellement vide, et un suffixe de $v[1 \dots j]$, potentiellement vide. Dans le cas où les deux suffixes optimaux correspondent à la séquence vide, leur score de similarité est nul. Par conséquent, $s^+_{i,j}$ est égal au score $s$ de similarité maximale entre un suffixe de $u$ se terminant en $i$ et un suffixe de $v$ se terminant en $j$ si $s > 0$, à 0 sinon. Formellement,

$$s^+_{i,j} = \max\{0\} \cup \{s(u[k \dots i],v[l \dots j]) | (k,l) \in [\![1,i]\!] \times [\![1,j]\!]\}. \tag{17}$$

Le calcul pratique de $s^+_{i,j}$ est réalisé par récurrence sur $i$ et $j$. Son initialisation se fait selon la proposition :

**Proposition 3 (Initialisation)** *Pour tout $(i,j) \in [\![0,n]\!] \times [\![0,m]\!]$,*

$$\begin{cases} s^+_{i,0} = 0 \\ s^+_{0,j} = 0 \end{cases} .$$

La récurrence sur $i$ et $j$ est donnée par la proposition :

**Proposition 4 (Récurrence)** *Soient $\phi$ le symbole spécial d'alignement et $\lambda$ un schéma de scores de pondération bien formé. Pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$,*

$$s^+_{i,j} = \max \begin{cases} s^+_{i-1,j} & + & \lambda(u[i],\phi) & (i) \\ s^+_{i,j-1} & + & \lambda(\phi,v[j]) & (ii) \\ s^+_{i-1,j-1} & + & \lambda(u[i],v[j]) & (iii) \\ 0 & & & (iv) \end{cases} .$$

Comme précédemment, le calcul par programmation dynamique permet de déduire le score de similarité $s^+_{i,j}$ à partir d'une optimisation locale entre (i) le score $s^+_{i-1,j}$ amoindri d'un score de suppression du symbole $i$ de $u$, (ii) le score $s^+_{i,j-1}$ amoindri d'un score d'insertion du symbole $j$ de $v$, (iii) le score $s^+_{i-1,j-1}$ additionné à un score de substitution (correspondance ou remplacement) du symbole $i$ de $u$ par le symbole $j$ de $v$, et (iv) la condition d'alignement local 0 qui permet d'aligner des suffixes vides de $u$ et $v$, et dont le choix marque ainsi le début d'un nouvel alignement. La preuve de cet algorithme peut être trouvée, par exemple, dans [Gus97, p.233-234]

$s^+_{i,j}$ correspond au score de similarité optimal entre un suffixe des $i$ premiers symboles de $u$ et un suffixe des $j$ premiers symboles de $v$. Par conséquent, le score de similarité locale recherché $s^*(u,v)$ de $u$ et $v$ est donné par :

$$s^*(u,v) = \max_{(i,j)\in[\![1,n]\!]\times[\![1,m]\!]} s^+_{i,j} . \tag{18}$$

### 3.1.4 Implémentation

Nous explicitons dans cette section plusieurs points d'implémentation qui permettent de calculer les alignements précédemment définis. Cette section se place dans un contexte pratique, où une certaine efficacité calculatoire doit être assurée.

#### 3.1.4.1 Matrice de programmation dynamique

Les points d'implémentation décrits dans cette section sont analogues pour la procédure d'alignement global et d'alignement local. La suite de cette section détaille le calcul dans le cas global, les explications étant facilement transférables à l'alignement local.

Les formules de récurrence données en Propositions 2 et 4 peuvent aisément être implémentées par une procédure récursive. Une approche récursive simple consiste à calculer $d_{n,m}$ en appelant récursivement le calcul de $d_{i-1,j}$, de $d_{i,j-1}$ et de $d_{i-1,j-1}$ pour tous les indices $i$ de $n$ à 0 et $j$ de $m$ à 0. Cependant, le nombre d'appels récursifs nécessaire au calcul de $d_{n,m}$ croît dans ce cas de manière exponentielle par rapport à $n$ et $m$, ce qui rend la distance d'édition particulièrement coûteuse en temps de calcul. Or, en remarquant qu'il n'existe que $(n+1)\times(m+1)$ couples $(i,j)$ distincts, on déduit qu'au plus $(n+1)\times(m+1)$ appels récursifs distincts peuvent être effectués. Par conséquent, il est plus efficace de calculer la distance $d_{n,m}$ en obtenant $d_{i,j}$ pour

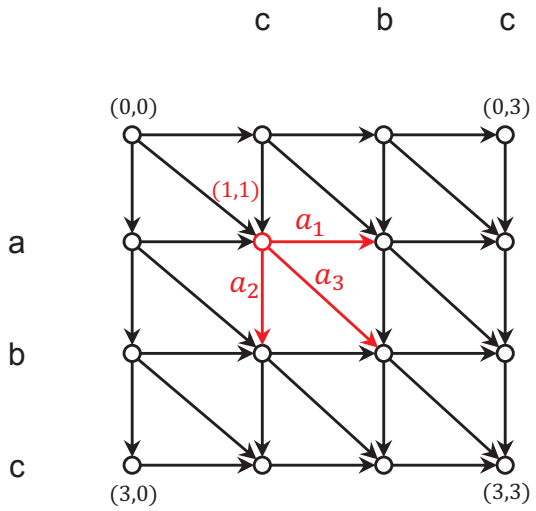


FIG. 3.1 – *Graphe d'édition entre* abc *et* cbc.

des indices $i$ et $j$ croissants et en mémorisant les distances à mesure qu'elles sont calculées. Il s'agit d'un principe de programmation dynamique.

Formellement, on introduit une *matrice de programmation dynamique* $M$ de taille $(n+1) \times (m+1)$ dont chaque coefficient $(i,j)$ correspond au résultat $d_{i,j}$ d'une étape de la Proposition 2 :

$$\forall (i,j) \in [\![0,n]\!] \times [\![0,m]\!], M[i][j] = d_{i,j}. \tag{19}$$

La ligne 0 et la colonne 0 de $M$ sont directement renseignées à partir des conditions initiales de la récurrence (Prop. 1). Les valeurs de $M$ sont ensuite calculées ligne à ligne de haut en bas à partir du coefficient $M(1,1)$ jusqu'au coefficient $M(n,m)$. Dans chaque ligne, les valeurs de $M$ sont calculées de gauche à droite.

Le calcul progressif de $M$ permet ainsi d'obtenir, pour $i$ et $j$ croissants, toutes les distances $d_{i,j}$ nécessaires à l'évaluation par programmation dynamique de $d_{n,m}$. Cette procédure assure d'évaluer exactement $n \times m$ fois la formule de la Proposition 2, et permet ainsi le calcul exact de la distance d'édition de $u$ en $v$ par un nombre suffisant d'opérations [Gus97].

Le calcul par programmation dynamique permet d'obtenir la distance d'édition entre deux séquences $u$ et $v$. Outre cette distance, en fonction de l'application, la comparaison de séquences implique souvent de transcrire l'édition optimale effectuée pour obtenir cette distance, afin d'expliciter l'alignement global entre $u$ et $v$.

#### 3.1.4.2 Graphe d'édition

L'espace parcouru pour le calcul par programmation dynamique de l'alignement de deux séquences $u$ et $v$ peut être visualisé en représentant toutes les opérations d'édition possibles entre les séquences sous forme d'un graphe. Formellement, on introduit le *graphe d'édition* de $u$ de taille $n$ et $v$ de taille $m$ comme le graphe acyclique orienté défini par :

- $(n+1) \times (m+1)$ nœuds, chacun étiquetés par $(i,j)$, correspondant à une paire de positions dans $u$ et $v$ ;

- À chaque nœud $(i,j)$ sont associés un arc $a_1$ vers le nœud $(i,j+1)$, un arc $a_2$ vers le nœud $(i+1,j)$ et un arc $a_3$ vers le nœud$(i+1,j+1)$, si ceux-ci existent ;
- Chaque arc est pondéré par le poids de l'opération qu'il représente : le poids de la suppression de $u[i]$ pour $a_1$, le poids de l'insertion de $v[j]$ pour $a_2$ ou le poids de la substitution de $u[i]$ en $v[j]$ pour $a_3$.

À titre d'exemple, le graphe d'édition entre les séquences abc et cbc est donné en Figure 3.1.

Ainsi, tout chemin dans le graphe d'édition de $u$ et $v$ depuis la position (0,0) jusqu'à la position $(n,m)$ décrit un alignement global de $u$ en $v$, et le chemin de score minimal correspond à la transcription optimale.

La représentation sous forme de graphe d'édition est ainsi équivalente à la définition de la matrice de programmation dynamique. La première est plus adaptée pour visualiser les alignements effectués, tandis que la seconde est utile pour les descriptions plus formelles de l'implémentation associée à l'alignement des séquences.

#### 3.1.4.3 Obtention des transcriptions optimales

Selon le cadre applicatif, il peut être utile de disposer non seulement de la valeur associée à l'alignement (distance d'édition ou score de similarité), mais aussi de la transcription optimale effectuée.

**Transcription de l'alignement global**

Une fois la distance d'édition $d(u,v)$ obtenue pour deux séquences $u$ et $v$, il est possible d'expliciter les transcriptions d'édition optimales de coût $d(u,v)$ par une procédure appelée *tracé arrière*. Son calcul suppose d'enregistrer, pour chaque coefficient de $M$ évalué dans le calcul par programmation dynamique, un *pointeur d'édition* indiquant les coefficients voisins qui correspondent aux opérations d'édition optimales pour le coefficient en cours. Formellement, pour tout coefficient $(i,j)$ de $M$,

- si $i > 0$ et $d_{i,j} = d_{i-1,j} + \delta(u[i],\phi)$ (choix (i) dans la Prop. 2), alors un pointeur vertical est créé de $(i,j)$ vers $(i-1,j)$ ;
- si $j > 0$ et $d_{i,j} = d_{i,j-1} + \delta(\phi,v[j])$ (choix (ii) dans la Prop. 2), alors un pointeur horizontal est créé de $(i,j)$ vers $(i,j-1)$ ;
- si $i > 0$, $j > 0$ et $d_{i,j} = d_{i-1,j-1} + \delta(u[i],v[j])$ (choix (iii) dans la Prop. 2), alors un pointeur diagonal est créé de $(i,j)$ vers $(i-1,j-1)$.

Cette évaluation est effectuée à chaque calcul d'un nouveau coefficient de $M$. Par définition de $M$, au moins un pointeur est défini pour chaque coefficient $(i,j)$ de $M$. De plus, pour chaque coefficient, plusieurs pointeurs peuvent être définis en cas d'égalité des poids d'édition.

Le graphe formé par tous les pointeurs d'édition peut ainsi être vu comme un sous-graphe du graphe d'édition de $u$ et $v$ dont les arcs sont inversés. Grâce à l'ensemble des pointeurs ainsi définis, les transcriptions optimales de $u$ en $v$ peuvent être reconstruites en suivant un *tracé arrière*. En effet, d'après la définition du calcul par programmation dynamique, à tout chemin $\mathcal{C}$ qui suit les pointeurs d'édition depuis la position finale $(n,m)$ jusqu'à la position initiale (0,0) correspond une transcription $t$ de coût optimal. Cette transcription peut aisément être déduite du chemin $\mathcal{C}$ en interprétant chaque pointeur horizontal comme une insertion, chaque

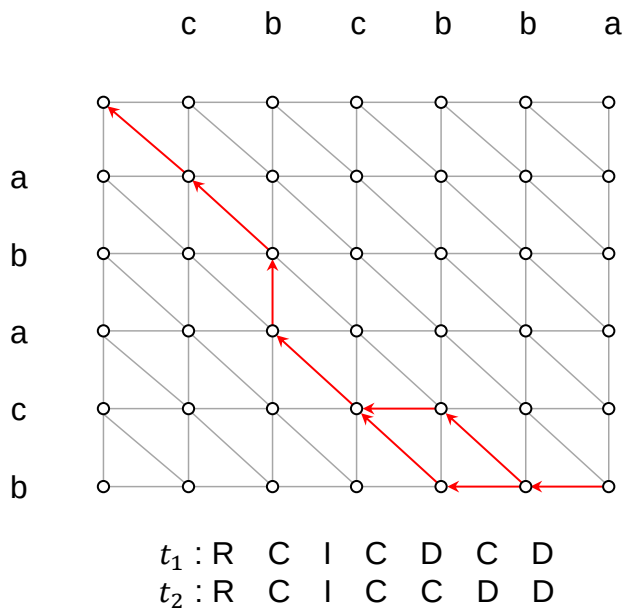


FIG. 3.2 – *Tracé arrière sur le graphe d'édition entre les séquences* abacb *et* cbcbba. *Les transcriptions* $t_1$ *et* $t_2$ *correspondent aux chemins possibles du tracé arrière.*

pointeur vertical comme une suppression et chaque pointeur diagonal comme une substitution. La figure 3.2 présente l'exemple d'un tel tracé arrière, où les insertions, suppressions et substitutions correspondant aux transcriptions optimales sont représentées par des arcs rouges. Ce tracé arrière permet ainsi d'obtenir, après calcul par programmation dynamique, toutes les transcriptions optimales, donc tous les alignements globaux optimaux des deux séquences comparées.

### Transcription de l'alignement local

Comme expliqué en Section 3.1.3.3, l'alignement local $s^*(u,v)$ de deux séquences $u$ et $v$ correspond à l'alignement global de deux facteurs de $u$ et $v$ de score maximal. Par conséquent, transcrire l'alignement local entre $u$ et $v$ équivaut à transcrire l'alignement global des facteurs alignés.

Soient $u^*$ et $v^*$ les deux facteurs respectifs de $u$ et $v$ correspondant à l'alignement optimal calculé lors de la procédure d'alignement local, c'est-à-dire vérifiant $s^*(u,v) = s(u^*,v^*)$. On pose $i$, $j$, $k$ et $l$ les quatre entiers positifs tels que $u^* = u[i \dots k]$ et $v^* = v[j \dots l]$. On note $M$ la matrice d'alignement local de $u$ sur $v$ qui décrit la récurrence sur $s^+$ de la Proposition 4.

D'après l'algorithme d'alignement local, $s^*(u,v) = s^+_{k,l}$. On en déduit que la transcription de l'alignement local de $u$ sur $v$ se *termine* au couple d'indices de score maximal dans la matrice $M$. La transcription de l'alignement local peut ensuite être obtenue en appliquant la procédure de tracé arrière, définie précédemment, à partir du coefficient $(k,l)$ de la matrice $M$. Le tracé s'effectue alors jusqu'à ce qu'un coefficient $(i,j)$ de $M$ nul soit atteint. En effet, dans ce cas, le choix de réinitialisation (iv) dans la Proposition 4 assure que les indices $i$ et $l$ correspondent au début des facteurs alignés. Ainsi, la transcription de l'alignement local *commence* au premier couple parcouru d'indices correspondant à un score nul.

Cette procédure permet d'obtenir le chemin de transcription locale optimale entre $u$ et $v$. Comme dans le cas de l'alignement global, les arêtes de celui-ci peuvent être interprétées comme différentes opérations d'édition. Il convient de noter que l'unicité de la transcription optimale dans le cas d'alignement local n'est pas assurée, l'ensemble des transcriptions pouvant être obtenu en suivant tous les chemins possibles en remontant de l'indice $(k,l)$ à l'indice de coefficient nul $(i,j)$.

#### 3.1.4.4 Complexité algorithmique

**Complexité temporelle**

Soient $u$ et $v$ deux séquences de longueurs respectives $n$ et $m$. Dans le cas global ou local, le calcul de l'alignement de $u$ et $v$ par programmation dynamique requiert le calcul des coefficients de la matrice $M$. Chacun de ces coefficients est évalué en réalisant un nombre constant de calculs : 3 comparaisons et 3 opérations arithmétiques pour l'alignement global, 4 comparaisons et 3 opérations arithmétiques pour l'alignement local. La procédure d'alignement global ou local a donc une complexité de $\Theta(nm)$.

Comme précisé en Section 3.1.4.3, le calcul des pointeurs d'édition peut être effectué au fur et à mesure du calcul des coefficients de $M$. Pour chaque case de la matrice, un nombre constant de 3 pointeurs est mis en place, ainsi l'établissement de ceux-ci n'augmente pas la complexité du calcul par programmation dynamique. La procédure de tracé arrière requiert ensuite le parcours d'un chemin de la dernière case de $M$ à la première case, dans le cas d'un alignement global et dans le pire des cas d'un alignement local. Par conséquent, le tracé arrière engendre une complexité temporelle de $O(n+m)$ dans le cas global et local.

**Complexité spatiale**

Le calcul d'un alignement global ou local requiert le stockage de toutes les valeurs de la matrice de programmation dynamique $M$, impliquant un besoin en espace de $\Theta(nm)$. Or, cette contrainte s'avère problématique pour la comparaison de longues séquences dans de grandes bases de données. Cependant, cette complexité peut aisément être réduite à $O(\min(n,m))$ en observant que lors de l'évaluation d'un coefficient de $M$ à la ligne $i$, seuls les coefficients de la même ligne et de la ligne $i-1$ sont nécessaires. Par conséquent, le calcul par programmation dynamique peut être réalisé en ne conservant que les deux dernières lignes de $M$ au cours de son évaluation. De plus, dans le cas où la transcription d'alignement est requise, l'algorithme d'Hirschberg [Hir75] permet, par une approche *Diviser pour régner*, de réduire également la complexité spatiale à $\Theta(\min(n,m))$ en conservant le même ordre de complexité temporelle.

### 3.1.5 Variante robuste aux transpositions locales

Le phénomène de transpositions locales de séquences tonales introduit une variation tonales susceptible de faire tomber en échec les techniques d'alignement décrites précédemment. Dans le cadre de la comparaison de séquences tonales, une variante de l'alignement proposée par Allali *et al.* [AFHI07] assure une prise en compte des

transpositions locales. La variante est présentée dans le cas de l'alignement global, avec l'indication qu'elle peut aisément être adaptée à l'alignement local. Cette section décrit donc le principe de cette méthode dans le cas de l'alignement local.

### Définitions complémentaires

**Définition 8 (Fonction de transposition)** *Soit $\Sigma$ un alphabet. On appelle fonction de transposition toute fonction $\gamma : \Sigma \times \mathbb{N} \to \Sigma$ qui à un symbole et une valeur de transposition associe un autre symbole de $\Sigma$.*

Par exemple, la fonction $\gamma(x,n) = x + n$ pour $x$ et $n$ entiers correspond à une fonction de transposition simple dans l'alphabet des entiers positifs.

Dans le cas d'un alphabet tonal, comme expliqué en Section 2.3.2, le nombre de transpositions possibles d'un symbole est susceptible d'être réduit à un sous-ensemble fini de $\mathbb{N}$, noté $[\![1,B]\!]$, où $B$ est le coefficient entier de transposition maximale (par exemple $B = 12$ pour une division en demi-tons de l'octave). Une fonction de transposition peut correspondre à un décalage du profil de classes tonales, comme défini par la Formule 13, par exemple.

**Définition 9 (Transposition globale)** *Soient $u$ une séquence de taille $n$, $k$ un entier et $\gamma$ une fonction de transposition. On appelle transposition globale de $u$ de $k$ symboles, et on note $u^k_\gamma$, la séquence de longueur $n$ obtenue en transposant par $\gamma$ chacun des symboles de $u$ :*

$$\forall i \in [\![1,n]\!], u^k_\gamma[i] = \gamma(u[i],k).$$

**Définition 10 (Transposition locale)** *Soient $u$ une séquence de taille $n$, $i$ et $j$ deux entiers positifs tels que $i \leq j \leq n$ et $\gamma$ une fonction de transposition. On appelle transposition locale de $u$ sur l'intervalle $[i,j]$ toute séquence $v$ pouvant être obtenue par transposition globale par $\gamma$ du facteur $u[i \dots j]$. Formellement, $v$ sur $\Sigma$ est une transposition locale de $u$ par $\gamma$ sur l'intervalle $[i,j]$ si et seulement si*

$$\exists k \in \mathbb{N} : v = u^k_\gamma[i \dots j].$$

**Définition 11 (Transposition locale recouvrante)** *Soient $u$ une séquence de taille $n$ et $\gamma$ une fonction de transposition. On appelle transposition locale recouvrante de $u$ toute séquence $v$ composée d'une concaténation de transpositions locales de $u$ sur des intervalles disjoints deux à deux et d'union $[1,n]$. Formellement, une séquence $v$ sur $\Sigma$ est une transposition locale recouvrante de $u$ si et seulement si*

$$\begin{aligned}&\exists m \in \mathbb{N}^+, (k_1, \dots, k_m) \in \mathbb{N}^m, (i_1, \dots, i_m) \in [\![1,n]\!]^m : \\ &\quad v = u^{k_1}_\gamma[1 \dots i_1] u^{k_2}_\gamma[i_1 + 1 \dots i_2] \quad \dots \quad u^{k_m}_\gamma[i_{m-1} + 1 \dots n].\end{aligned}$$

Le nombre de transpositions locales $m$ est appelé *rang* de la transposition locale recouvrante $v$, et noté $r(v)$. Enfin, on note $\Gamma(u)$ l'ensemble des transpositions locales recouvrantes de $u$.

### Problème

Pour deux séquences $u$ et $v$ définies sur $\Sigma$, le problème de similarité locale avec transpositions locales est défini comme une optimisation du score de similarité locale entre $v$ et une transposition locale recouvrante de $u$ :

**Problème 2 (Similarité locale avec transpositions locales)** *Soient $u$ et $v$ deux séquences, et $\lambda_T$ un score d'édition stricte. Trouver le score de similarité locale avec transpositions locales $s^*_\gamma(u,v)$ défini par :*

$$s^*_\gamma(u,v) = \max_{u' \in \Gamma(u)} \{s^*(u',v) + (r(u') - 1) \cdot \lambda_T\}.$$

En pratique, le score $\lambda_T$ est spécifié par le schéma de scores de pondération.

### Calcul pratique de la similarité locale avec transpositions

Le calcul par programmation dynamique de l'alignement local avec transpositions locales entre deux séquences $u$ et $v$ sur $\Sigma$ consiste à calculer simultanément des matrices de programmation dynamique correspondant à l'alignement entre $v$ et toutes les transpositions locales recouvrantes possibles de $u$. Dans la suite, on suppose que le nombre de transpositions possibles d'un symbole de $\Sigma$ est fini, et noté $B$.

Pour tout $(i,j) \in [\![1,n]\!] \times [\![1,m]\!]$, on désigne par $s^+_{k\,i,j}$ le score de similarité maximale entre un suffixe de $v$ se terminant en $j$, et un suffixe de $u$ se terminant en $i$ transposé tel que la dernière transposition appliquée est de $k$ symboles.

Le calcul pratique de $s^+_{k\,i,j}$ est réalisé par récurrence sur $i$, $j$ et $k$. Son initialisation se fait selon la proposition :

**Proposition 5 (Initialisation)** *Pour tout $(i,j,k) \in [\![0,n]\!] \times [\![0,m]\!] \times [\![1,B]\!]$,*

$$\begin{cases} s^+_{k\,i,0} = 0 \\ s^+_{k\,0,j} = 0 \end{cases}.$$

La récurrence est alors donnée par la proposition :

**Proposition 6** *Soient $\phi$ le symbole spécial d'alignement et $\lambda$ un schéma de scores de pondération bien formé. Pour tout $(i,j,k) \in [\![1,n]\!] \times [\![1,m]\!] \times [\![1,B]\!]$,*

$$s^+_{k\,i,j} = \max \begin{cases} s^+_{k\,i-1,j} + \lambda(u^k_\gamma[i],\phi) & \textit{(i)} \\ s^+_{k\,i,j-1} + \lambda(\phi,v[j]) & \textit{(ii)} \\ s^+_{k\,i-1,j-1} + \lambda(u^k_\gamma[i],v[j]) & \textit{(iii)} \\ s^+_{l\,i-1,j-1} + \lambda(u^k_\gamma[i],v[j]) + \lambda_T \quad \forall l \in [\![1,B]\!]\backslash\{k\} & \textit{(iv)} \\ 0 & \textit{(v)} \end{cases}.$$

Comme précédemment, le calcul par programmation dynamique permet ainsi de déduire le score de similarité $s^+_{k\,i,j}$ à partir d'une optimisation locale entre (i) le score $s^+_{k\,i-1,j}$ amoindri d'un score de suppression du symbole $i$ de la séquence $u$ transposée, (ii) le score $s^+_{k\,i,j-1}$ amoindri d'un score d'insertion du symbole $j$ de

$v$, (iii) le score $s^+_{k\,i-1,j-1}$ additionné à un score de substitution (correspondance ou remplacement) du symbole $i$ de $u$ transposée par le symbole $j$ de $v$, (iv) les scores $s^+_{l\,i-1,j}$ pour tout $l \in [\![1,B]\!]\backslash\{k\}$ amoindris d'un score de transposition, et (v) la condition d'alignement local 0 qui permet de débuter un nouvel alignement. La preuve de cet algorithme est indiquée par [AFHI07] comme directement liée à celle donnée par Gusfield dans [Gus97, p.233-234].

Le score de similarité locale recherché $s^*_\gamma(u,v)$ de $u$ et $v$ est donné par :

$$s^*_\gamma(u,v) = \max_{(i,j,k)\in[\![1,n]\!]\times[\![1,m]\!]\times[\![1,B]\!]} s^+_{k\,i,j} \quad . \tag{20}$$

Comme précédemment, la complexité de ce calcul par programmation dynamique est de $\mathcal{O}(mn)$ en temps et en espace. Cependant, l'introduction de la robustesse aux transpositions locales induit une multiplication du nombre d'opérations effectuées d'une constante $B$ correspondant au nombre de transpositions possibles pour chaque symbole. Si cette constante ne change pas la complexité théorique, elle peut s'avérer problématique pour l'application pratique de cet algorithme.

Pour plus de détails sur l'implémentation optimale de cette méthode, nous invitons le lecteur à consulter [AFHI07].

## 3.2 Application à la similarité musicale

Le système de comparaison introduit dans la section précédente permet d'estimer la similarité entre des séquences quelconques. Dans cette section, nous proposons d'évaluer la pertinence de ce système dans le cadre de l'analyse musicale en l'utilisant pour l'estimation de la similarité entre séquences de descripteurs tonaux dans le cadre applicatif de la recherche de reprises.

### 3.2.1 Recherche de reprises

La recherche de reprises est une application suscitant un intérêt croissant de la communauté scientifique d'analyse de la musique [SGH10]. La place importante de celle-ci dans la production musicale ainsi que la multiplication de réinterprétations sur des plateformes sociales populaires telles que *Youtube*[1] ou spécialisées dans le recensement de reprises comme *SecondHandSongs*[2] incitent en effet à développer des systèmes d'indexation automatique des reprises.

La *reprise* musicale peut être définie comme toute version, interprétation, ou enregistrement d'une œuvre musicale déjà enregistrée [Lar98], cette dernière étant appelée version *canonique*. Les reprises sont très employées par exemple dans le but de traduire une œuvre dans un langage différent de celui d'origine, pour introduire un nouvel artiste, pour adapter le style musical d'un morceau, ou encore pour le simple plaisir d'interpréter une chanson connue [SGH10]. Pour l'analyse musicale, la dénomination de reprise varie selon les études de la littérature [TYW08, GÓ6, SGH10]. Dans cette thèse, on associe à cette notion une définition perceptive très large : une reprise d'un morceau est un autre morceau pouvant être identifié comme tel par une majorité d'auditeurs humains. Par exemple, tous les termes suivants peuvent être assimilés à des reprises : enregistrement concert, réinterprétation, version acoustique,

1. http://www.youtube.com
2. http://www.secondhandsongs.com

version instrumentale, version *a capella*, *remix*, pot-pourri, *re-master*, adaptation, parodie *etc.* (voir [SGH10] pour une description précise de la plupart de ces termes). L'ensemble des reprises dérivant de la même version canonique est alors désigné par le terme de *classe* de reprises. La *version* désigne un type particulier de reprises. Deux versions sont deux interprétations différentes d'une même musique [SGHS08]. Les différentes versions d'un morceau présentent ainsi des caractéristiques musicales proches, alors que les reprises au sens large sont plus susceptibles de différer de leur morceau canonique.

Les reprises et versions d'un titre peuvent prendre de nombreuses formes et présenter des signaux musicaux très variés. Néanmoins, lorsque l'œuvre canonique ou l'une de ses reprises est connue, notre perception nous permet généralement d'identifier très aisément une reprise de manière non équivoque et indépendante du contexte [SGH10]. Il est donc possible de constituer une *vérité terrain* décrivant des classes de reprises. Pour cette raison, l'identification de reprises présente l'intérêt d'être une application de la similarité musicale pouvant être évaluée en pratique.

Les mécanismes cognitifs mis en jeu lors de l'identification d'une reprise par un auditeur ne sont à ce jour pas clairement identifiés [SGH10]. Cependant, il est établi que le signal audio d'une reprise présente de nombreuses caractéristiques musicales communes avec le morceau d'origine permettant d'identifier les deux enregistrements comme proches [Lev08]. Ainsi, plusieurs méthodes proposent de retrouver les reprises à partir d'une estimation des caractéristiques musicales partagées par les morceaux. En particulier, la similarité tonale semble jouer un rôle important dans le processus d'identification de reprises [SGHS08, Whi60]. D'une manière générale, une hypothèse-clé pour cette identification consiste à supposer qu'une reprise identifiable par un auditeur humain possède des variations harmoniques et mélodiques similaires avec celles de la version canonique.

### Problématique et travaux antérieurs

Dans une première approche, la recherche de reprises peut être assimilée à un problème d'identification des morceaux dérivés d'un morceau canonique :

*À partir d'un morceau $\mathcal{M}$ (la requête), identifier tous les morceaux de la base de données correspondant à des reprises de $\mathcal{M}$.*

Cependant, la recherche de reprises est généralement formulée d'une manière plus ouverte comme un problème de classification de tous les morceaux de musique d'une base de données :

*À partir d'un morceau $\mathcal{M}$ (la requête), identifier tous les autres morceaux de la base de données correspondant à des reprises du morceau canonique dont $\mathcal{M}$ est lui-même une reprise.*

Cette seconde formulation du problème est celle utilisée dans le cadre d'évaluations standards, telles que le *Music Information Retrieval Evaluation eXchange*[1] (MIREX) [DBEJ08]. Ainsi définie, l'identification des reprises peut alors être ramenée à un problème de recherche par similarité entre tous les morceaux d'une base de données [SGH10]. Cette deuxième formulation du problème permet en outre de correspondre à un cadre d'utilisation pratique plus large que la première, un utilisateur

1. http://www.music-ir.org/mirexwiki

pouvant obtenir la classe de reprises à laquelle se rapporte un morceau quelconque sans nécessairement connaître sa forme canonique.

Les méthodes d'identification de reprises existantes emploient différentes techniques afin d'identifier des similarités musicales en présence de fortes variations tonales, timbrales, rythmiques ou encore structurelles [SGH10].

La robustesse aux changements tonaux et timbraux est généralement assurée par le calcul d'un descripteur mélodique [Mar06, SD06] ou tonal (chroma) [EP07, GH06, JCEJ08, KN08] permettant d'isoler les tons joués du reste du signal, adjoint à une représentation ou une technique de comparaison relative rendant possible l'identification de reprises transposées ou désaccordées [SGH10].

La robustesse aux variations liées à la temporalité du signal (comme le tempo et le rythme) est assurée dans les méthodes existantes par différentes techniques :

- L'une d'entre elles consiste à calculer des contractions et expansions temporelles dans les signaux comparés [KM08, MKC05]. En ré-échantillonnant le signal à différentes échelles temporelles musicalement plausibles, plusieurs représentations peuvent être comparées et le meilleur compromis choisi pour estimer la similarité entre deux morceaux sur la même unité de temps [SGH10]. Cette technique présente cependant l'inconvénient de requérir un nombre élevé de calculs, chaque distorsion de l'échelle temporelle exigeant une nouvelle estimation des descripteurs.
- Une alternative à cette technique consiste à estimer le tempo afin de calculer des descripteurs sur une même unité de temps [EP07, Mar06, NKM02]. Les séquences descriptives peuvent alors être comparées sur une représentation indépendante du tempo.
- Une troisième stratégie consiste à utiliser des techniques d'alignement de séquences afin de calculer les variations temporelles en éditant les séquences représentatives (voir section précédente). Plusieurs systèmes d'identification des reprises mettent en avant la robustesse de cette technique face à de fortes altérations de la temporalité des séquences musicales [SGHS08, Bel07, GH06, GÓ6, Mar06]. En outre, des études de Serrà *et al.* [SGHS08] ou Bello [Bel07] rapportent une performance significativement meilleure de cette approche d'utilisation de techniques d'alignement par rapport à la précédente de calcul et prise en compte du tempo.

Enfin, la robustesse aux variations structurelles est prise en compte dans les systèmes les plus performants [SGH10, p.14][DBEJ08]. Un résultat majeur est l'amélioration significative des résultats d'identification des reprises en considérant une mesure *locale* de la similarité [SGHS08]. Ainsi, il semble plus efficace de ne considérer que l'extrait le plus ressemblant dans deux reprises de la même version canonique pour établir un score de similarité pertinent. Cette approche a été mise en œuvre avec succès dans plusieurs systèmes d'identification de reprises [SGHS08, SSA09, Yan01, MHRF11b]. Une autre amélioration notable de la prise en compte de la structuration des reprises consiste à exploiter le résultat d'une inférence de structures répétitives avant de comparer les morceaux entre eux, afin de les indexer ou de les résumer [GH06, Mar06, MHRF11a]. Une telle approche se basant sur une structure répétitive particulière est présentée et évaluée en Section 4.2.5.

### 3.2.2 Évaluation

À l'instar de la technique proposée par Serrà *et al.* [SGHS08], notre système d'identification des reprises, décrit dans cette section, est basé sur une estimation de la similarité entre morceaux de musique en utilisant les techniques d'alignement local introduites précédemment. L'objectif de cette section est d'évaluer dans notre cadre expérimental spécifique la performance d'un système d'identification de reprises basé sur les algorithmes d'alignement.

#### Technique de comparaison

Soient $u$ et $v$ deux séquences. Le score de similarité entre $u$ est $v$, noté $\mathrm{sim}(u,v)$, est obtenu par le calcul :

$$\mathrm{sim}(u,v) = s^*_\gamma(u,v) \tag{21}$$

Le score ainsi obtenu présente une robustesse aux transpositions locales entre $u$ et $v$.

Le schéma de scores de pondération employé pour ce calcul est le suivant :

| | |
|---|---|
| Insertion, suppression : | $\lambda(h_1,\phi) = \lambda(\phi,h_2) = -0.5$ |
| Substitution : | $\lambda(h_1,h_2) = \lambda_{\mathrm{chr}}(h_1,h_2)$ (voir Éq. 15) |
| Correspondance : | $\lambda_+ = 1$ |
| Remplacement : | $\lambda_- = -0.7$ |
| Transposition locale : | $\lambda_\mathsf{T} = -10$ |

(22)

La détermination de ce schéma de scores est effectuée de manière empirique sur un sous-ensemble de la base de données présentée ci-dessous. Celui-ci est composé de 2 morceaux choisis aléatoirement dans chaque classe de reprises, pour un total de 32 morceaux. Une légère variation des scores de correspondance et de remplacement d'une part, de suppression et d'insertion d'autre part ne provoque pas de baisse significative des mesures d'évaluation. Un constat similaire a déjà été effectué par Serrà *et al.* [SGHS08]. La valeur du score de transposition $\lambda_\mathsf{T}$ est également déterminée de manière empirique, en considérant quelques paires de reprises incluant des transpositions locales, et en optimisant les facteurs alignés afin que ceux-ci correspondent à des sections similaires dans chacune des paires de reprises.

#### Bases de test

Afin d'évaluer notre système de recherche par similarité, on considère une base de données audio formée de trois ensembles extraits de collections personnelles. Cette base de données, notée $TSD$, est constituée de 2514 morceaux distincts. Le Tableau 3.2 liste l'ensemble des classes considérées, ainsi que le nombre de versions et de reprises dont elles sont composées.

- Le premier ensemble, noté $\mathcal{D}_\mathrm{V}$, contient 7 classes de versions comprenant chacune entre 4 et 10 morceaux correspondant à la même version canonique, pour une moyenne de 6 morceaux par classe. Cette base de données est ainsi constituée afin d'évaluer la robustesse de la recherche par similarité sur des versions peu altérées d'un morceau d'origine.

| | Morceau de référence | Taille classe |
|---|---|---|
| A | Aha - *Take on me* | 8 v, 8 r |
| B | The Beatles - *Yesterday* | 5 v, 39 r |
| C | The Animals - *The House of the Rising Sun* | 5 v, 85 r |
| D | Ben E. King - *Stand By Me* | 4 v, 23 r |
| E | Pachelbel - *Canon in D* | 10 v, 34 r |
| F | H. Mancini - *The Pink Panther* | 9 v, 27 r |
| G | T. Wynette - *Stand By Your Man* | 4 v, 17 r |
| H | The Beatles - *Across The Universe* | 0 v, 15 r |
| I | Nirvana - *Smells Like Teen Spirit* | 0 v, 21 r |
| J | G. Jones - *Tainted Love* | 0 v, 29 r |
| K | B. Dylan - *All Along The Watchtower* | 0 v, 14 r |
| L | Big Joe Williams - *Baby Please Don't Go* | 0 v, 31 r |
| M | A. Villoldo - *El Choclo* | 0 v, 30 r |
| N | C. Granda - *La Flor De La Canela* | 0 v, 44 r |
| O | S. Linda - *The Lion Sleeps Tonight* | 0 v, 30 r |
| P | M. Reynolds - *Little Boxes* | 0 v, 41 r |
| Q | R. Thomas - *Walking the Dog* | 0 v, 26 r |

TAB. 3.2 – *Classes de reprises et de versions de la base TSD utilisées pour les évaluations. La dernière colonne indique le nombre de versions (v) et de reprises (r) appartenant à chacune des classes.*

- Le deuxième ensemble, noté $\mathcal{D}_{\mathrm{r}}$, contient 17 classes de reprises comprenant chacune entre 8 et 85 morceaux correspondant à la même version canonique, pour une moyenne de 30 morceaux par classe. Cette base de données est ainsi constituée afin d'évaluer la robustesse de la recherche par similarité sur des reprises très diverses, celles-ci étant choisies d'une manière très variée et conforme à la définition de la reprise exposée précédemment.
- Le troisième ensemble comprend un ensemble de 2000 morceaux choisis de manière arbitraire parmi des œuvres de styles musicaux similaires aux reprises et versions des deux autres bases de données. Cette base est constituée afin de prouver la robustesse du système de recherche par similarité en présence d'un contenu musical homogène.

### Méthode d'évaluation

Comme souligné par Serrà *et al.* [SGH10], l'évaluation des systèmes de recherche de reprises est une tâche complexe dont la méthodologie varie en fonction des études proposées dans la littérature. La méthode d'évaluation décrite dans cette section suit la seule tentative de standardisation proposée jusqu'ici, établie dans le cadre du *Music Information Retrieval Evaluation eXchange* (MIREX) [1] [Dow08, DBEJ08].

Soit $\mathcal{C}$ une classe de reprises de $\mathcal{D}_{\mathrm{r}}$. Chaque élément $i$ de $\mathcal{C}$ est soumis comme requête du système de recherche par similarité. On obtient donc une liste $\mathcal{L}_i$ de scores de similarité pour chacun de ces éléments indiquant un degré de similarité entre ces morceaux et la requête.

1. http://www.music-ir.org/mirexwiki

| Classe | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| Nb éléments | 8 | 39 | 85 | 23 | 34 | 27 | 17 | 15 | 21 |
| MAP | 74.3 | 86.5 | 82.3 | 45.2 | 90.4 | 17.2 | 77.7 | 80.9 | 56.0 |
| R-précision | 71.4 | 81.6 | 75.0 | 45.5 | 78.8 | 11.5 | 68.8 | 78.6 | 50.0 |

| Classe | J | K | L | M | N | O | P | Q | Tout |
|---|---|---|---|---|---|---|---|---|---|
| Nb éléments | 29 | 14 | 31 | 30 | 44 | 30 | 41 | 26 | 514 |
| MAP | 54.9 | 50.19 | 22.3 | 72.14 | 13.0 | 57.8 | 53.7 | 28.7 | 56.7 |
| R-précision | 53.6 | 53.8 | 20.0 | 69.0 | 11.6 | 55.2 | 50.0 | 32.0 | 53.3 |

TAB. 3.3 – *Résultats de l'identification de reprises (en pourcentages) par recherche de similarité avec alignement local sur $TSD$.*

La première métrique que nous utilisons est une variante de la technique rappel-précision à différents rangs [MRS08]. Pour un rang donné $k$, on relève le nombre d'éléments appartenant à $\mathcal{C}$ parmi les $k$ morceaux les plus similaires identifiés dans $\mathcal{L}_i$, et on calcule les taux de rappel et précision. La répétition de cette technique pour tous les rangs possibles donne une information précise sur la qualité du système d'identification [MRS08].

La précision moyenne, ou *Mean average precision* (MAP), correspond à la moyenne sur l'ensemble des classes de reprises des scores moyens obtenus à partir des valeurs de précision aux différents rangs possibles [MRS08]. Cette évaluation a l'avantage de présenter une mesure de la qualité d'un système de recherche par similarité sous la forme d'une unique valeur.

Une troisième métrique régulièrement employée est la *R-précision*. Celle-ci est obtenue pour une classe $\mathcal{C}$ comportant $N$ reprises en calculant le rappel parmi les $N$ morceaux les plus similaires, valeur alors égale à la précision au même rang. Cette valeur peut être calculée pour chacune des classes de reprises et moyennée afin d'obtenir à nouveau une mesure de la qualité d'un système de recherche par similarité. Il convient de noter qu'en pratique la R-précision moyenne et la valeur MAP semblent corrélées [MRS08].

### Résultats

Les résultats d'un système d'identification des reprises dépendent très fortement de la taille de la base de données, du type de reprises et des styles musicaux considérés [SGH10]. Par conséquent, notre objectif n'est pas ici d'évaluer la performance de notre système par rapport à l'état de l'art, mais d'évaluer sa précision et la rapidité de son exécution en guise d'indicateurs de référence. Ces indicateurs sont utilisés dans la suite du document comme une base d'améliorations de la recherche par similarité, ainsi que décrites en Sections 3.3.4 et 4.2.5.

Le Tableau 3.3 présente les résultats obtenus sur chacune des classes de reprises de $TSD$. Les mesures MAP et R-précision sont indiquées. Comme dit précédemment, ces deux mesures semblent en pratique suivre une distribution proche pour chaque classe.

On remarque que les mesures de précision varient en fonction des classes de reprises. Par exemple, pour la classe *Yesterday* (B), le système semble correctement

identifier les reprises avec une précision moyenne de 86.5% et une R-précision de 81.6% ; en revanche, la classe *The Pink Panther* (F) est identifiée avec une précision beaucoup plus faible, avec une valeur MAP de 17.2% et une R-précision de 11.5%.

La précision générale du système peut être représentée par la moyenne des précisions moyennes sur les différentes classes de reprises. Ainsi, le système a une valeur MAP globale de 56.7% et une R-précision de 53.3%. Il convient de noter que si ces mesures permettent d'avoir un aperçu général de l'efficacité du système, il est important en pratique de dissocier chacune des classes de reprises, dont la précision d'identification peut fortement varier.

Le temps d'exécution des calculs de similarité locale sur $TSD$ requis pour l'obtention de ces résultats est de plus de 18 heures sur notre configuration matérielle [1], pour un temps moyen d'exécution constaté de 129 secondes par requête. Cette durée élevée impose une forte contrainte sur l'utilisation d'un tel système dans une application pratique. En particulier, appliquer l'analyse sur une base de données sensiblement plus grande que $TSD$, de l'ordre du million de titres, semble inenvisageable.

Afin de permettre à notre méthode d'être utilisable en pratique, il est donc primordial de la *passer à l'échelle* des bases de données disponibles à l'heure actuelle.

## 3.3 Passage à l'échelle

Les techniques d'alignement permettent une évaluation précise de la similarité entre des séquences musicales. Cependant, comme expliqué en Section 3.1.4.4, elles sont calculables en un temps proportionnel au produit des tailles des séquences comparées. Cette propriété peut s'avérer problématique dès lors que les séquences musicales comparées sont de grandes longueurs, ou encore dès que de nombreux alignements doivent être calculés rapidement à travers des bases de données de taille conséquente.

Afin de permettre une utilisation pratique des systèmes de comparaison musicale, de nombreuses approches proposent d'améliorer l'efficacité calculatoire des méthodes d'estimation de la similarité (voir [Sch12] pour une revue des systèmes les plus utilisés). Cependant, peu d'approches visant une grande efficacité calculatoire considèrent les techniques d'alignement pour la comparaison musicale.

La comparaison de séquences biologiques, fortement basée sur les techniques d'alignement, est pourtant évaluée sur de très grandes bases de données, de l'ordre de millions de séquences [KYB03]. La mise en place de méthodes de calculs rapides d'alignement est donc critique pour ce type d'évaluations.

La technique *Basic Local Alignment Search Tool* [AGM+90], ou BLAST, répond à ce besoin en proposant une méthode heuristique d'évaluation efficace de l'alignement local. BLAST a été adaptée à de nombreux contextes applicatifs différents en bio-informatique, menant à la définition de plusieurs outils spécialisés pour différentes tâches communes de séquençage biologique. En particulier, BLASTN permet de comparer des séquences d'ADN pour l'analyse génomique, alors que BLASTP, BLASTX, TBLASTN et TBLASTX comparent des séquences de protéines notamment pour l'analyse génomique ou phylogénétique [KYB03].

L'approche présentée ici explore un nouveau champ applicatif, la comparaison

1. Processeur Intel Xeon X5675 à 3.07 GHz, 12M mémoire cache, 32Go mémoire RAM

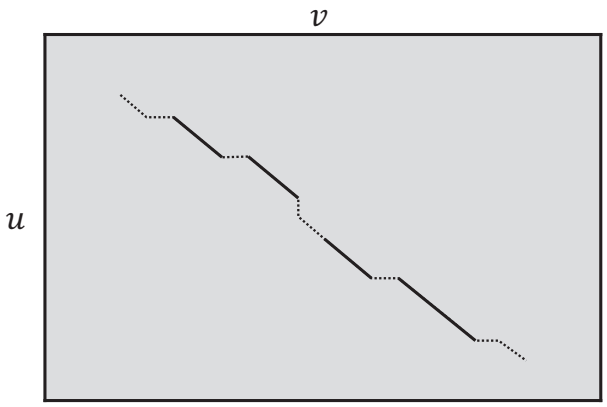


FIG. 3.3 – *Espace de recherche entre $u$ et $v$. Le chemin de transcription optimale est représenté en pointillés. Les séries de correspondances successives sont matérialisées par des traits pleins. La surface grise représente l'ensemble des coefficients de programmation dynamique à calculer pour obtenir le chemin optimal.*

de séquences de descripteurs audio musicaux. Si la méthode BLAST a déjà été appliquée avec succès pour évaluer la similarité entre extraits de musique symbolique [KH04], son utilisation pour l'indexation de signaux audio musicaux, suggérée en perspective dans [Kil04], n'a à notre connaissance pas été mise en œuvre jusqu'ici [MBHF12].

### 3.3.1 Principe de BLAST

Afin de décrire le principe de BLAST, on considère un cas pratique d'utilisation de techniques de comparaison pour analyser une grande base de données de séquences musicales. Étant donné une séquence-requête, on souhaite identifier dans la base de données toutes les séquences *localement similaires*. Avec les outils présentés précédemment, une solution mise en œuvre en Section 3.2.1 consiste à calculer la similarité locale de la séquence-requête avec chacune des séquences de la base de données, puis à choisir le meilleur de ces alignements.

L'amélioration apportée par BLAST repose sur une hypothèse d'hétérogénéité de la base de données, qui peut être résumée par l'assertion suivante : il est probable qu'une grande majorité des séquences de la base de données soient très différentes de la requête. Ainsi, il est probable qu'un temps de calcul important soit consacré à l'évaluation précise d'alignements non pertinents. Pour éviter de tels calculs inutiles, la technique BLAST peut être implémentée comme un *filtre* à plusieurs niveaux sur les séquences comparées. Chaque niveau de filtrage agit alors comme une méthode heuristique qui élimine les séquences les moins similaires de la base de données dans une évaluation nécessitant un nombre faible de calculs. La section suivante décrit en détail ces différents niveaux de filtrage, ainsi que le gain en nombre d'opérations induit par l'indexation.

### 3.3.2 Méthode d'indexation

Soient $u$ et $v$ deux séquences de tailles respectives $n$ et $m$ définies sur un alphabet $\Sigma$. On désigne par *espace de recherche* entre $u$ et $v$ l'ensemble des paires d'indices dans $u$ et $v$, de taille $nm$. L'espace de recherche comprend toutes les positions

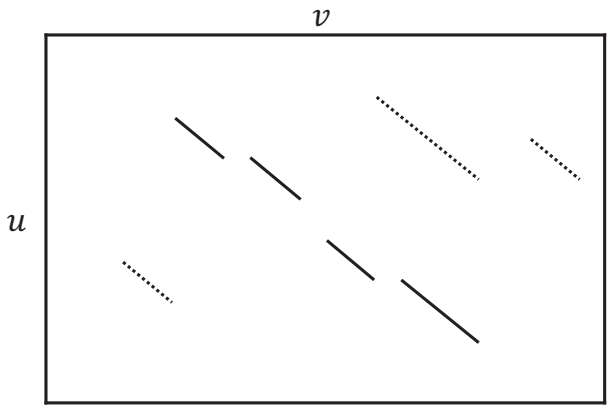


FIG. 3.4 – *Graines de l'espace de recherche pour une taille minimale fixée. Les graines significatives appartiennent à l'alignement optimal, et sont représentées par des traits pleins. Les graines non significatives n'appartiennent pas à l'alignement optimal et sont matérialisées par des pointillés.*

desquelles peuvent démarrer ou terminer tout alignement entre $u$ et $v$. L'objectif de l'indexation est d'évaluer la similarité locale entre $u$ et $v$ à partir de l'espace de recherche sans avoir à calculer l'ensemble du graphe d'édition.

Dans une première étape, la technique BLAST consiste à repérer et indexer rapidement des régions de similarité exacte de l'espace de recherche de $u$ et $v$, telles que décrit dans la section suivante.

#### 3.3.2.1 Repérage de graines d'alignement

La Figure 3.3 montre l'espace de recherche des séquences $u$ et $v$. Le chemin de la transcription optimale pour l'alignement local entre $u$ et $v$ est représenté en pointillés. Comme expliqué précédemment, le calcul de cette similarité locale nécessite le calcul de l'intégralité du graphe d'édition. Le fond grisé de la Figure 3.3 souligne ainsi le nombre de cases de programmation dynamique nécessaires à ce calcul.

Le premier filtre BLAST, tel que défini par Altschul *et al.* [AGM+90] en 1990 pour l'alignement de séquences biologiques, repose sur la supposition que tout alignement local entre séquences similaires inclut de nombreuses séries de correspondances successives. Chaque section diagonale du chemin représenté sur la Figure 3.3 indique une série de substitutions successives entre $u$ et $v$. Parmi ces substitutions, les séries de correspondances successives sont représentées par des traits pleins.

On définit un entier naturel positif $W$ pour désigner une taille arbitraire minimale de correspondances successives. La première étape de BLAST consiste à repérer toute paire de facteurs identiques dans $u$ et $v$ de taille au moins $W$. En d'autres termes, l'algorithme détecte toute section diagonale dans le graphe d'édition de $u$ en $v$ correspondant à une série de correspondances de taille supérieure à $W$. Chacune de ces séries de correspondances étant susceptible de faire partie de l'alignement local optimal entre $u$ et $v$, cette étape permet de repérer des points de départ potentiels pour déclencher le calcul d'alignement des séquences. Chaque série de correspondances de taille suffisante est ainsi appelée *graine* de l'espace de recherche.

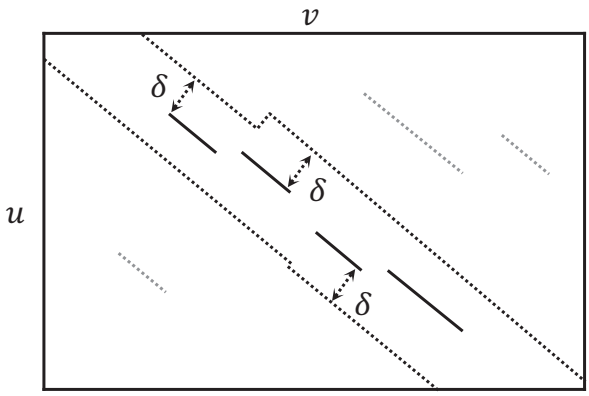


FIG. 3.5 – *Trajectoire pseudo-diagonale. Les graines en traits pleins sont organisées selon une trajectoire pseudo-diagonale comprise dans le bandeau représenté en pointillés.*

Le premier filtre consiste à identifier rapidement ces séries de correspondances successives, et ainsi à éliminer toutes les comparaisons ne présentant pas suffisamment de graines. La Figure 3.4 illustre la sélection automatique des graines d'une taille minimale $W$ sur l'espace de recherche de $u$ et $v$. Parmi les graines identifiées, certaines, marquées en traits pleins, correspondent à un extrait de l'alignement optimal $t$ ; elles sont dites *significatives*. À l'inverse, des graines marquées en traits pointillés sont localisées dans des régions étrangères à l'alignement optimal ; elles sont dites *non significatives*.

Le premier filtre appliqué par BLAST, noté $\mathcal{F}_1$, est donc caractérisé par l'heuristique suivante :

*Deux séquences similaires ont un nombre de graines significatives de taille $W$ plus élevé que deux séquences dissimilaires quelconques*

La seconde étape de BLAST consiste à filtrer l'ensemble des graines identifiées afin d'écarter les graines non significatives.

#### 3.3.2.2 Sélection de graines

Le deuxième filtre BLAST présenté dans cette section a été introduit dans le cadre de cette thèse pour l'indexation de données musicales [MBHF12]. Il a pour but d'éliminer les graines n'appartenant pas à l'alignement local optimal. Une solution simple consiste à regrouper les graines de l'espace de recherche en fonction de leurs positions relatives, tentant ainsi de reconstruire le chemin le plus probable pour l'alignement local.

Une première propriété des séries de correspondance appartenant à un alignement local entre $u$ et $v$ est leur disjonction dans $u$ d'une part, et leur disjonction dans $v$ d'autre part. Ainsi, seule une graine par colonne et seule une graine par ligne de l'espace de recherche peut correspondre à un alignement local. Cette propriété permet de distinguer facilement des graines n'appartenant pas à un même chemin.

Afin d'établir cette seconde propriété, on considère $t$ la transcription d'alignement optimal entre les séquences $u$ et $v$, et $\mathcal{C}$ le chemin correspondant dans le graphe d'édition. Par définition, $t$ est formée :

- De séries de substitutions, se traduisant par des sections diagonales dans $\mathcal{C}$,

- De séries d'insertions, se traduisant par des sections horizontales dans $\mathcal{C}$,
- De séries de suppressions, se traduisant par des sections verticales dans $\mathcal{C}$.

Si $u$ et $v$ sont similaires, on formule l'hypothèse que chaque série d'insertions et chaque série de suppressions a une longueur faible. En effet, dans ce cas, $t$ est susceptible de comporter de multiples séries de correspondance témoignant de la forte similarité des séquences comparées, séparées de courtes opérations de décalage dues à l'insertion ou la suppression de symboles. Graphiquement, $\mathcal{C}$ est composé d'une série de courtes diagonales et suit une trajectoire qui dévie légèrement à chaque nouvelle série de correspondances analysée.

Plus précisément, pour un entier positif $k$ et un chemin $\mathcal{C}$ du graphe d'édition de deux séquences, on dit que $\mathcal{C}$ suit une trajectoire *$k$-pseudo-diagonale* s'il s'inscrit dans un bandeau de largeur $k$ centré sur ses graines. Dans ce cas, on dit que les graines du chemin $\mathcal{C}$ sont *organisées selon une trajectoire pseudo-diagonale.*

La Figure 3.5 représente ainsi un ensemble de graines (en traits pleins) organisées selon une trajectoire pseudo-diagonale située dans le bandeau (en pointillés) de largeur constante centré sur ces graines.

Cette observation heuristique est mise à profit dans le cadre d'un filtre BLAST en introduisant un entier naturel positif $K$ qui décrit la largeur de bandeau considérée pour le regroupement des graines. Ce seuil correspond donc à la déviation maximale autorisée entre deux graines successives pour les considérer membres d'un même groupe. Ainsi, toutes les graines repérées par $\mathcal{F}_1$ sont regroupées selon des bandeaux pseudo-diagonaux de taille $K$. D'après notre hypothèse, on considère alors que le bandeau contenant le plus de graines contient l'alignement optimal des séquences comparées. Par conséquent, à l'issue de ce filtre noté $\mathcal{F}_2$, seules sont conservées les graines appartenant à ce bandeau optimal.

Le filtre $\mathcal{F}_2$ peut ainsi être résumé par l'heuristique suivante :

*Les graines de l'espace de recherche de deux séquences similaires sont organisées selon une trajectoire pseudo-diagonale*

Les graines de l'espace de recherche identifiées à l'issue du filtre $\mathcal{F}_2$ sont supposées correspondre à la région contenant le plus probablement l'alignement optimal. Le troisième filtre BLAST, plus précis, permet de déterminer la pertinence du groupe de graines identifié, et ainsi d'estimer la probabilité que celui-ci correspond effectivement à un alignement optimal de séquences similaires.

#### 3.3.2.3 Extension des graines

L'étape d'extension de graines présentée dans cette section est un raffinement de la méthode BLAST introduit par Altschul *et al.* [AMS+97] en 1997 pour les séquences biologiques.

Soient $u$ et $v$ les deux séquences comparées de tailles respectives $n$ et $m$. L'étape précédente a permis d'identifier les graines les plus significatives de l'espace de recherche. Le groupe de graines ainsi identifié est supposé être proche de l'alignement local optimal des deux séquences comparées. Ce troisième filtre permet de s'assurer de la pertinence de cet ensemble de graines en déclenchant de courts alignements autour de chacune d'entre elles. Cette opération de transformation d'une graine en une portion d'alignement est appelée *extension.*

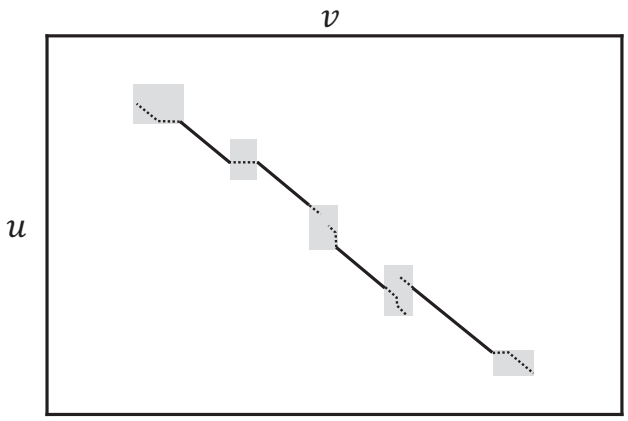


FIG. 3.6 – *Résultat de l'extension des graines significatives. Les pointillés indiquent les extensions optimales calculées. Les régions grisées matérialisent les coefficients de programmation dynamique devant être évalués.*

Soit $g$ une graine de l'espace de recherche débutant aux coordonnées $(i,k)$ et $(j,l)$ ; on a alors $u[i \dots j] = v[k \dots l]$. On introduit un seuil d'extension $X$ entier et positif.

*Étendre* $g$ vers la fin des séquences $u$ et $v$ peut être obtenu en calculant un alignement optimal entre $u[j \dots n]$ et $v[l \dots m]$ débutant en $(j,l)$. Afin d'éviter un alignement trop coûteux, BLAST impose que l'extension soit stoppée dès que le score de similarité devient trop faible. Plus précisément, lors de l'extension d'une graine, si la valeur de similarité $s$ d'un coefficient de programmation dynamique passe au-dessous du seuil $s_{max} - X$, où $s_{max}$ est le score maximal de similarité calculé pour le chemin en cours, alors l'extension est stoppée [AMS+97]. L'extension de score maximal qui respecte cette condition est alors appelée *extension droite* de la graine $g$.

De manière analogue, la graine $g$ est étendue à partir de $(i,k)$ vers le début des séquences en un alignement optimal respectant la condition d'arrêt, définissant alors l'*extension gauche* de la graine $g$.

Pour chaque graine significative, les extensions gauche et droite sont calculées. Le score correspondant à l'alignement ainsi généré à partir de la graine $g$, appelé *score d'extension* de $g$, est défini comme la somme des scores des deux extensions. Finalement, le *score d'indexation* de l'espace de recherche de $u$ et $v$ est obtenu en sommant les scores d'extension des graines précédemment identifiées comme significatives.

La Figure 3.6 représente un exemple d'extension des graines significatives de l'espace de recherche. Les régions grisées correspondent aux coefficients de programmation dynamique devant être calculés pour évaluer les extensions. On remarque qu'à l'issue de ce filtre, l'ensemble de chemins identifiés est proche du chemin optimal de l'alignement local, représenté en Figure 3.3. Cependant, dans le cas de la Figure 3.6, un nombre sensiblement inférieur de coefficients de programmation dynamique, en gris, ont dû être évalués.

En introduisant un seuil $S$ de sensibilité d'indexation, le troisième filtre, noté $\mathcal{F}_3$, consiste à ne conserver parmi toutes les comparaisons effectuées que celles dont le score d'indexation est supérieur à $S$.

Le filtre $\mathcal{F}_3$ est caractérisé par l'heuristique suivante :

*La somme des scores d'extension de graines significatives de deux séquences est proche du score d'alignement local optimal de ces séquences*

### 3.3.3 Indexation de séquences tonales

Afin d'appliquer les principes de BLAST aux séquences musicales, il convient d'abord d'analyser leur distribution et d'optimiser les réglages heuristiques à ces données. Cette section décrit la stratégie d'indexation de séquences tonales, et expose les résultats expérimentaux menant à une indexation efficace dans le cadre applicatif de l'identification des reprises. L'étude présentée dans cette section a fait l'objet d'une publication dans les actes de la conférence *International Society for Music Information Retrieval* (ISMIR) 2012 [MBHF12] [1].

#### 3.3.3.1 Contraintes de représentation

Comme expliqué en Chapitre 2, le chroma est un vecteur comprenant $B$ valeurs indépendantes représentant l'information tonale d'un extrait audio. L'espace de définition du symbole chroma est donc l'espace non fini

$$\Sigma_{\text{chr}} = \mathbb{R}^B. \tag{23}$$

Comme expliqué dans la section précédente, l'indexation effectuée par BLAST repose sur la détection rapide de graines de l'espace de recherche, donc de correspondances exactes entre symboles. Il est donc crucial pour l'indexation de représenter les symboles sur un alphabet fini et de faible taille, afin d'identifier des correspondances exactes dans les séquences musicales.

Une possibilité de réduction de l'espace de représentation des chromas à un alphabet fini consiste à transformer ceux-ci en accords. Ainsi, de nombreuses études de la littérature proposent des techniques de conversion de chromas en accords, par exemple en utilisant une analyse par profils de référence (voir [Roc11, p.71–76] pour une revue des techniques existantes). Ces techniques sont généralement dédiées à l'inférence précise des accords joués, et évalués en comparant ceux-ci à des annotations manuelles pour différentes bases de données audio [Roc11]. Cependant, dans le cadre de BLAST, si cette quantification sur l'espace des accords doit nécessiter peu de calculs et conserver une pertinence musicale, une grande précision n'est pas requise. En effet, puisque BLAST permet d'indexer de courtes graines de l'espace de recherche, on peut supposer que la méthode est tolérante à quelques erreurs d'analyse pouvant être introduites par cette quantification.

Une quantification simple consiste à représenter chaque chroma par sa *triade* prépondérante [Gó6], c'est-à-dire en repérant le triplet (tonique, tierce (majeure ou mineure), dominante) [Bit87] dont la somme de coefficients dans le chroma est maximale. On quantifie alors un chroma $h$ en le substituant par la tonique de l'accord correspondant à ce triplet prépondérant.

Cette opération de quantification réduit alors l'espace de description à un alphabet $\Sigma_{\text{acc}}$ de $B$ symboles, que l'on peut représenter comme l'ensemble des toniques possibles sur la gamme tempérée. Par commodité de notation, on associe une

1. Les résultats numériques présentés dans cette section diffèrent légèrement de ceux publiés, les expériences et réglages ayant été approfondis dans le cadre de cette thèse

lettre à chaque tonique possible. Par exemple, pour une résolution de 12 demi-tons ($B = 12$), l'alphabet $\Sigma_{\text{acc}}$ peut être défini comme :

$$\Sigma_{\text{acc}} = \begin{Bmatrix} \texttt{a}, & \texttt{b}, & \texttt{c}, & \texttt{d}, & \texttt{e}, & \texttt{f}, & \texttt{g}, & \texttt{h}, & \texttt{i}, & \texttt{j}, & \texttt{k}, & \texttt{l} \\ \text{C}, & \text{C}\sharp, & \text{D}, & \text{D}\sharp, & \text{E}, & \text{F}, & \text{F}\sharp, & \text{G}, & \text{G}\sharp, & \text{A}, & \text{A}\sharp, & \text{B} \end{Bmatrix}, \tag{24}$$

la seconde ligne indiquant la correspondance entre chaque symbole et la notation musicale usuelle.

Cette quantification des chromas en accords est susceptible d'introduire des erreurs d'analyse. Par exemple, la projection du chroma illustré Figure 2.5-(iv) sur $\Sigma_{\text{acc}}$ induit la suppression d'une partie de l'information tonale, et peut conduire à une mauvaise interprétation de l'harmonie. Ainsi, ce chroma risque d'être interprété sur $\Sigma_{\text{acc}}$ comme un accord simple (Ré ou La), tandis que le reste de l'information tonale est ignoré. En conservant la représentation sous forme de chroma, l'ensemble de l'information est préservé et le descripteur caractérise plus précisément les informations tonales de l'extrait. D'une manière générale, la représentation sur l'espace de définition du vecteur chroma $\Sigma_{\text{chr}}$ permet de conserver toute l'information tonale présente au sein du descripteur, tandis que la projection sur l'alphabet des accords $\Sigma_{\text{acc}}$ ne conserve qu'une partie de cette information. Une estimation de l'erreur introduite par cette quantification peut être consultée en [Gó6, chap.4].

Malgré une perte de précision de la description tonale du signal, une telle projection a plusieurs avantages pour l'application d'indexation :

- Elle permet de définir comme mesure de similarité la comparaison triviale entre symboles ;
- Elle facilite l'interprétation musicale des symboles identifiés ;
- Elle augmente fortement la probabilité de correspondances successives dans l'espace de recherche.

En outre, comme expliqué en Section 3.2.1, la transposition (locale ou globale) est une variation courante que les techniques de comparaison doivent prendre en compte. Ainsi, comme souligné par Kurth et Müller [KM08], une stratégie d'indexation efficace pour la similarité musicale doit identifier des progressions harmoniques *relatives* plutôt que des informations tonales *absolues*. Par exemple, les séquences bcbbfd et deddhd peuvent être considérées comme identiques puisque la seconde n'est qu'une transposition de 2 tons de la première. En pratique, une technique permettant de comparer de manière relative est de transposer chaque séquence avant toute comparaison afin qu'elle commence par un même symbole, a par exemple.

#### 3.3.3.2 Bases de données

Afin d'évaluer l'utilité de la méthode d'indexation, nous considérons un ensemble de morceaux de taille conséquente, de l'ordre du million de morceaux de musique. Deux bases de données sont donc utilisées pour évaluer notre technique. La première, notée $TSD$ et introduite en Section 3.2.2, correspond à un ensemble de 2514 morceaux de musique, comprenant 514 reprises réparties en 17 classes. Celle-ci est utilisée comme témoin de la précision du système d'identification. La seconde, notée $MSD$, est extraite de la base de données *Million Song Dataset*[1] [BMEWL11] et correspond à un ensemble d'un million d'œuvres de différents styles musicaux.

1. http://labrosa.ee.columbia.edu/millionsong/

| | | *TSD* | | *MSD* | |
|---|---|---|---|---|---|
| | Taille graine | Faux négatifs | Faux positifs | Faux négatifs | Faux positifs |
| (i) | 3 | 0.00 | 8.91 | 0.11 | 4.70 |
| | 4 | 0.03 | 3.69 | 4.28 | 1.15 |
| | 5 | 0.84 | 1.68 | 18.3 | 0.39 |
| | 6 | 4.50 | 0.82 | 37.2 | 0.16 |
| | 7 | 11.7 | 0.44 | 54.7 | 0.08 |
| | 8 | 21.8 | 0.25 | 68.6 | 0.04 |
| (ii) | 3 | 0.00 | 1.06 | 0.84 | 1.33 |
| | 4 | 0.21 | 0.41 | 7.08 | 0.23 |
| | 5 | 2.59 | 0.18 | 22.6 | 0.06 |
| | 6 | 6.67 | 0.08 | 41.8 | 0.02 |
| | 7 | 14.9 | 0.04 | 58.9 | 0.01 |
| | 8 | 26.0 | 0.02 | 72.1 | .003 |

TAB. 3.4 – *Compromis sensibilité/spécificité sur les bases de données* TSD *et* MSD. *Les scores sont donnés en pourcentages. Dans (i), tous les mots de la base de données sont indexés. Dans (ii), les mots mono-symboliques ne sont pas indexés.*

Parmi ces morceaux, la base *Second Hand Songs dataset*[1] comprend 18196 reprises réparties en 5854 classes. Cette base de données est à notre connaissance la plus grande collection disponible de descripteurs audio calculés à partir de reprises.

Pour les morceaux des deux bases de données, la description tonale sous forme de séquences de chromas est utilisée. Néanmoins, il convient de noter que la représentation sous forme de chromas varie pour l'une et l'autre. Pour $TSD$, notre implémentation des chromas telle que décrite en Section 2.2.5 est appliquée avec une taille de trame constante et une résolution $B = 36$. Pour $MSD$, la représentation est obtenue grâce au *framework EchoNest*[2], et est calculée sur des trames dites *segment-synchronized*, de durées non constantes et une résolution $B = 12$, comme décrit plus précisément dans [Jeh05a, p.57–59]. Ainsi, chaque chroma de $MSD$ représente l'information tonale sur 12 dimensions pour une trame de durée variable (entre 80 et 300ms sans chevauchement [Jeh05a]), tandis que chaque chroma de $TSD$ représente l'information tonale sur 36 dimensions pour une trame de durée fixe (743ms avec un chevauchement de moitié).

#### 3.3.3.3 Analyse statistique

Les heuristiques formulées par la technique BLAST reposent sur la sélection de graines, ou séries de correspondances entre deux séquences comparées. Un point clé de la méthode est de déterminer un compromis raisonnable entre une bonne *sensibilité* du filtrage, indexant autant d'alignements optimaux que possible, et une bonne *spécificité* de celui-ci, qui doit introduire aussi peu d'alignements non pertinents que possible. La suite de cette section décrit l'étude statistique permettant d'évaluer ce compromis.

1. http://www.secondhandsongs.com
2. http://the.echonest.com/

### Sensibilité

La sensibilité peut être évaluée en analysant des alignements de séquences similaires. On pose $N$ un entier positif, correspondant à une taille de graine. Un alignement donné ne peut être indexé par BLAST que s'il contient au moins une graine de taille $N$. Connaître la sensibilité du test pour $N$ consiste donc à évaluer combien d'alignements de séquences similaires ne peuvent être indexés par une graine de taille $N$.

Pour chaque classe de reprises $\mathcal{C}$ de $TSD$ et de $MSD$, on calcule les alignements locaux optimaux entre toutes les paires de séquences de $\mathcal{C}$. Chaque alignement est alors considéré comme correctement indexé s'il contient au moins une série de $N$ correspondances successives. Dans le cas contraire, il n'est pas indexé par BLAST et est dit *faux négatif*. Les deuxième et quatrième colonnes du Tableau 3.4-(i) présentent, en fonction de la taille de graine $N$, le taux de faux négatifs correspondant à la probabilité qu'un alignement ne soit pas indexé par la méthode.

### Spécificité

Pour une taille de graine $N$, la spécificité du test correspond à la probabilité de trouver une série de $N$ correspondances dans deux morceaux de séquences choisis aléatoirement dans la base de données.

En pratique, on évalue cette probabilité en comptant le nombre d'occurrences de chacune des séquences de $N$ symboles sur $\Sigma_{\text{acc}}$, ou *mots* de taille $N$, dans la base de morceaux quelconques (ne correspondant pas à des reprises). Les nombres d'occurrences sont calculés pour tous les mots possibles et stockés dans une liste $L$. Ainsi, un élément de $L$ indique le nombre d'instances d'un mot particulier, et $\sum_i L[i]$ est le nombre total de mots de taille $N$ dans la base de données. La probabilité de trouver *un* mot $w$ dans une séquence choisie aléatoirement correspond à la valeur $\frac{L[j]}{\sum_i L[i]}$, où $j$ est l'indice dans $L$ correspondant au mot $w$. La probabilité de trouver *deux* mots $w$ dans deux séquences choisies aléatoirement correspond alors au carré de cette valeur. Par conséquent, la probabilité de trouver *deux* mots identiques de taille $N$ (donc une série de $N$ correspondances) dans des séquences aléatoires de la base de données est donnée par :

$$Pr[\text{faux positif}] = \frac{1}{(\sum_i L[i])^2} \sum_j L[j]^2. \tag{25}$$

Les troisième et cinquième colonnes du Tableau 3.4-(i) présentent les résultats de calcul de cette probabilité en fonction de la taille des mots indexés, pour chacune des bases de données.

### Distribution des mots

Le Tableau 3.5 présente les quinze mots de taille 7 apparaissant le plus souvent dans $TSD$ et $MSD$. Le mot le plus probable est une succession de 7 fois le même symbole, qui apparaît dans 6.36% des séquences dans $TSD$ et 14.1% des séquences dans $MSD$. La probabilité d'apparition de ce mot mono-symbolique est significativement plus élevée que celle de tous les autres. Sa contribution au taux de faux

| *TSD* | | *MSD* | |
|---|---|---|---|
| Mot | % | Mot | % |
| aaaaaaa | 6.36 | aaaaaaa | 14.1 |
| ahhhhhh | 0.55 | aaaaaah | 1.45 |
| aaaaaah | 0.53 | aaaaaaf | 1.31 |
| afffffff | 0.49 | afffffff | 1.13 |
| aaaaaaf | 0.46 | ahhhhhh | 0.98 |
| aahhhhh | 0.46 | aaaaaaj | 0.81 |
| aaahhhh | 0.46 | aaaaahh | 0.77 |
| aaaahhh | 0.43 | aaaaaha | 0.77 |
| aaaaahh | 0.41 | aaaaaff | 0.75 |
| aaaaaff | 0.34 | aafffff | 0.74 |
| aafffff | 0.34 | aafffff | 0.74 |
| aaaafff | 0.33 | aaaafff | 0.68 |
| aaaffff | 0.32 | aaaaaad | 0.67 |
| aaaaaae | 0.26 | aahhhhh | 0.67 |
| aaaaaaj | 0.25 | aaaffff | 0.66 |

TAB. 3.5 – *Les quinze mots les plus probables dans TSD et MSD, et leur fréquence d'apparition (en % des mots de chaque base de données) pour une graine de longueur* 7 *et un alphabet à* 12 *symboles.*

positifs, donné par l'équation 25, est donc très importante. Plus précisément, 91.2% des faux positifs dans $TSD$ (respectivement 88.8% dans $MSD$) estimés en colonne 3 (resp. colonne 5) du Tableau 3.4 correspondent à des mots mono-symboliques. Ainsi, les mots mono-symboliques sont responsables de l'indexation de nombreuses séquences non similaires tout en ne représentant pas de régions particulièrement caractéristiques dans les alignements de séquences similaires. La partie (ii) du Tableau 3.4 présente les taux de faux positifs et faux négatifs induits par une indexation ne tenant pas compte des mots mono-symboliques. Comme espéré, la non indexation de ces mots augmente la spécificité d'un facteur 4 à 10, signifiant qu'un nombre important de correspondances erronées est évité. Cette nouvelle indexation a également un impact sur le taux de faux positifs, en diminuant la sensibilité d'environ 3% sur chaque base de données.

En outre, il est intéressant de remarquer que les mots les plus fréquents indiqués dans le Tableau 3.5 dans les deux bases de données correspondent à des variantes de progressions harmoniques très répandues en musique occidentale [Roc11] : par exemple, la quinte juste montante (a vers h), quinte juste descendante (a vers f) ou encore la tierce mineure descendante (a vers j) (voir [Bit87] pour une définition précise de ces termes).

| | Méthode | Base de données | MAP (%) | Durée (s/requête) |
|---|---|---|---|---|
| (i) | Alignement | $TSD$ | 56.66 | 129 |
| (ii) | | $MSD_{2k}$ | 5.71 | 388 |
| (iii) | | $\widetilde{TSD}$ | 6.15 | 273 |
| (iv) | | $MSD$ | - | 193765 |
| (v) | BLAST-$\{\mathcal{F}_1\}$ | $TSD$ | 3.70 | 0.09 |
| (vi) | | $MSD$ | - | 4.58 |
| (vii) | BLAST-$\{\mathcal{F}_2\}$ | $TSD$ | 19.23 | 0.24 |
| (viii) | | $MSD$ | - | 12.20 |
| (ix) | BLAST-$\{\mathcal{F}_2,\mathcal{F}_3\}$ | $TSD$ | 33.08 | 0.33 |
| (x) | | $MSD$ | - | 16.90 |

TAB. 3.6 – *Résultats (précision moyenne) et temps de calcul de la tâche d'identification des reprises pour les bases de données $TSD$ et $MSD$.*

### 3.3.4 Stratégie d'indexation et résultats

Les résultats statistiques présentés dans les Tableaux 3.4 et 3.5 mettent en avant une différence significative entre les deux bases de données $TSD$ et $MSD$. D'abord, les alignements de reprises partagent sensiblement moins de mots communs dans $MSD$ que dans $TSD$. Par exemple, pour une graine de taille 7, seuls 41.1% des alignements de reprises dans $MSD$ partagent des mots non mono-symboliques, contre 85.1% pour les reprises de $TSD$. L'identification de reprises dans $MSD$ est donc susceptible d'être plus complexe que dans $TSD$. On remarque également une différence du taux de faux positifs entre les bases $MSD$ et $TSD$, qui suggère que la distribution des mots dans les deux bases de données est différente et donc que l'indexation de celles-ci doit être effectuée séparément.

#### Identification par alignement

Afin de tester la pertinence du système d'indexation, les résultats indexés sur $TSD$ sont comparés aux résultats d'identification des reprises avec techniques d'alignement, tels que présentés en Section 3.2.2. Comme précédemment, chaque classe de reprises est recherchée par similarité parmi 2000 morceaux non corrélés. La valeur de précision moyenne du système sur $TSD$ est de 56.7%, comme indiqué en ligne (i) du Tableau 3.6. La distribution des scores en fonction des classes de reprises est représentée sous la forme de barres noires en Figure 3.7. Comme expliqué en Section 3.2.2, la précision de l'identification varie fortement en fonction de la classe considérée.

Une évaluation similaire est appliquée sur $MSD$ ; cependant, face à la forte complexité calculatoire des techniques d'alignement (évaluée à environ 53 heures par requête pour $MSD$, voir ci-dessous), l'ensemble du jeu de données $MSD$ ne peut être comparé en pratique. Pour permettre d'évaluer une partie de $MSD$, un sous-ensemble, noté $MSD_{2k}$, est constitué en choisissant aléatoirement 30 classes de reprises ainsi que 2000 morceaux non corrélés. De cette façon, $MSD_{2k}$ est de taille équivalente à $TSD$, et son évaluation peut être effectuée en un temps raisonnable. La valeur de précision moyenne du système d'identification de reprises obtenue

sur $MSD_{2k}$ est de 5.71% (Tableau 3.6-(ii)). Cette valeur de précision très faible suggère que le système d'évaluation de la similarité par alignement *ne permet pas* d'identifier les reprises dans $MSD$. Nous identifions deux causes qui peuvent justifier une précision si faible :

1. Les reprises de $MSD$ ($MSD_{2k}$ en particulier) présentent des variations tonales trop importantes pour être identifiées par les techniques d'alignement ;
2. Les chromas de la base $MSD$ ne constituent pas de séquences pouvant être comparées par les techniques d'alignement.

Ces alternatives sont également envisagées dans [BME11] pour expliquer une limitation de l'identification des reprises sur la base $MSD$. Si les auteurs relèvent une précision du système plus faible sur $MSD$ que sur une base de données personnelle évaluée dans [EP07], leur étude ne permet pas de discriminer laquelle des deux alternatives cause cette baisse de précision.

Les détails de l'implémentation des chromas $MSD$ ne sont pas accessibles à ce jour. Néanmoins, le principe général décrit dans [Jeh05a] nous permet de conjecturer que ce problème est dû à des différences importantes d'implémentation, telles que la résolution des chromas, le filtrage temporel ou la normalisation effectués. En outre, comme souligné par Serrà *et al.* [SGHS08], le calcul des chromas sur des trames de durées non constantes peut avoir un effet négatif sur les techniques d'alignement.

Afin de vérifier cette hypothèse par l'expérience, une solution consiste à isoler les deux alternatives en représentant les morceaux de la base $MSD$ avec l'implémentation des chromas utilisée pour $TSD$ ; cependant, la non disponibilité des données audio dans $MSD$ rend cette opération impossible. En revanche, puisque les descripteurs de $MSD$ sont calculés grâce au *framework EchoNest*[1], l'opération inverse consistant à représenter $TSD$ avec l'implémentation des chromas de $MSD$ est possible. On représente donc $TSD$ avec l'API *EchoNest* pour obtenir une nouvelle base de séquences, notée $\widetilde{TSD}$. La tâche d'identification de reprises est à nouveau évaluée pour $\widetilde{TSD}$. La valeur de précision moyenne obtenue est de 6.15%, comme indiqué en ligne (iii) du Tableau 3.6. La distribution des précisions en fonction des classes de reprises est représentée sous la forme de barres blanches en Figure 3.7. Cette baisse significative de la précision de l'identification entre $TSD$ et $\widetilde{TSD}$ pour chacune des classes appuie notre hypothèse en montrant que la cause d'échec du système sur $MSD$ est due à l'implémentation des chromas, et non à la complexité des reprises.

Le Tableau 3.7 représente pour deux exemples les 50 premiers symboles de séquences obtenues par projection des chromas pour notre implémentation et celle d'*EchoNest*. La différence forte entre les deux implémentations et la plus grande variabilité des symboles dans les séquences de la seconde implémentation renforcent la conclusion de limitation de ces dernières pour l'alignement.

### Identification par BLAST

La stratégie d'indexation introduite dans la Section 3.3.2 est appliquée sur les bases de données $TSD$ et $MSD$. Les techniques d'alignement ne permettant pas de réaliser l'identification des reprises sur $MSD$, la méthode heuristique BLAST est également non pertinente. Les résultats sur $MSD$ présentés dans la suite ne

1. http://developer.echonest.com

| *Blues Brothers - The Pink Panther* | |
|---|---|
| Séquence $TSD$ : | `aglafaagaiiiiialaiiliiiiiliaifliiaiglgaiiaiaifgag` |
| Séquence $\widetilde{TSD}$ : | `acbbchjbgbbkkakckickkakckickkaackdckkaackkgjcckad` |
| *The Beatles - Accross the Universe* | |
| Séquence $TSD$ : | `aiiiidfdaaaaaaaaaaadddkkdkciiiidaadaafaaaadafikbk` |
| Séquence $\widetilde{TSD}$ : | `aieeeeleliiblblblijlgllddd geeeedbblflibclbibigggl` |

TAB. 3.7 – *Les 50 premiers symboles de deux morceaux avec notre propre implémentation des chromas ($TSD$) et l'implémentation d'EchoNest ($\widetilde{TSD}$).*

doivent donc être considérés que comme un indicateur suggérant la performance calculatoire du système sur une base de données de l'ordre du million de morceaux.

À partir de l'étude statistique présentée dans le Tableau 3.4, on choisit la taille de graine $W = 7$ qui présente un compromis raisonnable avec un taux de faux négatifs de 14.9% pour un taux de faux positifs de 0.04% sur $TSD$. Ce choix est justifié par le faible nombre de reprises dans la base de données au regard d'un grand nombre de morceaux non pertinents ; ainsi, choisir un taux de faux positifs très faible assure de n'indexer que peu de morceaux non pertinents. De plus, comme expliqué précédemment, les classes de reprises considérées présentent une grande variabilité musicale ; ainsi, chacune d'entre elles est susceptible de comporter une certaine proportion de morceaux particulièrement complexes à détecter, même en utilisant des techniques d'alignement. C'est pourquoi le taux de faux négatifs d'environ 15% constitue une valeur raisonnable pouvant correspondre à l'élimination des reprises les moins identifiables pour chaque classe.

La base d'indexation est construite sous la forme d'une table de hachage associant à chaque mot $w$ de taille $W$ les positions où $w$ apparaît dans la base de données. Pour une requête $u$ donnée, le filtre $\mathcal{F}_1$ est d'abord appliqué en attribuant au score de similarité entre $u$ et une séquence $v$ le nombre de graines de l'espace de recherche correspondant. La valeur de précision moyenne constatée pour l'identification de reprises dans $TSD$ est alors de 3.7%, comme indiqué en ligne (v) du Tableau 3.6. Cette faible précision souligne l'insuffisance du filtre $\mathcal{F}_1$ seul pour l'indexation des reprises.

On applique à présent le filtre $\mathcal{F}_2$ afin de sélectionner les graines significatives dans l'espace de recherche entre $u$ et chaque morceau de la base de données (voir Section 3.3.2.2). Ce deuxième filtre peut être évalué en assignant comme score de similarité entre $u$ et une séquence $v$ le nombre de graines significatives conservées après application du filtre $\mathcal{F}_2$. La valeur de précision moyenne alors obtenue sur la base $TSD$ est de 19.23%, comme indiqué en ligne (vii) du Tableau 3.6. La distribution des précisions moyennes est donnée par la première barre grise (la plus à gauche) pour chaque classe de reprises sur la Figure 3.7. La précision de $\mathcal{F}_2$ semble améliorer l'identification proposée par $\mathcal{F}_1$. Elle reste cependant sensiblement inférieure à celle de la technique d'alignement pour chacune des classes.

On applique enfin le filtre plus précis $\mathcal{F}_3$ de sélection des graines identifiées par $\mathcal{F}_2$ par extension de celles-ci. Le score de similarité entre deux séquences $u$ et $v$ correspond alors à la somme des scores d'extension de graines, telle que décrite en Section 3.3.2.3. La précision moyenne obtenue est de 33.08%, comme indiqué en ligne (ix) du Tableau 3.6, soit une perte de précision de 23.58% par rapport à la

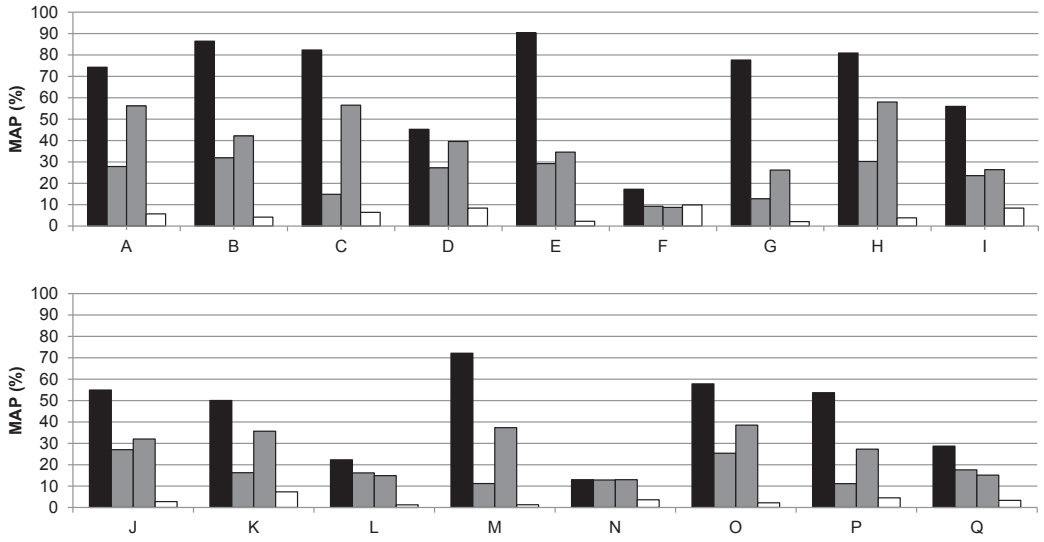


FIG. 3.7 – *Distribution des précisions moyennes (MAP) avec différentes indexations calculées pour chaque classe de reprises. Les lettres en abscisse désignent la classe considérée (voir Tableau 3.2). Noir : aucune indexation (alignement), gris : de gauche à droite, application de BLAST-*$\{\mathcal{F}_2\}$*, application de BLAST-*$\{\mathcal{F}_1,\mathcal{F}_2\}$*, blanc : alignement avec implémentation des chromas EchoNest.*

technique précise d'alignement. La répartition des précisions moyennes est donnée par la deuxième barre grise (la plus à droite) pour chaque classe de reprises sur la Figure 3.7. Ce filtre permet d'améliorer l'identification par rapport au précédent pour chaque classe de reprises. Son efficacité reste inférieure à la technique d'alignement, avec une perte de précision moyenne comprise entre 7 et 55%. On peut conclure que l'application des filtres BLAST correspond à une perte d'environ un tiers de la précision d'un système par alignement de séquences.

### Efficacité calculatoire

À cause d'un nombre élevé de mots différents à indexer sur l'ensemble de la base de données, l'implémentation de BLAST est effectuée sous la forme d'un système d'association clé/valeur en langage C. Il convient de préciser qu'aucun mécanisme de parallélisme n'a été mis en place pour ces évaluations, afin d'estimer l'efficacité calculatoire du système grâce aux seules heuristiques mises en œuvres par BLAST.

L'indexation de la base de données $MSD$ dans son intégralité nécessite environ 16 minutes de calcul sur notre configuration matérielle [1]. Les temps de calcul moyens *par requête* sont indiqués pour chaque tâche sur la dernière colonne du Tableau 3.6. Comme attendu, les opérations d'alignement nécessitent de longs calculs, requérant 129 secondes par requête sur $TSD$ et 388 sur $MSD_{2k}$. En comptant le nombre de symboles dans la base $MSD$, cette durée est extrapolée à 193765 secondes (soit environ 53 heures) par requête sur $MSD$. Notons que le retrait de la variante robuste aux transpositions locale permettrait de ne réduire cette durée que par un

1. Processeur Intel Xeon X5675 à 3.07 GHz, 12M mémoire cache, 32Go mémoire RAM

facteur 12, toujours insuffisant pour une utilisation pratique du système.

Grâce à l'indexation de BLAST, le temps de calcul est très fortement diminué, avec un temps de calcul de seulement 12.2 secondes par requête sur $MSD$ (0.24 sur $TSD$) pour le filtre de sélection $\mathcal{F}_2$ seul, et 16.9 secondes par requête sur $MSD$ (0.33 secondes sur TSD) pour les filtres $\mathcal{F}_2$ et $\mathcal{F}_3$ combinés, en moyenne. Ce temps d'exécution est donc sensiblement inférieur à celui constaté pour la technique d'alignement, et rend possible l'utilisation d'une logique de comparaison de séquences pour la recherche par similarité dans de grandes bases de données musicales.

Il convient de noter que, malgré l'échec du système à identifier les reprises avec une précision suffisante sur $MSD$, la durée d'exécution de BLAST sur cette base de données correspond tout de même au résultat d'une indexation pratique de $MSD$. BLAST étant basé sur des heuristiques permettant d'éliminer rapidement les alignements non significatifs, on ne peut garantir l'exactitude du gain en temps de calcul estimé sur $MSD$ si les séquences de cette base de données ne représentent pas une information pertinente. Néanmoins, cette estimation suggère qu'une stratégie d'indexation fondée sur les heuristiques de BLAST pourrait, à condition de disposer de séquences adaptées, être appliquée à $MSD$ pour identifier la similarité musicale en un temps de l'ordre de quelques secondes grâce à une logique d'alignement de séquences.

## 3.4 Conclusion du chapitre

Dans ce chapitre, nous avons introduit un formalisme de manipulation de séquences de symboles et décrit des techniques existantes pour la comparaison de telles séquences. Ce formalisme a ensuite été mis en pratique pour la recherche par similarité musicale. Son évaluation dans un cadre applicatif particulier, la recherche de reprises, a permis de mettre en avant la pertinence et la précision des techniques de comparaison de séquences pour le contenu musical. Face au coût calculatoire élevé de ces techniques qui empêchent leur utilisation pour le traitement de grandes bases de données musicales, une méthode d'indexation existante pour les données biologiques a été adaptée pour le contenu musical. Son évaluation avec les séquences de bases de données musicales conséquentes a permis de réduire de manière drastique le temps d'exécution en diminuant d'environ un tiers la précision du système d'identification.

Une perspective majeure des travaux présentés dans ce chapitre consiste à définir des opérations ou des heuristiques plus spécifiques à l'information tonale manipulée, afin d'identifier des similarités musicales plus complexes. Pour l'alignement, il convient d'étudier l'ajout d'opérations d'édition supplémentaires. Pour l'indexation adaptée de BLAST, des résultats préliminaires suggèrent que la définition de graines espacées [MTL02] augmente dans certains cas la sensibilité de la méthode.

Chapitre 4

# Identification de répétitions musicales

La musique est composée de nombreuses structures répétitives à plusieurs niveaux. Les répétitions peuvent être effectuées sur différents critères musicaux et à des échelles temporelles distinctes. En guise de première étape de l'étude de la répétitivité des signaux musicaux, nous proposons dans ce chapitre d'identifier plusieurs types de répétitions particulières dans un morceau quelconque représenté de manière séquentielle.

Les outils algorithmiques utiles à la comparaison de séquences musicales ont été présentés en Chapitre 3 et leur utilité pour estimer la similarité musicale entre plusieurs séquences tonales, notamment pour l'identification de reprises, a été détaillée. Dans ce chapitre, nous proposons d'utiliser les mêmes outils afin d'identifier des similarités musicales au sein même d'une séquence, et ainsi d'analyser sa structuration répétitive. Pour un morceau donné, l'objectif général de ce chapitre est ainsi de qualifier et d'extraire une répétition prépondérante au sein du morceau. Pour chaque répétition étudiée, nous proposons dans la suite un algorithme d'extraction et l'évaluons dans le cadre d'une application musicale concrète.

Ce chapitre étudie deux problèmes d'analyse de structures répétitives musicales.

Dans un premier temps, sous l'hypothèse qu'un segment est choisi dans un morceau de musique donné, nous proposons de définir et d'identifier la "meilleure" répétition de ce segment au sein du morceau. Nous étudions ce problème en Section 4.1 dans le cadre de l'application à la reconstruction musicale du signal audio. Nous détaillons une solution algorithmique et l'évaluons dans des conditions de tests subjectifs [MHT+11].

Dans un deuxième temps, sans hypothèse particulière, nous proposons de définir et d'identifier la "meilleure" répétition au sein d'un morceau de musique. Nous introduisons ainsi dans la Section 4.2 le problème de la *répétition majeure*, définie selon un critère d'optimalité. Nous proposons alors une méthode d'extraction de cette répétition et comparons nos résultats à des annotations de la structure perçue [MHRF11a], avant de les évaluer en tant que méthode d'indexation pour la recherche de reprises [MHRF11b].

## 4.1 Répétition d'un segment choisi

Notre première approche de l'étude de structures répétitives dans les œuvres musicales consiste à étudier la répétition d'un segment donné dans un morceau de musique. Ainsi, pour un morceau donné et un segment particulier de ce morceau, on cherche à identifier la "meilleure" répétition de ce segment selon un certain critère. On peut alors définir grossièrement le problème abordé ici comme l'extraction dans un morceau de musique du segment distinct le plus *similaire* au segment donné.

Cette formulation du problème n'induit aucune contrainte particulière sur le segment à identifier. En particulier, celui-ci peut chevaucher ou non le segment choisi. Dans la suite du problème, pour correspondre à une définition commune de la répétition musicale, on suppose que les deux segments de la répétition identifiée sont *disjoints*. Cependant, la prise en compte d'un chevauchement, qui peut avoir un sens musical dans certains cas, constitue une perspective de notre problème.

Afin de vérifier la pertinence musicale de la répétition analysée, nous plaçons dans la suite le problème dans le contexte applicatif de la *reconstruction audio*. Dans cette application, un ensemble de données audio manquantes, dont la position est connue, doit être reconstitué à partir du reste du morceau auquel il appartient. Cette application, présentée plus précisément dans la suite, a l'avantage de correspondre à un cas d'utilisation pratique : reconstruire un signal détérioré. En revanche, cette application requiert la définition d'un problème plus spécifique, décrit ci-dessous.

### 4.1.1 Reconstruction d'un extrait audio

#### 4.1.1.1 Présentation du problème et travaux antérieurs

La reconstruction est un problème particulièrement connu et étudié en traitement du signal audio [Ett96, GR98, LMR05, LRKO+08]. Il consiste à reconstituer un segment manquant dans un signal audio détérioré. Ce problème est principalement motivé par le souhait de corriger automatiquement les dégradations auquel peut être exposé le signal audio, pouvant être causées par de mauvaises conditions d'enregistrement, des erreurs liées à la numérisation ou des problèmes de transfert des données sonores, par exemple. Les effets indésirables incluent de courtes interférences d'amplitude (ou *clics*), des effets de larsen, des distorsions, des bruits parasites ou simplement l'absence de son [GR98, p.6–7].

Ces détériorations induisent une transformation locale d'un segment du signal audio qui ne peut généralement pas être facilement inversée. Dans cette application, la position et la durée exacte du segment altéré sont généralement connues. La section altérée est alors considérée comme *manquante*, l'objectif des algorithmes de reconstruction étant de combler cette section par un extrait perceptivement satisfaisant. Enfin, même si cette contrainte est implicitement admise dans les études existantes, on considère généralement que la temporalité du signal existant ne doit pas être modifiée par l'opération de reconstruction : la position temporelle des échantillons antérieurs et ultérieurs à la section manquante reste identique avant et après reconstruction, et ainsi le morceau reconstruit est de durée identique au morceau original.

En fonction du contexte applicatif, ce problème de reconstruction de données manquantes est traité sous différentes dénominations : interpolation, extrapolation, imputation, induction, extension ou encore occultation du signal audio [AEJ+11]. D'une manière générale, cette opération est réalisée en mettant en relation le contenu audio du morceau analysé (ou provenant d'autres œuvres) avec le voisinage du segment manquant, comme décrit plus précisément dans la section suivante.

La Figure 4.1 représente le problème général de reconstruction de données manquantes sur le domaine temporel (i) ou sur le domaine temps-fréquence (ii). Dans les deux cas, il s'agit de reconstruire une partie de l'information audio non disponible, dont la taille peut varier de quelques millisecondes à plusieurs secondes.

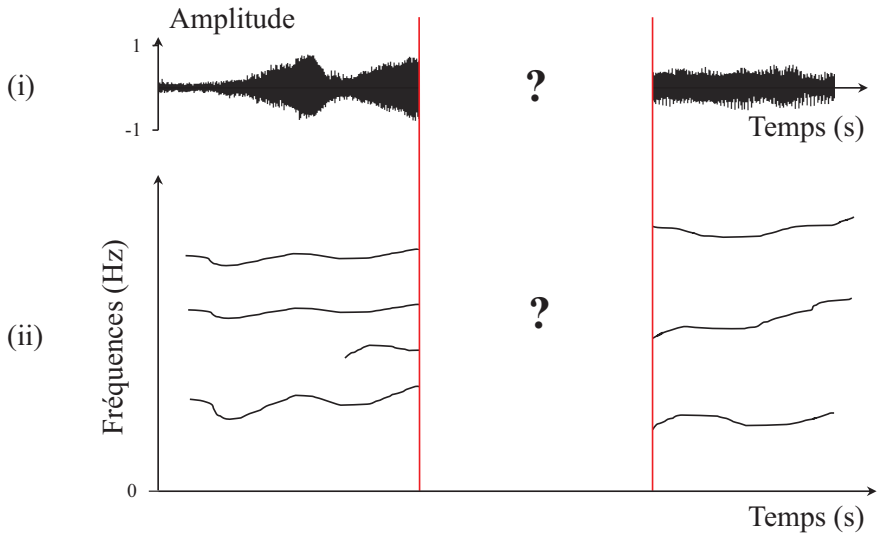


FIG. 4.1 – *Vue schématique du problème de reconstruction audio. (i) : Approche temporelle (forme d'onde). (ii) : Approche temps-fréquence (spectrogramme).*

La reconstruction audio peut d'abord être calculée par interpolation des composantes sinusoïdales du signal. Par exemple, Maher [Mah94] propose une technique basée sur l'interpolation linéaire de l'amplitude et cubique de la phase du signal. Cette technique permet d'obtenir des reconstructions fidèles au signal original pour des sections manquantes durant entre 20ms et 100ms. Lagrange *et al.* [LMR05] étendent la méthode en appliquant directement une prédiction linéaire afin d'extrapoler le contenu fréquentiel précédant la partie manquante d'une part, et celui suivant la partie manquante d'autre part. Un seuil de correspondance permet ensuite d'apparier les deux extrapolations, et une opération de fondu linéaire fréquentiel assure une transition progressive, créant ainsi une interpolation continue d'un des composants du signal. Sous l'hypothèse d'une modélisation sinusoïdale stationnaire des signaux, cette amélioration permet de reconstruire des sections manquantes de tailles plus importantes, allant jusqu'à 450ms dans le cas de signaux polyphoniques. Une technique d'interpolation similaire est proposée sur le domaine temporel par Esquef *et al.* [EVRK03], testée dans le cadre de parties manquantes d'environ 50ms.

De nombreuses approches statistiques ont été développées pour la reconstruction de signaux audio. Celles-ci incluent l'utilisation d'algorithmes de poursuite (*orthogonal matching pursuit*) [AEJ+11], le calcul d'estimateurs Bayésiens au voisinage de la section manquante [GR98] ou encore la factorisation en matrices non négatives [LRKO+08]. Bien qu'utilisant des modèles distincts, ces méthodes comblent les données manquantes en se basant sur des tendances statistiques locales à la section altérée ou globales au morceau analysé [SRS10].

Ces différentes approches sont basées sur une analyse précise des composantes du signal. Or, l'aspect non stationnaire de l'audio musical rend la tâche de reconstruction impossible pour des parties manquantes de durées élevées. La durée des sections manquantes est ainsi limitée à quelques centaines de millisecondes, voire à la seconde dans le cas d'hypothèses particulières de stationnarité [LMR05].

Quelques études de la littérature s'intéressent au problème de reconstruction

FIG. 4.2 – *Image originale (gauche) et résultat d'une reconstruction (droite) après suppression de la silhouette, d'après [CPT04].*

de plus larges sections manquantes (de plusieurs secondes) dans une optique de reconstruction du signal audio. Lu *et al.* [LWZ04b] proposent une technique de restauration fondée sur la segmentation de descripteurs timbraux par courbe de nouveauté et l'estimation de probabilités de transitions entre ceux-ci. Citons également Jehan [Jeh05a, p.103–106] qui suggère, sans l'implémenter ou l'évaluer, une approche similaire en prenant en compte des structures répétitives sur des niveaux d'abstraction plus élevés.

Les objectifs des approches de reconstruction diffèrent selon l'application considérée. D'une part, les méthodes existantes de traitement du signal audio reconstituent des sections de quelques centaines de millisecondes en minimisant une mesure entre le signal obtenu et le signal d'origine. D'autre part, la reconstruction à l'échelle de plusieurs secondes n'est pas de reconstituer un signal altéré sous sa *forme originale*, mais de produire un signal dont la section reconstruite est *perceptivement* satisfaisante, c'est-à-dire ne perturbe pas notre perception de la musique. La qualité de la reconstruction effectuée est donc recherchée sur le plan *subjectif*, et non sur une comparaison *objective* des signaux.

Si la reconstruction de grandes parties manquantes est peu étudiée en signal audio, elle est particulièrement développée en analyse et synthèse d'image, sous le nom d'*inpainting*. Motivé notamment par la restauration d'œuvres picturales ou d'anciennes photographies dégradées, l'*inpainting* a pour but de reconstituer des régions manquantes au sein d'images ne présentant pas une forme facilement reconnaissable [BSCB00]. De nombreuses méthodes dédiées à la restauration d'images se basent, comme dans le cas du signal sonore, sur des techniques d'analyse statistique des variations du signal [MM98, BSCB00, TD05]. Cependant, une autre technique répandue en synthèse de texture utilise la notion d'*auto-ressemblance* en considérant que l'image comprend un nombre élevé de répétitions locales. Cette technique peut être vue comme un procédé de reconstitution par l'exemple [CPT04, EL99, Ash01]. L'avantage majeur de cette approche est la taille importante des motifs manquants qui peuvent être reconstruits [BSCB00]. La Figure 4.2 montre, à partir d'une image originale (à gauche), un exemple de reconstruction de l'image (à droite) après retrait de la silhouette et reconstruction par l'exemple des pixels manquants [CPT04].

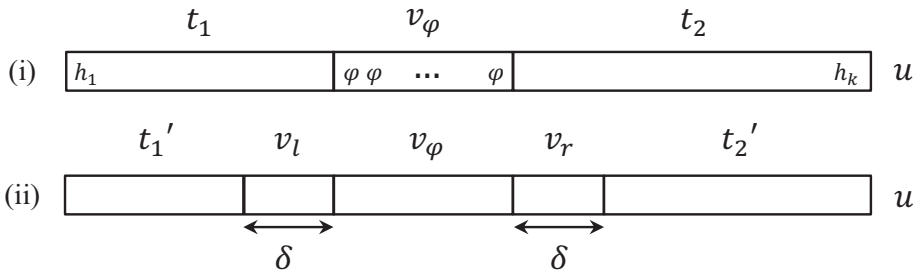


FIG. 4.3 – *Modélisation du problème de reconstruction audio. (i) : Séquence de symboles $t_1 v_\varphi t_2$. (ii) : Introduction des contextes locaux.*

Dans cette section, on se pose le problème de reconstruction de grandes parties manquantes dans le signal audio, de l'ordre de plusieurs secondes. Notre approche du problème de reconstruction est fondée sur une description du signal audio sur un niveau *musical*. La reconstruction de données audio avec une telle approche est aussi connue sous le nom de *restauration de textures audio* [LWZ04b]. Comme expliqué par Jehan [Jeh05a, p.103–106], utiliser une représentation musicale du signal audio permet d'identifier des répétitivités structurelles de textures musicales, et permet de considérer la restauration de sections beaucoup plus longues qu'avec les approches de traitement du signal. En résumé, l'étude présentée dans cette section se concentre sur la reconstitution perceptivement satisfaisante de grandes parties manquantes. Notre approche est inspirée du procédé de reconstruction par l'exemple développé dans le cadre de l'analyse d'image.

La méthode détaillée dans la suite de cette section a été publiée dans les actes de la conférence *International Society for Music Information Retrieval* [MHT+11].

#### 4.1.1.2 Représentation

Soit $x$ un signal audionumérique comprenant $N$ échantillons. On suppose que $x$ contient une section de signal manquante, c'est-à-dire qu'il existe deux indices d'échantillons $n_i$ et $n_o$ tels que $n_i < n_o \leq N$ et $\forall i \in [\![n_i, n_o]\!], x[i] = 0$.

L'étape de représentation décrite Sections 2.3.1 et 2.2.5 fournit pour le signal connu une séquence de vecteurs $h_1 h_2 \dots h_k$ qui décrivent l'évolution d'un critère musical, ici l'harmonie, chaque vecteur étant défini sur l'alphabet des chromas $\Sigma_{\text{chr}}$. La section manquante est représentée par un symbole spécial $\varphi$, qui indique l'absence d'information audio. Ce symbole est attribué à toute trame de $x$ qui contient au moins un échantillon audio manquant. On représente ainsi le signal $x$ par la séquence de symboles :

$$u = t_1 v_\varphi t_2 \text{ définie sur l'alphabet } \Sigma = \Sigma_{\text{chr}} \cup \{\varphi\}, \tag{26}$$

où $v_\varphi = \varphi \dots \varphi$ représente la section manquante de $x$ et les séquences de chromas $t_1$ et $t_2$, définies sur $\Sigma_{\text{chr}}$, représentent les sections de signal audio bien définies (Figure 4.3-(i)).

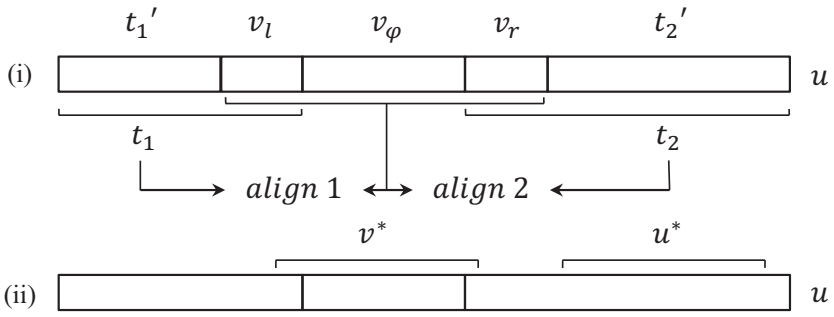


FIG. 4.4 – *Solution algorithmique au problème de reconstruction. (i) : Les deux alignements locaux calculés. (ii) : Facteurs alignés résultants.*

D'une manière générale, les méthodes de reconstitution par l'exemple utilisent le caractère local répétitif du signal pour identifier les sections à reproduire [CPT04]. Dans le cas du signal audio, on peut ainsi considérer le *contexte tonal local* d'un instant $t$, qui peut être vu comme la description de l'harmonie présente juste avant (*contexte gauche*) et juste après (*contexte droit*) l'instant $t$. Formellement, on introduit un seuil $\delta$ correspondant à la taille de chaque contexte, en nombre de symboles. Le *contexte gauche* $v_l$ est défini comme le suffixe de $t_1$ de longueur $\delta$, et le *contexte droit* $v_r$ est défini comme le préfixe de $t_2$ de longueur $\delta$. La Figure 4.3-(ii) illustre la représentation de $u$ avec les contextes tonaux locaux de $v_\varphi$.

La séquence $v = v_l v_\varphi v_r$ contient à la fois l'information manquante et ses contextes tonaux locaux. C'est de cette séquence que l'on souhaite trouver une répétition musicale approchée afin de combler la section manquante.

#### 4.1.1.3 Problème et algorithme

**Problème 3 (Reconstruction par l'exemple)** *Soient $\Sigma$ un alphabet, $\varphi$ un symbole absent de $\Sigma$ et $u = t_1' v_l v_\varphi v_r t_2'$ une séquence définie sur $\Sigma \cup \{\varphi\}$ vérifiant :*

$$\begin{cases} t_1', v_l, v_r, t_2' \in \Sigma^* \\ v_\varphi = \varphi\varphi\ldots\varphi \\ |v_l| = |v_r| = \delta \end{cases}.$$

*Le problème de reconstruction par l'exemple consiste à identifier le facteur $v$ de $u$ tel que :*

$$\begin{cases} v \in \Sigma^* & (i) \\ s^*(v, v_l v_\varphi v_r) = \max\limits_{w \in \mathcal{S}(u), w \cap_u v_\varphi = \varepsilon} \{s^*(w, v_l v_\varphi v_r)\} & (ii) \end{cases}.$$

Le problème de reconstruction est résolu par le calcul de la similarité locale entre $v$ et $t_1$ d'une part et la similarité entre $v$ et $t_2$ d'autre part. La Figure 4.4-(i) représente les deux alignements locaux effectués : *align* 1, représentant l'alignement de $t_1$ et de $v_l v_\varphi v_r$ et *align* 2, représentant l'alignement de $t_2$ et de $v_l v_\varphi v_r$.

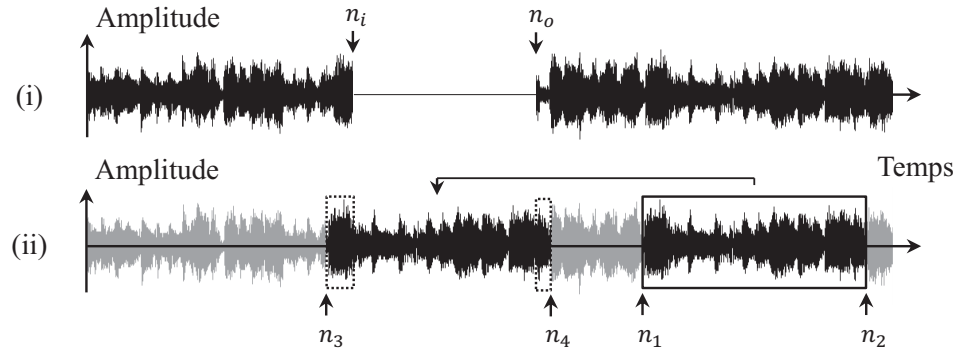


FIG. 4.5 – *Reconstruction du signal audio. (i) : Signal d'origine avec données manquantes (silence). (ii) : Forme d'onde reconstruite. Les sections noires indiquent la recomposition effectuée. Les régions entourées de pointillés correspondent à la reconstruction par chevauchement audio.*

Formellement, on calcule la répétition de score de similarité locale optimal $s^r$ avec $v$ par la proposition :

**Proposition 7 (Calcul de reconstruction)** *Avec les notations du Problème 3, on calcule le score optimal $s^r$ de la manière suivante :*

$$s^r = \max\{s^*(t_1, v_l v_\varphi v_r), s^*(t_2, v_l v_\varphi v_r)\}.$$

*On note $u^*$ le facteur de $t_1$ ou $t_2$ effectivement aligné avec le facteur noté $v^*$ de $v_l v_\varphi v_r$ lors du calcul d'alignement local optimal de score $s^r$. $u^*$ est alors solution du problème de reconstruction.*

**Preuve 1** *Le score $s^*(t_1, v_l v_\varphi v_r)$ correspond à la similarité maximale entre $v_l v_\varphi v_r$ et $t_1$, le préfixe de $u$ avant $v_\varphi$, tandis que le score $s^*(t_2, v_l v_\varphi v_r)$ correspond à la similarité maximale entre $v_l v_\varphi v_r$ et $t_2$, le suffixe de $u$ après $v_\varphi$. $s^r$ correspond donc au score de similarité maximale entre $v_l v_\varphi v_r$ et un facteur de $u$ disjoint de $v_\varphi$, d'où la propriété (ii). On en déduit également que $u^*$ est disjoint de $v_\varphi$, ce qui assure que $u^*$ est défini sur $\Sigma$, d'où la propriété (i).*

Un exemple de répétition identifiée est représenté en Figure 4.4-(ii) : le facteur $u^*$ de $t_2$ a un score de similarité avec la séquence $v^*$ optimal, et constitue une solution au problème de reconstruction.

#### 4.1.1.4 Application à l'audio

Afin de satisfaire une contrainte pratique de reconstruction, il convient de favoriser l'alignement de l'intégralité de $v_\varphi$, c'est-à-dire d'assurer, si possible, que $v_\varphi$ est bien un facteur de $v^*$. Pour ce faire, on définit un schéma de scores de pondération particulier. Pour deux chromas $h_1$ et $h_2$ de $\Sigma_{\text{chr}}$, le schéma de scores de pondération

est le suivant :

| | |
|---|---|
| Insertion, suppression : | $\begin{cases} \lambda(h_1,\phi) = \lambda(\phi,h_2) = -0.7 \\ \lambda(\varphi,\phi) = \lambda(\phi,\varphi) = 0 \end{cases}$ |
| Substitution : | $\begin{cases} \lambda(h_1,h_2) = \lambda_{\text{chr}}(h_1,h_2) \\ \lambda(h_1,\varphi) = \lambda(\varphi,h_2) = 0.1 \\ \lambda(\varphi,\varphi) = -\infty \end{cases}$ |
| Correspondance sur $\Sigma_{\text{chr}}$ : | $\lambda_+ = 1$ |
| Remplacement sur $\Sigma_{\text{chr}}$ : | $\lambda_- = -0.9$ |

(27)

La définition d'un score positif pour la substitution d'un symbole avec $\varphi$ impose que la substitution d'un symbole de $\Sigma$ avec $\varphi$ soit toujours choisie dans le calcul de similarité. De plus, la définition d'un score infiniment faible pour la correspondance de symboles $\varphi$ assure que les données manquantes ne sont jamais alignées entre elles.

Une fois cet algorithme appliqué sur les séquences de chromas, la dernière étape de la technique de reconstruction consiste à utiliser la répétition identifiée pour reconstituer la section manquante, comme illustré par la Figure 4.5. Notons $x[n_1 \dots n_2]$ les échantillons intervenant dans le calcul des symboles de la séquence $u^*$, et $x[n_3 \dots n_4]$ les échantillons (éventuellement nuls) intervenant dans le calcul des symboles de la séquence $v^*$. La section manquante dans $x$ est comblée en insérant à la place des échantillons de $x[n_1 \dots n_2]$ ceux de $x[n_3 \dots n_4]$. Afin d'assurer des transitions progressives entre les composantes sinusoïdales, les sections chevauchantes entre $v^*$ et $t_1$ d'une part et $v^*$ et $t_2$ d'autre part sont modifiées selon une technique simple d'*overlap-add* [Cro80], qui effectue un fondu croisé linéaire entre les deux sections. La Figure 4.5-(ii) représente l'opération de reconstruction des échantillons audio. Les zones entourées de pointillées mettent en valeur les sections de chevauchement par *overlap-add*.

Formellement, le signal reconstruit $\widetilde{x}$ est donné, pour $n \in [\![1,N]\!]$, par la formule :

$$\widetilde{x}[n] = \begin{cases} x[n] & \text{si } n < n_3 \\ \frac{n_i - n}{n_i - n_3} x[n] + \frac{n - n_3}{n_i - n_3} x[n_1 + n - n_3] & \text{si } n_3 < n < n_i \\ x[n_1 + n - n_3] & \text{si } n_i < n < n_o \\ \frac{n - n_o}{n_4 - n_o} x[n] + \frac{n_4 - n}{n_4 - n_o} x[n_1 + n - n_3] & \text{si } n_o < n < n_4 \\ x[n] & \text{si } n > n_4 \end{cases} . \tag{28}$$

Il convient de noter que l'on suppose dans cette formule que $n_4 - n_3 = n_2 - n_1$, c'est-à-dire que $|u^*| = |v^*|$. Cependant, l'alignement local n'imposant pas l'égalité des tailles de séquences résultantes, il est possible que cette assertion soit fausse. Ce cas est susceptible d'arriver, par exemple, lorsque $x[n_1 \dots n_2]$ est une répétition de $x[n_3 \dots n_4]$ jouée à un tempo différent. Dans ce cas, le facteur aligné de plus grande taille est tronqué afin de conserver l'égalité des longueurs. La prise en compte de séquences de différentes longueurs, par exemple en utilisant une technique de *time stretching* [MC90, Fer99] à la reconstruction, est laissée en perspective de ce travail.

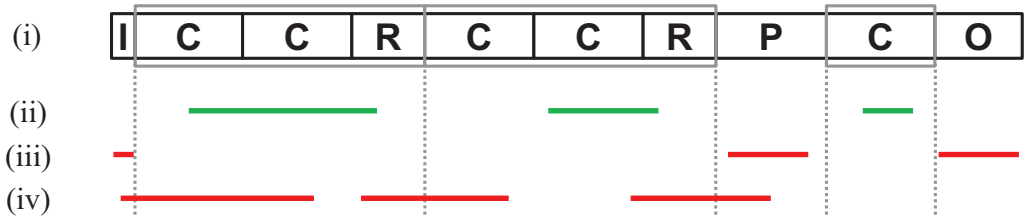


FIG. 4.6 – *Illustration de la contrainte de répétitivité. Les motifs répétés dans (i) sont encadrés en gris. Les sections vertes représentées en (ii) satisfont la contrainte de répétitivité, alors que les sections rouges en (iii) et (iv) ne satisfont pas cette contrainte.*

#### 4.1.1.5 Protocole d'évaluation

L'algorithme décrit dans les sections précédentes est appliqué à une description tonale du signal audio (voir Chapitre 2) pour déterminer la reconstitution optimale. Des variations du timbre, du rythme ou des paroles, par exemple, peuvent intervenir entre des sections identifiées comme répétées. Un signal d'origine et sa reconstruction par l'algorithme peuvent donc différer fortement sur leurs composantes tout en restant perceptivement proches. Comme expliqué en introduction du problème, l'objectif de notre étude de la reconstruction est de reconstituer une section *perceptivement satisfaisante*, c'est-à-dire qui ne gêne pas la perception musicale.

**Occultation**

Pour tester l'algorithme de reconstruction, il convient tout d'abord de constituer un corpus de morceaux possédant des sections manquantes. Pour ce faire, on propose d'effacer artificiellement, ou d'*occulter* des données audionumériques dans une base de données. Ainsi, la génération de données manquantes est effectuée de manière aléatoire afin de constituer un corpus de test aussi représentatif que possible de l'ensemble des altérations susceptibles d'apparaître dans les enregistrements audio. Cependant, puisque notre algorithme se base sur un procédé de reconstitution par l'exemple, les données manquantes peuvent être reconstruites si elles font partie d'une répétition approchée au sein d'un morceau. Afin d'évaluer rigoureusement la méthode dans son champ d'application, on ne souhaite donc pas occulter les sections non répétées. Plus précisément, on introduit une *contrainte de répétitivité* par l'assertion suivante :

*Une section ne peut être occultée que si elle appartient à un segment audio répété, selon la vérité structurelle correspondante.*

La Figure 4.6 illustre cette contrainte sur un exemple de structure de musique populaire occidentale. La séquence de symboles ICCRCCRPCO représentée en (i) indique la forme structurelle que suit ce morceau au cours du temps, telle qu'annotée par un expert. Les symboles correspondent aux motifs classiquement identifiables en musique populaire occidentale : l'introduction I, le couplet C, le refrain R, le pont P et la conclusion O. Dans cette séquence de symboles, les motifs compris dans un cadre gris C, R, CC, CR et CCR sont répétés. Tout segment compris dans

l'un de ces motifs, à l'image de ceux représentés en vert Fig. 4.6-(ii), satisfait donc la contrainte de répétitivité. En revanche, les segments inclus dans une section non répétée, comme les sections matérialisées en rouge en Fig. 4.6-(iii), ne satisfont pas cette contrainte. De même, tout segment chevauchant une frontière non répétée, tel que ceux représentés en Fig. 4.6-(iv), ne satisfait pas la contrainte.

Cette contrainte induit une restriction sur l'étendue de l'application de la méthode ; celle-ci est évaluée précisément sur la base de tests ci-dessous. Dans le cas où la contrainte n'est pas respectée, le résultat de la reconstruction de sections non répétées est susceptible de fortement varier en fonction des signaux musicaux manipulés et de leur structure. Ainsi, dans l'exemple de la Figure 4.6, on peut raisonnablement supposer que les motifs définis comme uniques peuvent difficilement être reconstruits d'une manière perceptivement satisfaisante dans le cas général. Néanmoins, il est envisageable que l'occultation du motif P et son remplacement par un motif C, par exemple, produise un résultat acceptable car conservant une cohérence musicale. La contrainte de répétitivité semble donc nécessaire si l'on souhaite conserver une indépendance entre les résultats et la nature des morceaux reconstruits.

Pour un morceau donné, le processus d'occultation est effectué selon le protocole suivant :

1. Choix aléatoire d'une durée d'occultation $l$ comprise entre des bornes fixées $l_{min}$ et $l_{max}$ ;
2. Sélection aléatoire d'une section répétée (d'après la vérité structurelle) de durée $L > l$ ;
3. Choix aléatoire d'une date de début $t$ dans cette section telle que $t \leq L - l$ ;
4. Occultation du son : mise à zéro des échantillons compris entre les dates $t$ et $t + l$.

### Base de données

La contrainte de répétitivité nécessite l'utilisation d'une base d'annotation des répétitions musicales. Cependant, aucune base de données annotant rigoureusement la répétition au sens de notre étude n'est disponible. On choisit alors d'utiliser des annotations de la structure répétitive perçue telles qu'annotées par des experts. Pour chaque morceau de musique, ces bases de données proposent un découpage répétitif binaire en segments caractérisés par des étiquettes. La récurrence d'une étiquette indique une répétition musicale, tandis que deux étiquettes différentes soulignent des segments dissimilaires. La méthode d'annotation ainsi que la précision de telles bases annotées sont discutées plus en détail en Section 5.3.4.2.

La base de données considérée, notée $STRUCT$, correspond un ensemble de données de différents laboratoires (voir Section 5.3.4.2) et est notamment distribuée par le projet OMRAS2[1] [MCD+09]. Les données utilisées dans nos expériences se composent de 252 annotations de la structuration des morceaux de musique de trois artistes différents : un ensemble $STRUCT_B$ de 180 morceaux du groupe *The Beatles*, un ensemble $STRUCT_Q$ de 34 morceaux du groupe *Queen* et un ensemble $STRUCT_J$ de 38 morceaux du chanteur *Michael Jackson*.

1. http://isophonics.net/content/reference-annotations

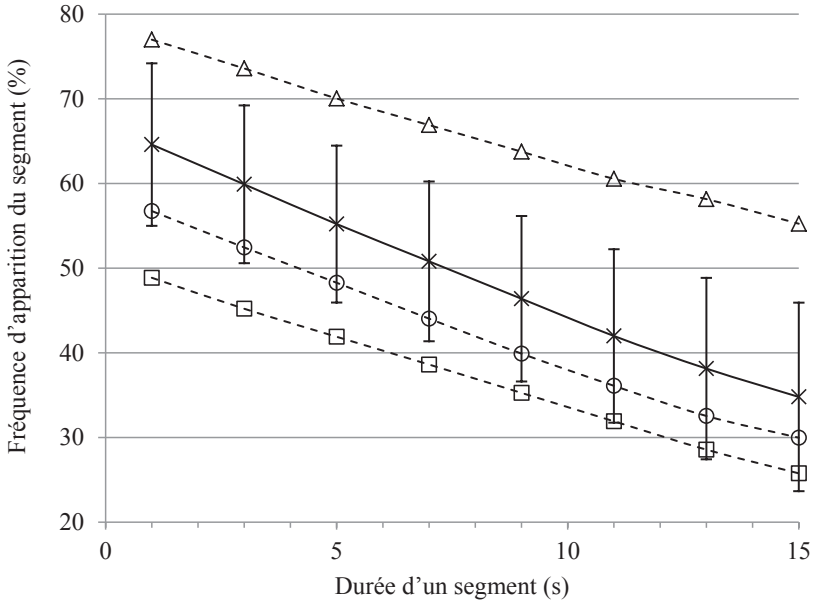


FIG. 4.7 – *Restriction induite par la contrainte de répétitivité sur les différentes bases annotées. Le graphique représente la fréquence d'apparition d'un segment répété choisi aléatoirement dans l'ensemble de la base de données en fonction de la taille du segment. Carrés : sous-base $\mathcal{D}_Q$, cercles : sous-base $\mathcal{D}_J$, triangles : sous-base $\mathcal{D}_B$, croix : moyenne sur $\mathcal{D}$.*

L'application de la contrainte de répétitivité sur ces annotations est alors effectuée selon l'algorithme suivant :

1. Retirer les segments dont les étiquettes ne sont pas répétées ;
2. Retirer les frontières répétées, c'est-à-dire fusionner deux segments consécutifs qui se répètent de manière consécutive.

À l'issue de cet algorithme, les segments restants correspondent aux sections pouvant être occultées. Les sections manquantes sont générées en appliquant sur chaque morceau correspondant à une annotation de $STRUCT$ le processus d'occultation tel que défini à la section précédente. La base de morceaux partiellement occultés est notée $\mathcal{D}$, les sous-bases regroupées par artistes étant désignées par $\mathcal{D}_B$, $\mathcal{D}_Q$ et $\mathcal{D}_J$ selon la nomenclature précédente. À l'issue de cette étape, 252 morceaux partiellement occultés sont donc présents dans la base de tests $\mathcal{D}$ ; la section effectivement occultée dans chaque morceau alors d'une durée comprise entre 1 et 16 secondes, avec une moyenne constatée de 8.2 secondes sur $\mathcal{D}$.

### Mesure de la répétitivité

Afin d'évaluer la restriction induite sur la base de tests par la contrainte de répétitivité, on peut calculer le nombre de sections dans les morceaux de la base de tests qui satisfont cette contrainte. La Figure 4.7 représente la fréquence d'apparition d'une section répétée, choisie aléatoirement, d'après la vérité structurelle, en fonction de la taille de la section. La courbe en traits pleins indique la moyenne sur tous les morceaux de la base $\mathcal{D}$ (les lignes verticales représentant l'écart-type de la mesure), tandis que les courbes pointillées détaillent les résultats par artiste :

| Score | Qualité | Imperfections |
|---|---|---|
| **5** | Excellente | Imperceptibles |
| **4** | Bonne | À peine perceptibles |
| **3** | Passable | Acceptables |
| **2** | Médiocre | Dérangeantes |
| **1** | Mauvaise | Très dérangeantes |

TAB. 4.1 – *Échelle subjective définie pour établir la note moyenne d'opinion.*

points en carrés pour $\mathcal{D}_Q$, en cercle pour $\mathcal{D}_J$ et en triangle pour $\mathcal{D}_B$. Par exemple, si l'on choisit aléatoirement une section de 5 secondes dans un morceau de la base de données, alors la section est répétée, en moyenne, dans 56% des cas sur $\mathcal{D}$, dans 42% des cas sur $\mathcal{D}_Q$, dans 48% des cas sur $\mathcal{D}_J$ et dans 70% des cas sur $\mathcal{D}_B$. La répétitivité semble fortement varier en fonction des artistes dans la base de données ; ainsi, la fréquence de sections répétées dans un morceau de *The Beatles* est entre 8.7% et 16.2% plus élevée que dans un morceau de *Queen*. Cette différence, qui peut être due aux choix d'agencements structurels différents selon les compositeurs, est également liée à un manque d'homogénéité des annotations structurelles, comme souligné notamment par Peeters et Deruty [PD09] et discuté plus précisément en Section 5.3.4.2.

En respectant la contrainte de répétitivité, l'occultation est donc effectuée sur une partie des segments de la base de données. Ainsi, 64.6% des morceaux en moyenne peuvent être occultés par des sections d'une seconde, 50.8% par des sections de 7s et jusqu'à 34.7% pour des sections de 15s.

#### Génération des données

Chaque morceau de $\mathcal{D}$ est partiellement occulté selon le processus décrit précédemment. L'algorithme de reconstruction est alors appliqué sur chaque morceau. Le paramètre $\delta$ de durée des contextes tonaux locaux doit être fixée à une valeur suffisamment élevée afin de capter une progression harmonique représentative, mais suffisamment faible pour que le contexte conserve un caractère local à la section manquante. Des tests préliminaires sur un sous-ensemble de morceaux de $\mathcal{D}$ suggèrent la valeur $\delta = 4$s comme compromis raisonnable pour reconstituer les sections occultées. Ce réglage, obtenu de manière empirique, dépend fortement de la base de données et de la durée des structures répétitives qui la composent.

À l'issue de cette étape, on dispose ainsi de 252 morceaux partiellement occultés, de manière aléatoire, puis reconstruits par notre algorithme.

### 4.1.1.6 Tests perceptifs

Notre protocole d'évaluation subjective est inspiré du test de *Note moyenne d'opinion* (ou *Mean Opinion Score*, MOS), telle que définie par l'Union Internationale des Télécommunications [1] [IT94]. Cette technique correspond à un standard particulièrement utilisé pour évaluer la qualité des algorithmes de codage, notamment dans le domaine de la parole pour la téléphonie, la transmission réseau ou

1. http://www.itu.int/fr

la synthèse vocale (voir par exemple [VV05] et les références incluses). Cette technique de test simple est particulièrement adaptée à l'évaluation perceptive du signal audio.

### Méthode d'évaluation

Conformément au test MOS, on commence par définir une échelle d'évaluation comprenant 5 niveaux de satisfaction, le niveau 5 représentant la meilleure qualité et le niveau 1 la plus mauvaise. Le Tableau 4.1 détaille les différents niveaux.

La spécification du test [IT94] précise en outre qu'il est important que l'auditeur soit exposé à de courts extraits audio sur une période relativement courte, de l'ordre de la dizaine de minutes. Afin de réduire la durée d'écoute de chaque extrait, on effectue une opération additionnelle sur les données de test consistant à retirer une grande partie de l'information audio inchangée. Formellement, on considère un morceau de $\mathcal{D}$ occulté sur une durée $d$ à partir de la date $t$. En tirant aléatoirement deux durées $t_1$ et $t_2$ comprises entre 5 et 10 secondes, on réduit l'extrait à écouter à l'intervalle $[t-t_1,t+d+t_2]$. De cette façon, la durée de l'extrait résultant correspond à la durée de la reconstruction plus 10 à 20 secondes de données non reconstruites. Cette technique permet ainsi de concentrer l'écoute sur la section reconstituée, tout en introduisant un minutage aléatoire qui ne permet pas de connaître *a priori* les bornes de cette reconstruction. Notons que si cette technique permet de rendre l'écoute des extraits moins fastidieuse, elle tend à concentrer l'attention des sujets autour de la reconstruction et rend plus exigeante l'évaluation de sa qualité.

De cette manière, 252 extraits sont ainsi générés, chacun durant entre 10 et 30 secondes pour une durée moyenne de 21.8s.

Pour réduire la durée de test à une dizaine de minutes d'écoute, chaque sujet est soumis à l'évaluation de 26 extraits. Parmi ceux-ci, 21 sont tirés aléatoirement parmi les 252 extraits reconstruits, et 5 morceaux, proposés à tous les testeurs, correspondent à des extraits originaux non reconstruits (ou *extraits-témoins*). La présence de ces derniers a pour objectif d'éliminer un éventuel *effet-individu* en détectant par exemple les sujets répondant de manière aléatoire selon la règle suivante : si un sujet juge la qualité de 3 extraits non altérés ou plus comme perfectible (note différente de 5), alors son jugement est qualifié de non pertinent, et est écarté de l'évaluation.

Il est également important, d'après le test MOS, de prêter attention à l'expertise du sujet pour l'évaluation d'extraits audio. En s'inspirant de ce principe, on propose de distinguer les sujets du test en deux catégories : les musiciens et les non musiciens, le sujet déclarant à quelle catégorie il appartient.

### Résultats

Les tests d'écoute ont été menés sur 80 auditeurs distincts, dont 34 se déclarant comme musiciens et 46 comme non musiciens. Chaque extrait audio a été écouté en moyenne 7.1 fois, avec un minimum d'une écoute et un maximum de 15 écoutes par extrait. Les écoutes des 5 extraits-témoins ont donné lieu à 400 observations dont 10 ont été évaluées de manière inexacte (imperfections perçues). Cependant, ces 10 notes invalides ont été attribuées par 10 utilisateurs distincts. Par conséquent, tous les jugements des 80 testeurs peuvent être considérés comme pertinents. Les

| Base de données | Musiciens | Non musiciens | Total |
|---|---|---|---|
| $\mathcal{D}_B$ (*The Beatles*) | 3.95 | 4.13 | 4.05 |
| $\mathcal{D}_J$ (*Michael Jackson*) | 4.21 | 4.26 | 4.24 |
| $\mathcal{D}_Q$ (*Queen*) | 3.40 | 3.94 | 3.71 |
| $\mathcal{D}$ (Tous morceaux) | 3.92 | 4.13 | 4.04 |

TAB. 4.2 – *Résultats d'évaluation subjective de la reconstruction, regroupés par base de données et par classe d'évaluateurs. Les valeurs sont données sur une échelle subjective telle que présentée dans le Tableau 4.1.*

statistiques présentées dans la suite ne comptabilisent pas ces 400 observations de morceaux-témoins, afin de ne pas introduire de biais dans l'évaluation des reconstructions effectuées.

Les résultats globaux de la reconstruction sont résumés dans le Tableau 4.2. La moyenne des notes attribuées aux morceaux de chacune des bases de données est présentée, pour chaque classe de sujets. La note moyenne globale des extraits évalués est de 4.04, jugeant ainsi la qualité de la reconstruction comme bonne avec des imperfections *à peine perceptibles* en moyenne. La perception des sujets musiciens semble légèrement plus exigeante, avec une note moyenne inférieure de 0.21 points par rapport aux sujets non musiciens.

On remarque également que les résultats semblent varier selon les artistes. Ainsi, les reconstructions semblent moins satisfaisantes sur l'ensemble d'extraits de *Queen* que sur les extraits de *Michael Jackson*, avec un écart de 0.53 points sur la note moyenne. La raison de cette variabilité en fonction des artistes n'est pas triviale et peut être expliquée par de nombreux facteurs. D'abord, les compositeurs sont susceptibles d'employer différemment la répétition de motifs structurels, en amenant plus ou moins de variations musicales entre deux couplets, sur la mélodie ou les instruments par exemple. Ces différences impactent la reconstruction, qui est d'autant plus perceptible que la répétition détectée comporte de variations. En outre, rappelons que les sections occultées puis reconstruites sont choisies en fonction d'une annotation par différents experts, dont les normes et règles de notation sont réputées peu homogènes selon les bases de données, donc selon les artistes (voir Section 5.3.4.2). Enfin, les habitudes d'écoute des sujets de test ont un impact direct sur leur exigence face aux reconstructions. Ainsi, un auditeur connaissant parfaitement bien un morceau est susceptible d'être plus critique à l'écoute d'une reconstitution de celui-ci. Dans un souci de simplification de la tâche de test et de réduction de sa durée, la pré-connaissance des extraits n'a pas fait partie du protocole des tests menés. Néanmoins, il est important que les futurs travaux sur la reconstruction renseignent cette information pour chaque observation effectuée afin d'évaluer l'importance de ce critère.

Les notes moyennes attribuées par les sujets de test suggèrent une bonne qualité de la reconstruction. La figure 4.8 décrit plus précisément la distribution des notes attribuées pour chaque observation, tous morceaux confondus, en détaillant chaque classe d'évaluateurs : tous les sujets (noir), sujets musiciens (gris foncé) et sujets non musiciens (gris clair). Cette figure présente la tendance de notation générale à l'écoute des reconstructions. Ainsi, pour plus de la moitié des observations (55.4%), tous morceaux et tous utilisateurs confondus, la note 5 est attribuée. Cela signifie

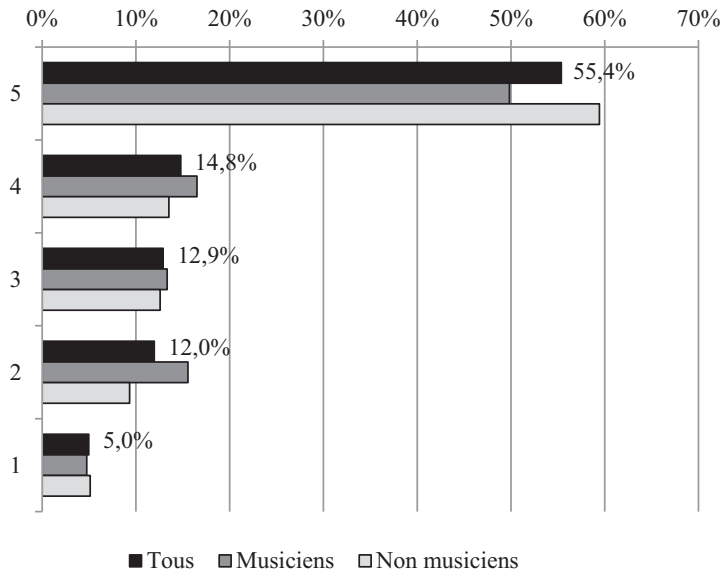


FIG. 4.8 – *Répartition des notes attribuées pour chaque classe de sujets*

que plus de la moitié des observations évaluent comme excellente (non perceptible) la reconstruction effectuée. La note 5 est moins fréquente pour les sujets musiciens, dont 49.8% des observations se voient attribuer cette note.

En regroupant les scores par extrait et en calculant leur note moyenne, on peut mesurer la qualité de reconstruction perçue par extrait. La Figure 4.9 présente la répartition des notes moyennes par extrait, en détaillant chaque classe de sujets. On rappelle que les observations effectuées par chaque sujet sont tirées aléatoirement parmi l'ensemble des reconstructions ; de ce fait, certains extraits ont été notés uniquement par des non musiciens, certains uniquement par des musiciens et certains par les deux types de sujets. Le score moyen d'évaluation par tous les sujets (en noir) ne correspond donc pas forcément à la moyenne des scores moyens des deux classes (en gris). Cette figure souligne donc que 31.2% des reconstitutions sont notées 5 par tous les sujets les ayant évaluées. La reconstruction est donc qualifiée d'excellente pour un tiers de la base de données. En outre, pour 33% des extraits, la reconstruction semble à peine perceptible avec un score compris entre 4 et 5. Le jugement des sujets musiciens semble à nouveau plus critique avec une notation relative aux non musiciens plus faible de 6.37% des extraits notés 5, plus forte de 3.8% des extraits notés entre 3 et 4 et plus forte de 6.9% des extraits notés entre 2 et 3. Les scores les plus faibles semblent rarement attribués dans les deux cas, avec seulement 4.5% des extraits et 5% de toutes les observations.

Les imperfections détectées dans les extraits reconstruits sont liées à différents critères musicaux comme le rythme, la voix, la dynamique ou encore le timbre perçu. La liste suivante détaille les imperfections les plus rapportées par les sujets de test, et propose une explication à leur apparition :

- Légers décalages rythmiques. De faibles changements dans le rythme sont fréquemment rapportés, particulièrement dans le cas d'une accélération du tempo perçu (voir par exemple [Wan84] pour une explication plus complète). Ceux-ci sont provoqués par la présence de sections répétées jouées à un tempo légère-

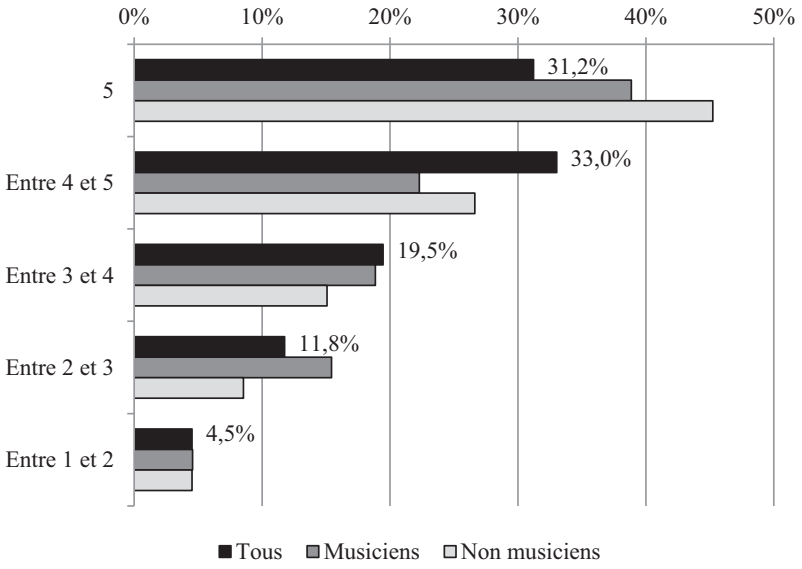


FIG. 4.9 – *Répartition des moyennes par morceau et pour chaque classe de sujets*

ment différent à chaque occurrence, et dont l'insertion apporte des modifications soudaines du tempo dans la section reconstruite. Cette imperfection est particulièrement rapportée pour la reconstruction d'extraits provenant de morceaux réputés pour n'avoir pas été enregistrés en suivant une pulsation régulière (métronome).

- Modifications timbrales subites. Le changement rapide (quelques secondes) d'instruments est également souvent perçu comme anormal au cours de transitions reconstruites. Cet effet est dû à l'utilisation de répétitions comportant une harmonie similaire mais une instrumentation différente.
- Incohérences des paroles chantées. Cette imperfection est régulièrement rapportée dans les cas où l'occultation supprime en partie une phrase chantée, puis où la reconstruction assigne une section similaire d'un point de vue tonal, mais dont les paroles ont été modifiées. Il convient de noter que sur les 80 sujets soumis à l'évaluation, 78 ont pour langue maternelle le français, alors que les morceaux proposés sont tous en langue anglaise. On peut donc supposer que le nombre d'incohérences sur les paroles détectées serait supérieur dans le cas de sujets anglophones.

**Bilan expérimental**

Cette évaluation met en avant le succès de la méthode à reconstruire de manière parfaitement satisfaisante 31.2% des extraits audio, et de manière à peine perceptible 33% d'entre eux. De plus, bien que l'expertise musicale rende l'évaluation plus exigeante, le fort taux d'attribution de la note maximale (plus d'une observation sur deux) suggère que la méthode est très efficace pour une majorité des signaux pour le panel de testeurs, résultat appuyé par la note moyenne de 4.04 sur 5 pour l'ensemble des extraits reconstruits.

Ce résultat est à nuancer par le caractère subjectif de notre évaluation, dont la fiabilité est nécessairement liée à l'échantillon de sujets (ici 80 personnes) et aux

bases de données évaluées. Une perspective de cette étude consiste à s'intéresser à un mode d'évaluation plus objectif, par exemple en définissant des métriques d'évaluation robustes à un certain nombre de variations musicales, ou en spécifiant plus les critères d'évaluation de la procédure subjective.

### 4.1.2 Conclusion

Dans cette section, nous avons décrit une technique basée sur l'alignement de séquences tonales pour l'identification de la meilleure répétition d'un segment choisi dans un morceau de musique. Dans le cadre de la reconstruction de données audio, un algorithme adapté a été présenté et évalué par des tests subjectifs. Le jugement des reconstructions calculées met en avant la pertinence de la méthode pour restaurer de manière perceptivement satisfaisante les données manquantes.

En choisissant un segment dans un morceau de musique, la méthode présentée permet donc d'identifier une répétition significative au sein du même morceau. Nous proposons à présent de relâcher la contrainte de choix d'un segment, et ainsi d'étudier un problème plus général de détection d'une répétition significative au sein d'un morceau : la répétition majeure.

## 4.2 Répétition majeure

Afin d'automatiser la détection de la structuration répétitive d'un morceau, il est nécessaire de désigner et d'extraire une répétition significative de celui-ci. Contrairement à la section précédente, on s'intéresse dans cette deuxième approche à l'identification d'une répétition musicalement significative dans un extrait audio *sans aucune pré-connaissance* sur celle-ci.

La répétition que nous choisissons d'étudier dans cette section est caractérisée par un critère d'optimalité. En effet, devant la complexité de la structuration répétitive des morceaux de musique, il semble naturel, sans connaissance *a priori* sur un morceau de musique, de rechercher d'abord sa "meilleure" répétition selon un certain critère.

On peut ainsi définir grossièrement le problème abordé ici comme l'extraction dans un morceau de musique quelconque de la répétition *prépondérante* du morceau, c'est-à-dire des deux extraits répétés les plus *significatifs*. Cette formulation imprécise du problème sous-entend l'importance de deux critères de sélection distincts : le *degré de similarité* et la *taille* des extraits répétés. Ainsi, pour deux répétitions de durées identiques, si l'information musicale est plus proche entre les occurrences de la première répétition qu'entre celles de la seconde, alors la première répétition peut être considérée comme plus significative que la seconde pour le morceau de musique. De la même manière, pour deux répétitions d'une quantité d'information musicale équivalente, si la durée des occurrences de la première est supérieure à celle des occurrences de la seconde, alors la première répétition peut être considérée comme plus significative que la seconde pour le morceau de musique. Chacun de ces deux critères présentent une importance dans le choix d'une répétition optimale.

En supposant connue une mesure de la similarité entre séquences musicales, on considère dans cette section la *répétition majeure* d'un morceau de musique, correspondant de manière informelle aux parties disjointes du morceau qui présentent le meilleur compromis entre un fort degré de similarité et une longueur importante.

La suite de cette section explicite la nature du compromis effectué, et définit formellement cette répétition.

Afin d'évaluer son sens musical, nous proposons ensuite de la comparer avec la structure répétitive perçue, telle qu'annotée par des experts. Enfin, nous décrivons une évaluation concrète de cette répétition pour indexer des morceaux de musique, dans le cadre de l'application à la détection de reprises.

### 4.2.1 Travaux antérieurs et motivation

Le problème d'extraction d'un segment caractérisé par sa répétition a été traité dans le cadre de la détection de segments représentatifs, avec pour objectif de produire un extrait caractéristique du morceau, ou *thumbnail* (voir Section 1.2.1). L'état de l'art se concentre essentiellement sur la détection d'une section caractéristique du style musical populaire occidental, le refrain [Got03, Got06, BW05, LC00, CF04].

Le refrain est un segment commun en musique populaire occidentale. Bien qu'il n'existe pas de définition précise de ce terme au sens de l'analyse musicale, le refrain peut être caractérisé par plusieurs constats dans certains morceaux. Par exemple, Peeters et Deruty [PD09] l'assimilent à *« une section d'un morceau qui inclut le chanteur, dans laquelle le titre du morceau est prononcé, et qui se répète au moins une fois dans le morceau »*. Depuis une analyse du contenu musical, le refrain est généralement identifié automatiquement à partir de la détection des sections les plus répétées d'un morceau de musique [Got06]. Cooper et Foote [CF04] proposent une telle détection, identifiant des sections comprenant refrains et couplets, en effectuant une analyse de la matrice d'*auto-distance* [Foo99] des descripteurs musicaux d'un morceau, puis en appliquant une méthode de *clustering* sur les segments obtenus pour sélectionner les plus fréquents. Logan et Chu [LC00] utilisent également une technique de *clustering* couplé à des modèles de Markov cachés pour regrouper de courts segments en fonction de caractéristiques audio, puis identifient une "phrase-clé" d'une durée fixée en sélectionnant la séquence la plus fréquente selon plusieurs heuristiques. De manière analogue, Bartsch et Wakefield [BW05] identifient un extrait de taille fixée par corrélation optimale de descripteurs audio en respectant une hypothèse sur leur évolution tout au long du morceau. Goto [Got03, Got06] parvient à repérer précisément les frontières des occurrences du refrain, potentiellement avec modulation, en optimisant une mesure de similarité établie sur différents niveaux de structures répétitives, tout en respectant des hypothèses arbitraires sur la position ou la durée probable du refrain. L'utilisation d'heuristiques dans l'ensemble de ces études pour décrire le refrain rend le problème de sa détection difficile à définir dans le cas général. Bien qu'affichant des résultats satisfaisants, ces techniques cherchent à travers une série d'hypothèses à résoudre un problème non formellement défini.

En outre, si le refrain est une structure caractéristique de la musique populaire occidentale, il ne convient pas à de nombreux styles musicaux qui comportent pourtant des structures répétitives fortes. Par exemple, on retrouve en musique classique des thèmes caractéristiques et récurrents tout au long d'un morceau mais ne correspondant pas aux hypothèses de répétitivité du refrain. À l'inverse, il est possible de citer de nombreuses œuvres considérées comme appartenant à la musique populaire mais ne comptant pas de refrain. Par exemple, Peeters et Deruty [PD09] soulignent qu'aucun refrain ne figure sur l'album *The Dark Side of The Moon* du

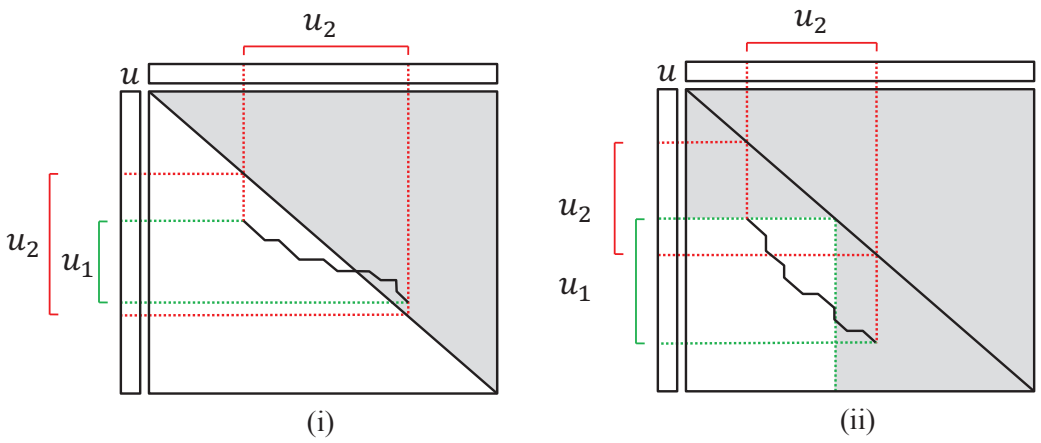


FIG. 4.10 – *Interprétation graphique de la contrainte de disjonction dans le graphe d'édition. (i) : Un chemin franchissant la diagonale du graphe représente l'alignement de séquences non disjointes. (ii) : Un chemin non inclus dans un sous-graphe rectangulaire représente également l'alignement de séquences non disjointes.*

groupe *Pink Floyd*. Dans la même étude, les auteurs remarquent également qu'avec une telle définition, le refrain n'apparaît que dans moins de la moitié de leur base de données de 112 morceaux de styles et époques variées [PD09]. L'analyse du refrain réduit donc l'ensemble des données musicales exploitables à une petite partie de la musique populaire occidentale.

C'est pourquoi on se propose dans la suite d'étudier la *répétition majeure*, définie de manière *moins intuitive* mais *plus générale* que le refrain. L'objectif des sections qui suivent est de formaliser cette répétition, de proposer un algorithme efficace permettant de l'extraire puis de justifier sa pertinence vis-à-vis des répétitions perçues dans un morceau de musique.

### 4.2.2 Modélisation

Soit $u$ une séquence de symboles. La répétition majeure de $u$ correspond aux deux facteurs disjoints dans $u$ de score de similarité locale optimal. Formellement, on cherche donc le *score de répétition majeure* $s^{\circledast}(u)$, donné par la formule :

$$s^{\circledast}(u) = \max_{(v,w)\in\mathcal{S}(u)^2 \ : \ v\cap_u w=\varepsilon} s^*(v,w) \quad . \tag{29}$$

On note $u_1^{\circledast}$ et $u_2^{\circledast}$ les facteurs disjoints dans $u$ effectivement alignés, c'est-à-dire tels que $s^{\circledast}(u) = s(u_1^{\circledast},u_2^{\circledast})$.

### 4.2.3 Algorithme

Une solution au problème d'identification des facteurs disjoints de similarité optimale a été proposée par Miller [Mil92], puis améliorée peu après notamment par Kannan et Myers [KM93] et Benson [Ben95].

Dans le cas où la condition de disjonction est relâchée, Miller [Mil92] remarque que le problème peut aisément être résolu en temps quadratique en effectuant une

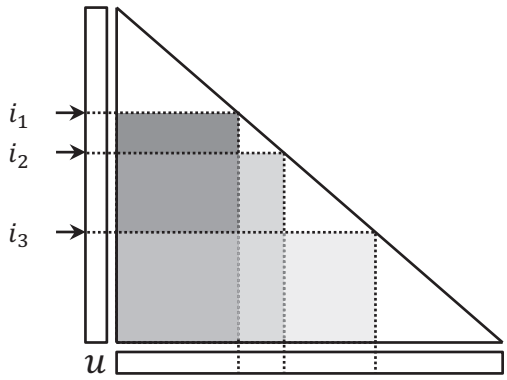


FIG. 4.11 – *Illustration du premier algorithme d'alignement de séquences disjointes.*

modification simple de l'algorithme d'alignement local [SW81]. La difficulté algorithmique du problème réside donc dans la condition de disjonction.

#### 4.2.3.1 Premier algorithme

Soient $u$ une séquence de symboles de longueur $n$ et $\mathcal{G}$ le graphe d'édition de la séquence $u$ vers elle-même. Supposons connu le chemin $\mathcal{P}$ dans $\mathcal{G}$ correspondant à une transcription locale optimale de séquences disjointes dans $u$. On note $(i,j)$ les coordonnées du point de départ de $\mathcal{P}$ dans $\mathcal{G}$, et $(k,l)$ les coordonnées de son point d'arrivée. En d'autres termes, $\mathcal{P}$ représente l'alignement optimal de $u[i \dots k]$ et $u[j \dots l]$. Comme représenté en Figure 4.10, un chemin franchissant la diagonale de $\mathcal{G}$ représente un alignement de séquences possédant au moins un symbole en commun dans $u$. Par conséquent, $\mathcal{P}$ ne franchit pas la diagonale de $\mathcal{G}$. $\mathcal{G}$ étant symétrique par rapport à sa diagonale, on suppose que $\mathcal{P}$ est contenu dans le triangle inférieur de $\mathcal{G}$ (quitte à considérer son image par symétrie), et notamment que $j < i$ et $l < k$. En d'autres termes, le facteur $u[j \dots l]$ est antérieur au facteur $u[i \dots k]$ dans $u$. Puisque ces deux facteurs sont disjoints dans $u$, on a $l < i$. Graphiquement, cette condition implique que le chemin $\mathcal{P}$ ne se termine pas dans une colonne d'indice $l$ supérieur à celui de la ligne $i$ à laquelle il commence. Comme représenté en Figure 4.10-(ii), si $\mathcal{P}$ ne respectait pas cette condition, il représenterait nécessairement un alignement de facteurs *non disjoints* dans $u$.

Par conséquent, le chemin optimal avec disjonction dans $\mathcal{G}$ est contenu dans un sous-graphe de $\mathcal{G}$ rectangulaire et délimité par la ligne $i$ et la colonne $i$, pour $i \in [\![1,|u|]\!]$. Cette remarque, représentée en Figure 4.11 pour quelques valeurs de $i$, conduit à la définition de l'Algorithme 1 de résolution du problème de la répétition majeure.

À chaque étape $i$, une matrice rectangulaire de taille $i \cdot (n-i)$ est calculée. Le nombre de coefficients de programmation dynamique calculés par l'algorithme 1 est donc donné par :

$$\sum_{i=1}^{n-1} i(n-i) = \frac{n(n-1)(n+1)}{6}. \tag{30}$$

Par conséquent, la complexité temporelle de ce premier algorithme est de $\mathcal{O}(n^3)$, avec précisément $\frac{n(n-1)(n+1)}{6}$ coefficients de programmation dynamique à évaluer.

**algorithme 1** Répétition majeure, premier algorithme [Mil92]

**ENTRÉE** Une séquence $u$
**SORTIE** Le score de répétition majeure $s^{\circledast}$ et les deux facteurs $u_1^{\circledast}$ et $u_2^{\circledast}$
$s \leftarrow 0$, $v \leftarrow \varepsilon$, $w \leftarrow \varepsilon$
**pour** $i = 1$ à $|u| - 1$ **faire**
  $v \leftarrow u[1 \dots i]$
  $w \leftarrow u[i+1 \dots |u|]$
  $s^* \leftarrow s^*(v,w)$
  **si** $s^* > s$ **alors**
    $s \leftarrow s^*$, $u_1 \leftarrow v^*$, $u_2 \leftarrow w^*$
$s^{\circledast} \leftarrow s$, $u_1^{\circledast} \leftarrow u_1$, $u_2^{\circledast} \leftarrow u_2$
**retour** $s^{\circledast}$, $u_1^{\circledast}$, $u_2^{\circledast}$

#### 4.2.3.2 Deuxième algorithme

Miller [Mil92] propose un algorithme plus efficace en pratique en modifiant le calcul de l'alignement local par programmation dynamique. Comme décrit en Section 3.1.3.3, l'alignement local entre deux séquences $u$ et $v$ est obtenu en calculant par récurrence sur $i$ et $j$ le score de similarité optimal $s^+_{i,j}$ entre un suffixe de $u[1 \dots i]$ et $v[1 \dots j]$. Dans le cas de recherche de facteurs disjoints dans une seule séquence $u$ de taille $n$, Miller [Mil92] définit le score de similarité optimal $s^+_{i,j,k}$ entre un suffixe de $u[k \dots i]$ et $u[1 \dots j]$. En d'autres termes, pour trois indices $i$,$j$,$k$ dans une séquence $u$, $s^+_{i,j}$ désigne le score du meilleur chemin se terminant ligne $i$ et colonne $j$ dans le graphe d'édition, tandis que $s^+_{i,j,k}$ désigne le score du meilleur chemin se terminant ligne $i$, colonne $j$ et commençant à la ligne $k$ dans le graphe d'édition. Le calcul de la similarité locale dans $u$ est alors effectué selon une récurrence sur $(i,j,k) \in [\![0,n]\!]^3$. Son initialisation est donnée la proposition :

**Proposition 8 (Initialisation)** *Pour $j \leq k \leq i$,*

$$\begin{cases} s^+_{0,j,k} = 0 \\ s^+_{i,0,k} = 0 \\ s^+_{i,j,0} = 0 \end{cases}.$$

La récurrence se fait alors selon la proposition :

**Proposition 9** *Soient $\lambda$ un schéma de scores de pondération bien formé, et $\phi$ le symbole spécial d'alignement. Pour $j \leq k \leq i$,*

$$s^+_{i,j,k} = \max \begin{cases} s^+_{i-1,j,k} & + & \lambda(u[i],\phi) & si\ i > k \\ s^+_{i,j-1,k} & + & \lambda(\phi,u[j]) & si\ i \geq k \\ s^+_{i-1,j-1,k} & + & \lambda(u[i],u[j]) & si\ i > k \\ 0 & & & si\ i = k \end{cases}.$$

Le chemin optimal dans le graphe d'édition ainsi calculé débute à la ligne $k$ et se termine à la ligne $i$ et la colonne $j$. L'inégalité $k \leq i$ indique l'ordre naturel

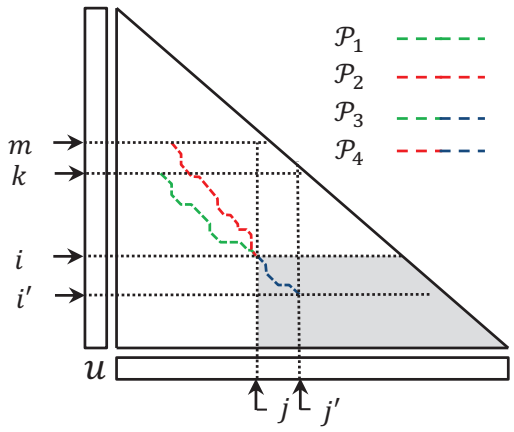


FIG. 4.12 – *Exemples de chemins dans le graphe d'édition et amélioration du calcul du chemin optimal telle que proposée par Miller [Mil92] (voir texte).*

des lignes de départ et d'arrivée du chemin dans le graphe d'édition. L'inégalité $j \leq k$ assure que le chemin optimal est bien situé dans le sous-graphe rectangulaire délimité par la ligne $k$ et la colonne $k$. Enfin, par transitivité, l'inégalité $i \leq j$ assure que seul le triangle inférieur du graphe d'édition de $u$ vers $u$ est considéré.

Le score de répétition majeure est alors donné par :

$$s^{\circledast}(u) = \max_{(i,j,k)\in[\![1,n]\!]^3, i\leq j\leq k} s^{+}_{i,j,k} \quad . \tag{31}$$

Le calcul de $s^{\circledast}$ avec l'algorithme de Miller est effectué par récurrence sur $i$,$j$ et $k$ ; comme le premier algorithme proposé, sa complexité temporelle théorique est donc de $\mathcal{O}(n^3)$.

Cependant, Miller [Mil92] relève que la complexité peut en pratique être ramenée à l'ordre quadratique. Pour décrire cette amélioration, on introduit quatre chemins de la manière suivante :

- Soit $\mathcal{P}_1$ le meilleur chemin se terminant en ligne $i$, colonne $j$ et commençant en ligne $k$ ;
- Soit $\mathcal{P}_2$ le meilleur chemin se terminant en ligne $i$, colonne $j$ et commençant en ligne $m < k$ ;
- Soit $\mathcal{P}_3$ le meilleur chemin se terminant en ligne $i' > i$, colonne $j' > j$, commençant en ligne $k$ et passant par le point $(i,j)$ ;
- Soit $\mathcal{P}_4$ le meilleur chemin se terminant en ligne $i' > i$, colonne $j' > j$, commençant en ligne $m$ et passant par le point $(i,j)$.

La Figure 4.12 représente graphiquement un exemple pour ces quatre chemins. Miller remarque que si $s^{+}_{i,j,k} > s^{+}_{i,j,m}$, alors $s^{+}_{i',j',k} > s^{+}_{i',j',m}$. En d'autres termes, si $\mathcal{P}_1$ est meilleur que $\mathcal{P}_2$, alors $\mathcal{P}_3$ est meilleur que $\mathcal{P}_4$. On dit que la ligne $k$ *domine* la ligne $m$ en $(i,j)$. Ainsi, il n'est pas nécessaire de considérer la ligne dominée $m$ dans le calcul de tout chemin optimal passant par le point $(i,j)$ et se terminant en une ligne $i' > i$ et une colonne $j' > j$ (zone grisée de la Figure 4.12).

Miller propose donc une optimisation du calcul de $s^{+}$ en conservant pour chaque couple d'indices $(i,j)$ une liste des *candidats* $(c,s^{+}_{i,j,c})$ telle que la ligne $c$ n'est dominée par aucune autre ligne en $(i,j)$. Cette liste est triée dans l'ordre des $c$ croissants,

donc dans l'ordre des $s^{+}_{i,j,c}$ décroissants. L'évaluation par programmation dynamique d'un score optimal en un point $(i,j)$, décrite par l'Équation 9, n'est alors effectuée que sur les valeurs de $k$ non dominées, indiquées par les listes de candidats des coefficients voisins. Or, l'auteur remarque qu'en pratique le nombre de lignes non dominées en un point donné semble être constant, et lorsque c'est le cas le calcul de la liste de candidats à partir des coefficients voisins demande un temps de calcul constant [Mil92, KM93]. Par conséquent, Miller observe une complexité temporelle de cet algorithme de l'ordre de $\mathcal{O}(n^2)$ en pratique, même si celle-ci peut toujours atteindre $\mathcal{O}(n^3)$ dans le pire des cas où aucune ligne n'est dominée.

#### 4.2.3.3 Autres optimisations

Des travaux ultérieurs à l'étude de Miller [Mil92] ont réduit la complexité temporelle du problème à $\mathcal{O}(n^2 \log^2 n)$ pour Kannan et Myers [KM93], ou encore $\mathcal{O}(n^2 \log n)$ pour Schmidt [Sch95]. Cependant, ces algorithmes introduisent des structures de données spécifiques et s'avèrent particulièrement plus délicats à mettre en œuvre. Notre implémentation de l'algorithme d'identification de la répétition majeure suit donc la méthode de Miller. On notera que bien que la complexité théorique de cet algorithme ne soit pas optimale, le comportement quadratique de l'algorithme de Miller le rend performant en pratique pour analyser des séquences musicales.

### 4.2.4 Évaluation

Nous proposons d'évaluer la répétition majeure sur des séquences tonales afin de qualifier sa pertinence pour décrire des données musicales. Les résultats préliminaires de l'évaluation décrite dans cette section ont été publiés dans les actes de la *Joint Conference on Digital Libraries* (JCDL) [MHRF11a].

L'algorithme de Miller est appliqué aux séquences de descripteurs tonaux sur un ensemble de morceaux de musique. Le schéma de scores de pondération utilisé pour les calculs de similarité locale est identique à celui établi pour l'évaluation de la similarité tonale en Section 3.2. La taille des trames de signal est déterminée empiriquement afin de représenter des sections comprenant une information tonale pertinente et concordante avec l'échelle de description de la répétition majeure. Chaque trame est ainsi calculée sur un intervalle de 741ms, avec un avancement de 370ms entre deux trames successives (chevauchement des trames de moitié).

#### Bases de données

Pour les besoins de l'évaluation, on utilise plusieurs bases de données musicales manuellement annotées et considérées comme vérité-terrain. Le premier jeu de données, noté $\mathcal{D}_s$, est constitué d'annotations des segments structurels composant un ensemble de 180 morceaux du groupe *The Beatles*. Cette base de données est issue de plusieurs projets de recherche, et plus précisément décrite en Section 5.3.4.2. Le second jeu de données, noté $\mathcal{D}_c$, consiste en une annotation manuelle des *accords* de ces mêmes morceaux, et provient d'un complément de données fourni par le projet *OMRAS2 Metadata* [MCD+09] [1]. Dans cette base de données, chaque mor-

1. http://www.isophonics.net/content/reference-annotations

| Évaluation | $R_p$ | $P_p$ | $F_p$ | $S_u$ | $S_o$ |
|---|---|---|---|---|---|
| RM Audio/RM Accords | 81.6 | 77.7 | 78.5 | 77.3 | 73.5 |

TAB. 4.3 – *Robustesse de l'algorithme de détection de la répétition majeure face aux descripteurs audio. Les scores (pourcentages) correspondent aux moyennes sur la base de tests, et sont obtenus en comparant les répétitions majeures (RM) calculées depuis l'audio à celles calculées depuis les annotations manuelles des accords.*

ceau est représenté par une séquence de symboles sur l'alphabet des accords $\Sigma_{\text{acc}}^{r+m}$, chaque symbole représentant la fondamentale et le mode d'un accord (selon une construction analogue à la Section 3.3.3.1).

**Validation de l'algorithme sur des séquences de descripteurs musicaux**

La base de données $\mathcal{D}_c$ comporte un ensemble de séquences de symboles décrivant les accords de morceaux de musique annotés par des experts. Afin d'évaluer la pertinence de la méthode sur des descripteurs audio, on propose donc d'appliquer l'algorithme de Miller sur ces séquences annotées d'une part, puis sur les séquences de descripteurs audio d'autre part, avant de comparer les résultats obtenus. Une convergence des répétitions identifiées soulignerait non seulement la qualité du descripteur audio à représenter les accords, mais aussi le succès de la méthode à identifier la répétition majeure à partir de descripteurs audio.

Pour chaque morceau de la base de tests, la répétition majeure est donc calculée par l'algorithme de Miller à partir de la séquence d'accord annotés et à partir de la séquence de chromas. Les deux segmentations obtenues sont alors comparées en utilisant deux métriques standard d'évaluation décrites en 5.3.4.3 : la comparaison trame-à-trame (scores $R_p$, $P_p$ et $F_p$) et la métrique de sur- et sous-segmentation (scores $S_u$ et $S_o$). Les résultats de cette première évaluation sont indiqués dans le Tableau 4.3. Les scores obtenus soulignent une forte correspondance entre les segmentations depuis ces deux différentes sources, avec une F-mesure trame-à-trame moyenne de 78.5% pour un rappel de 81.6%. Les scores élevés de sur-segmentation et de sous-segmentation aux valeurs respectives de 77.3% et 73.5% appuient cette conclusion. Leur faible écart indique en outre que les deux segmentations comparées semblent bien décrire un niveau structurel similaire. La méthode semble donc bien robuste au passage aux descripteurs audio.

Malgré cette forte concordance, les segmentations calculées à partir des accords et à partir de l'audio ne semblent pas correspondre exactement. Cette différence peut être due à la segmentation imposée par le découpage du signal en trame, qui est susceptible de ne pas correspondre à l'annotation des accords. À l'inverse, le découpage en accords étant réalisé par une oreille humaine, il peut présenter une relative imprécision temporelle non reproduite par l'analyse du signal. Cette non correspondance peut également être reliée à différents facteurs liés au signal audio ; en particulier, la construction du chroma peut introduire des incohérences locales, en identifiant par exemple une harmonie non pertinente sur une section n'en comportant pas, telle qu'un solo de batterie.

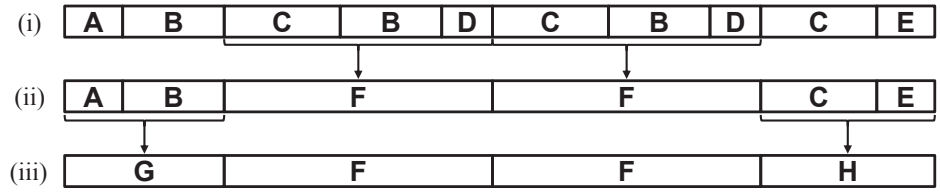


FIG. 4.13 – *Transformation appliquée sur l'annotation représentée en (i). (ii) illustre le résultat de la première étape de transformation de la plus longue répétition annotée, puis (iii) présente le résultat après application de la seconde étape de rassemblement des parties non modifiées.*

### Concordance avec une structure répétitive perçue

On propose à présent de comparer la répétition majeure analysée dans un morceau à sa structure perçue. La répétition majeure est définie selon un critère d'optimalité du score d'alignement local. Ainsi, les parties identifiées par cette répétition sont déterminées par compromis optimal sur la *longueur* des séquences comparées et sur leur *ressemblance*, grâce aux techniques d'alignement. Dans cette section, on souhaite comparer ce résultat à une annotation structurelle. Or, cette dernière ne quantifie la ressemblance entre les parties structurelles identifiées que dans une relation binaire : deux parties similaires sont représentées par la même étiquette, et deux parties non similaires sont représentées par des étiquettes distinctes. On en déduit que si la répétition majeure correspond bien à une répétition dans l'annotation de la structure, cette dernière ne peut être caractérisée par une *longueur optimale*. Dans la suite de cette section, on cherche donc à évaluer dans quelle mesure la répétition majeure identifiée dans un morceau de musique correspond à la plus longue répétition de son annotation structurelle.

Chaque annotation de la base $\mathcal{D}_s$ correspond à un certain niveau de description des répétitions de chaque morceau. Dans cette annotation, la plus longue répétition n'est pas directement accessible. Afin d'obtenir la plus longue répétition annotée, on applique une transformation simple permettant de la déduire des données disponibles. Cette transformation, inspirée du processus de *roll-up* défini par Chai [Cha05, p.58], est décrite par les deux opérations successives :

1. Trouver dans l'annotation la plus longue suite de parties consécutives exactement répétées, rassembler les deux occurrences de cette répétition et leur assigner une nouvelle étiquette ;
2. Rassembler toutes les parties non modifiées par l'étape 1 en un nombre minimal de nouvelles parties, et assigner à chacune d'entre elles une nouvelle étiquette.

La Figure 4.13 décrit ce processus sur un exemple. À partir de l'annotation représentée en (i), les parties CBD correspondant à la plus longue répétition annotée sont rassemblées en (ii) et assignées à une nouvelle étiquette F (étape 1), puis les parties restantes sont rassemblées en (iii) en un nombre minimal de parties, assignées aux nouvelles étiquettes G et H. À l'issue de cet algorithme simple, l'annotation de chaque morceau ne contient qu'une répétition correspondant à la plus longue répétition initialement renseignée.

| Évaluation | $R_p$ | $P_p$ | $F_p$ | $S_u$ | $S_o$ |
|---|---|---|---|---|---|
| (i) RM Audio/RL Annotation | 76.0 | 76.9 | 75.4 | 73.3 | 72.7 |
| (ii) RM Accords/RL Annotation | 75.0 | 76.7 | 74.6 | 72.9 | 72.9 |

TAB. 4.4 – *Concordance des répétitions majeures avec les plus longues répétitions annotées. Les scores (pourcentages) correspondent aux moyennes sur la base de tests. La ligne (i) est obtenue en comparant les répétitions majeures (RM) calculées depuis l'audio et les plus longues répétitions (RL) déduites des annotations. La ligne (ii) est obtenue en comparant les RM calculées depuis les annotations manuelles des accords et les RL déduites des annotations.*

Cette transformation est appliquée à chaque annotation de la base $\mathcal{D}_s$ pour former la base modifiée $\widetilde{\mathcal{D}_s}$ décrivant les plus longues répétitions annotées. On remarque que cette transformation n'introduit aucune approximation ou information supplémentaire, mais consiste en une simplification des informations annotées.

L'évaluation des répétitions majeures estimées est alors effectuée en comparant pour chaque morceau la répétition majeure obtenue depuis l'audio avec la plus longue répétition annotée. Les résultats sur la base de données sont présentés en ligne (i) du Tableau 4.4, sous forme de moyennes exprimées en pourcentages. Ceux-ci soulignent une bonne concordance des estimations et des annotations, avec F-mesure trame-à-trame moyenne de 75.4% pour un rappel de 76%. Comme précédemment, l'importance et la proximité des scores de sous- et de sur-segmentation, respectivement de 73.3% et 72.7%, témoignent de cette concordance et soulignent un résultat sur un même niveau de description des structures répétitives. Afin de proposer une interprétation plus précise de ces scores et de leur non maximalité, les répétitions majeures obtenues depuis les annotations d'accords sont également comparées aux plus longues répétitions annotées. Cette évaluation est présentée en ligne (ii) du Tableau 4.4. Les résultats soulignent à nouveau une bonne correspondance avec les plus longues répétitions annotés. De plus, les scores de concordance obtenus semblent particulièrement proches de ceux obtenus pour l'évaluation des répétitions majeures calculées depuis l'audio, avec un écart de 0.8% pour la F-mesure trame-à-trame, 0.4% pour le score de sous-segmentation et 0.2% pour le score de sur-segmentation. Ce résultat indique donc que la correspondance entre la répétition majeure obtenue depuis l'audio et la plus longue répétition annotée est comparable à la celle entre la répétition majeure obtenue depuis les accords et la plus longue répétition annotée. Depuis les deux descriptions de l'harmonie, estimées depuis l'audio ou annotées par des experts, la concordance semble donc importante et équivalente. Ce constat suggère que la non-maximalité des scores obtenus est liée à un aspect des structures répétitives annotées ne pouvant être déduit de l'harmonie, et met donc en avant une limitation de la représentation tonale pour décrire les structures annotées dans $\mathcal{D}_s$.

La Figure 4.14 représente trois exemples de répétitions majeures calculées sur des morceaux du groupe *The Beatles* mis en correspondance avec la vérité structurelle répétitive correspondante. Dans l'exemple (i), correspondant au morceau *The Night Before*, la répétition identifiée correspond à la plus longue répétition annotée `CPC`. Il convient de noter que la répétition `CCP` est de taille identique. Dans l'exemple (ii), correspondant au morceau *Fixing a Hole*, les frontières de la répétition ma-

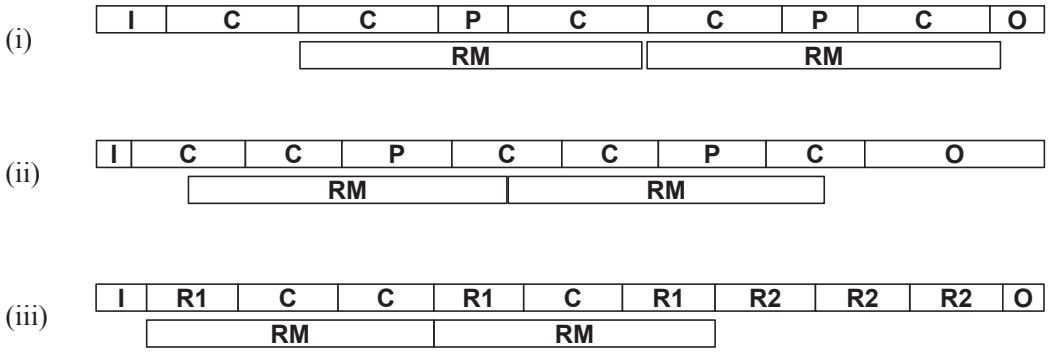


FIG. 4.14 – *Exemples illustrant la correspondance entre l'annotation structurelle (première ligne) et la répétition majeure calculée (deuxième ligne). Les trois exemples sont composés par The Beatles : (i) The Night Before, (ii) Fixing a Hole, (iii) Please Mister Postman. I : introduction ; O : conclusion (outro) ; C : couplet ; P : pont ; R, R1, R2 : refrain ; RM : répétition majeure.*

jeure ne correspondent pas à des frontières dans l'annotation, bien que la répétition identifiée corresponde effectivement à une répétition annotée. Enfin, l'exemple (iii) correspondant au morceau *Please Mister Postman* présente un exemple où la répétition majeure ne correspond pas à une répétition annotée, cette erreur étant due au fait que le motif `R1` et le motif `C` possèdent une information tonale très similaire.

#### Bilan expérimental

Les évaluations de l'algorithme d'identification de la répétition majeure soulignent le succès de son application aux séquences de descripteurs audio, avec une forte correspondance des répétitions identifiées depuis l'audio ou depuis des accords annotés. De plus, la comparaison avec une annotation des structures répétitives souligne que la répétition majeure correspond à la plus longue répétition perçue, avec une précision moyenne de 76.9% et un rappel moyen de 76%. Enfin, des résultats équivalents calculés depuis l'annotation des accords expliquent l'efficacité non optimale de l'algorithme d'identification par une possible limitation de la représentation tonale pour décrire les répétitions telles qu'annotées dans la vérité terrain.

### 4.2.5 Application à l'indexation pour la recherche de reprises

La section précédente montre que la répétition majeure correspond à une certaine perception de structures répétitives dans la musique. Dans cette section, on propose d'étudier une application pratique de cette répétition pour l'analyse du contenu musical.

La répétition majeure effectue, comme défini précédemment, un compromis entre les deux parties les plus grandes et les deux parties les plus similaires dans un morceau de musique. Elle est donc prépondérante dans le morceau considéré, chacune de ses occurrences contenant une information récurrente. On propose alors d'évaluer dans quelle mesure les répétitions ainsi identifiées dans deux morceaux de musique caractérisent la similarité entre ces deux morceaux. En d'autres termes, on souhaite

évaluer dans quelle mesure la similarité entre deux morceaux de musique correspond à la similarité entre les répétitions majeures de ces morceaux.

Comme expliqué dans le chapitre précédent, l'étude de la similarité peut être efficacement évaluée dans le cadre applicatif de l'identification de reprises. Dans cette section, nous étudions donc une application de la répétition majeure comme indexation du système de détection de reprises décrit en Section 3.2.1. Les résultats préliminaires de cette étude ont été publiés dans les actes de la conférence *Special Interest Group on Information Retrieval* [MHRF11b].

#### 4.2.5.1 Principe

Une étude présentée en Section 3.3 décrit une méthode heuristique permettant d'accélérer la technique d'alignement en conservant une précision raisonnable de la détection des reprises. Dans cette section, nous proposons une approche différente permettant d'accélérer ce calcul en utilisant la répétition majeure.

Gómez *et al.* montrent en 2006 [GOH06] que réduire l'alignement à des extraits de morceaux de musique permet de conserver une détection précise des reprises. Plus précisément, leur approche consiste à extraire par une méthode adaptée de Goto [Got03] deux répétitions prépondérantes d'une longueur fixée dans chaque morceau, puis à comparer ceux-ci pour la recherche de reprises. Les auteurs rapportent alors [GOH06] des résultats d'une précision équivalente pour la détection des reprises sur une durée de segment fixée à environ 25 secondes, et d'une précision accrue dans le cas d'une sélection manuelle des segments.

À l'image du travail de Gómez *et al.*, nous proposons dans cette section de compresser la représentation des morceaux comparés en ne prenant en compte qu'un extrait, leur répétition majeure. Notre principale contribution par rapport à cette étude consiste à ne pas fixer de durée d'indexation, celle-ci étant déterminée par la répétition majeure. Au vu de la grande variété de reprises utilisées, il semble en effet pertinent de considérer que la durée des segments similaires dans les reprises ne reste pas constante pour chaque classe. Par exemple, la classe de reprises *The Pink Panther* compte de nombreuses réinterprétations jouées par différents orchestres et proches de la version canonique, où le thème est répété de nombreuses fois ; en revanche, elle se compose également d'improvisations jazz et de remix qui se contentent de seulement une ou deux répétitions du thème musical d'origine.

Soient $u$ et $v$ deux séquences de chromas de tailles respectives $n$ et $m$ représentant deux morceaux de musique distincts. Comme expliqué en Section 3.2.1, le système de détection de reprises estime la similarité entre $u$ et $v$ à partir du score d'alignement :

$$\mathrm{sim}(u,v) = s^*_\gamma(u,v). \tag{32}$$

Le nombre de cases de programmation dynamique à calculer correspond alors au produit des tailles des séquences $u$ et $v$.

L'indexation des données par leur répétition majeure consiste à calculer la similarité tonale entre les deux séquences $u$ et $v$ comme le score :

$$\mathrm{sim}_{\mathrm{MR}}(u,v) = s^*_\gamma(u^{\circledast}_1, v^{\circledast}_1), \tag{33}$$

où $u^{\circledast}_1$ et $v^{\circledast}_1$ correspondent aux premières occurrences des répétitions majeures respectives de $u$ et $v$. Le nombre de cases de programmation dynamique intervenant dans ce calcul correspond au produit des tailles des répétitions majeures, inférieur

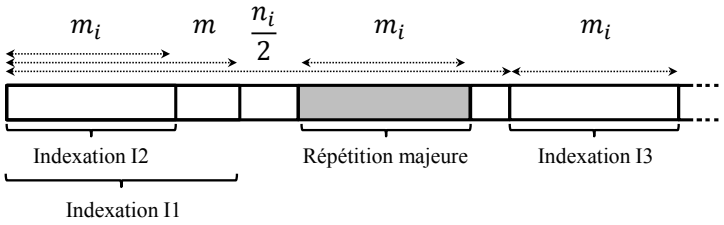


FIG. 4.15 – *Illustration des différentes techniques d'indexation sur le début d'une séquence.*

au produit des tailles de séquences. Si la complexité théorique demeure identique, l'obtention de $\text{sim}_{\text{MR}}$ implique donc en pratique un nombre de calculs inférieur que celle de sim.

Le gain de temps de calcul et la précision du système de détection avec indexation sont évalués en pratique sur des données audio dans la section suivante.

#### 4.2.5.2 Évaluation

Afin de faciliter l'interprétation des résultats, la base de données d'évaluation choisie est identique à celle utilisée pour la recherche de reprises sans indexation, présentée en Section 3.2.2. Elle se compose d'un ensemble de *versions*, noté $\mathcal{D}_{\text{V}}$, d'un ensemble de *reprises*, noté $\mathcal{D}_{\text{r}}$ et d'un ensemble de morceaux aléatoirement choisis et absents des ensembles précédents, noté $\mathcal{D}'$. Comme précédemment, chaque classe de reprises n'est jamais comparée à une autre classe de reprises, afin d'éviter d'introduire un biais dans la comparaison (voir Section 3.2.2).

Pour estimer la pertinence de la sélection effectuée par l'algorithme de répétition majeure, et à l'instar de l'évaluation proposée par Gómez *et al.* [GOH06], on introduit des méthodes d'indexation arbitraires des morceaux de musique. Ces indexations arbitraires sont construites de telle sorte qu'elles réduisent autant la quantité d'information que l'indexation par la répétition majeure, afin de permettre de comparer les résultats.

Plus précisément, pour le morceau $i$ de la base de données, on note $n_i$ la taille de la séquence de descripteurs représentant tout le morceau $i$, et $m_i$ la taille de la première occurrence de sa répétition majeure. Indexer par cette répétition induit alors une réduction de la quantité d'information dans le rapport $k_i = \frac{n_i}{m_i}$. On pose $m = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} m_i$ la moyenne sur $\mathcal{D}$ des tailles de répétitions majeures, et $k = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} k_i$ la moyenne sur $\mathcal{D}$ des rapports d'indexation par la répétition majeure. On introduit alors trois méthodes d'indexation arbitraire I1, I2 et I3 réduisant chacune la quantité d'information d'un facteur $k$. Elles sont définies de la manière suivante :

- I1 indexe chaque séquence $u_i$ par $u[1 \dots m]$
- I2 indexe chaque séquence $u_i$ par $u[1 \dots m_i]$
- I3 indexe chaque séquence $u_i$ par $u[\frac{n_i}{2} \dots \frac{n_i}{2} + m_i]$

La Figure 4.15 représente les parties sélectionnées par les méthodes I1, I2 et I3 sur un exemple. À partir de ces méthodes d'indexation, la similarité entre deux

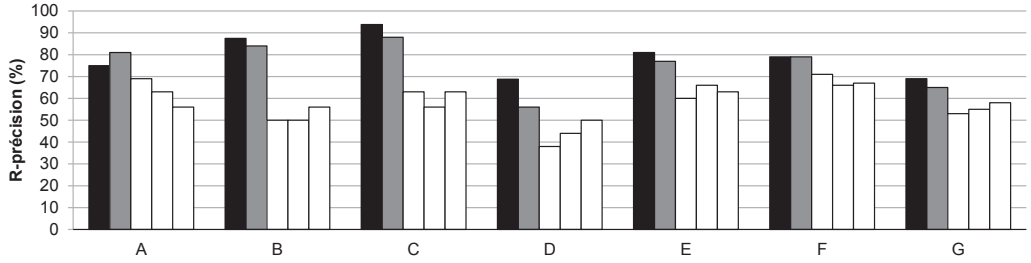


FIG. 4.16 – *Distribution des R-précisions avec différentes indexations calculées pour chaque classe de versions. Les lettres en abscisse désignent la classe considérée (voir Tableau 3.2). Noir : aucune indexation, gris : indexation par la répétition majeure, blanc : de gauche à droite, indexations par les méthodes arbitraires I1, I2 et I3.*

séquences $u$ et $v$ est finalement estimée à partir des scores d'alignement :

$$\begin{array}{l} \mathrm{sim}_{\mathrm{I1}}(u,v) = s^*(u[1\ldots m],v[1\ldots m]) \text{ pour la méthode I1 ;} \\ \mathrm{sim}_{\mathrm{I2}}(u,v) = s^*(u[1\ldots m_i],v[1\ldots m_j]) \text{ pour la méthode I2 ;} \\ \mathrm{sim}_{\mathrm{I3}}(u,v) = s^*(u[\frac{n_i}{2}\ldots\frac{n_i}{2}+m_i],v[\frac{n_j}{2}\ldots\frac{n_j}{2}+m_j]) \text{ pour la méthode I3} \end{array} , \quad (34)$$

où $i$ correspond à l'indice dans $\mathcal{D}$ du morceau représenté par $u$, et $j$ correspond à l'indice dans $\mathcal{D}$ du morceau représenté par $v$.

### Précision

La précision de la méthode d'indexation est d'abord évaluée sur la base de test composée uniquement de versions. Le protocole d'évaluation du système de détection des reprises est identique à celui présenté en Section 3.2.1. Pour chaque classe composée de $N$ versions, on calcule le taux de R-précision, décrit en Section 3.2.2 et correspondant à la précision du système pour les $N$ morceaux les plus similaires. On rappelle que dans ce cas la R-précision correspond à la fois au taux de précision et au taux de rappel du système [MRS08].

La ligne (i) du Tableau 4.5 présente les valeurs de R-précision obtenues en moyenne sur toutes les classes de versions. La précision du système avec indexation par la répétition majeure est de 75.7%, soit un score inférieur de 3.5% en moyenne par rapport au système non indexé. Les indexations arbitraires produisent des scores

| | Aucune | **RM** | I1 | I2 | I3 |
|---|---|---|---|---|---|
| (i) Versions | 79.2 | **75.7** | 57.7 | 57.1 | 59.0 |
| (ii) Reprises | 53.3 | **48.0** | 32.1 | 31.1 | 30.1 |
| (iii) Calcul similarité | 324 min | **41 min** | 41 min | 41 min | 41 min |
| (iv) Calcul indexation | - | **23 min** | - | - | - |

TAB. 4.5 – *Valeurs de R-précisions moyennes obtenues (pourcentages) et temps moyen de calcul requis par classe pour chaque méthode d'indexation : aucune indexation ("Aucune"), indexation par la répétition majeure ("MR"), et indexations arbitraires 1 à 3 ("I1", "I2" et "I3").*

plus faibles, compris entre 57.1% et 59%. Le détail de la distribution des scores pour chaque classe est représenté en Figure 4.16 (pour rappel, la liste des classes figure dans le Tableau 3.2). On constate que pour chacune des classes, les indexations arbitraires (en blanc) diminuent la précision du système d'identification des versions non indexé (en noir) de 5 à 25%, et s'avèrent toujours moins efficaces qu'une indexation par la répétition majeure (en gris). Par exemple, pour la classe C (*The House of the Rising Sun*), indexer par la répétition majeure diminue légèrement la précision du système d'identification non indexé, avec une précision de 88% pour un système indexé et 93.8% sans indexation ; en revanche, nos indexations arbitraires affichent des scores sensiblement plus faibles, avec des précisions respectives de 63, 56 et 63% en moyenne pour I1, I2 et I3. Ces résultats suggèrent que la répétition majeure semble plus pertinente pour indexer un système d'identification de versions qu'une méthode arbitraire. On remarque néanmoins que, bien que la technique d'indexation proposée semble discriminante, la suppression d'information engendre une diminution de la précision globale du système.

L'indexation est ensuite effectuée sur la base de tests composée de reprises au sens large. La ligne (ii) du Tableau 4.5 présente les valeurs de R-précision obtenues en moyenne sur toutes les classes de reprises. La reprise est une généralisation de la notion de version à tout rendu sonore de la même pièce ; en conséquence, la tâche des systèmes d'identification s'avère plus complexe et les scores sont plus faibles que dans le cas des versions, avec une précision de 53.3% pour le système non indexé. Cependant, le score de précision de l'indexation par la répétition majeure est à nouveau sensiblement supérieur à celui des indexations arbitraires, avec un taux de 48% pour la méthode proposée contre 32.1% pour la plus performante des indexations arbitraires. La distribution des scores pour chaque classe de reprises, représentée en Figure 4.17, conduit à nouveau au constat de précision systématiquement accrue de l'indexation proposée par rapport aux indexations arbitraires, avec un taux entre 1 et 11% supérieur.

Afin de qualifier la signification statistique de la ressemblance entre les scores de précision des différentes indexations effectuées, un test $t$ de Student [Ric06] est calculé avec pour paramètre $\alpha = 5\%$. Sur l'ensemble de la base de données de test des reprises, on peut conclure de ce test statistique les assertions suivantes :

- Il existe une différence significative entre les scores obtenus sans indexation et les scores obtenus avec une indexation arbitraire I1, I2 ou I3 ($p < 0.0005$ dans chaque cas) ;
- Il n'existe pas de différence significative entre les scores obtenus sans indexation et avec l'indexation par la répétition majeure.

**Efficacité calculatoire**

La durée moyenne des morceaux de la base de données complète $\mathcal{D}$ est de $n = 210$s. Les parties indexées par calcul de la répétition majeure ont une durée moyenne de $m = 69$s sur $\mathcal{D}$. Le gain moyen de l'indexation sur la longueur des séquences représentatives est d'environ $k = 3.26$[1]. Or, comme expliqué précédemment, la comparaison de deux séquences $u$ et $v$ requiert un nombre d'opérations de l'ordre de $|u| \times |v|$. Par conséquent, la tâche d'identification des reprises doit impliquer un

1. Pour rappel, cette valeur ne correspond pas au gain de la moyenne des durées mais à la moyenne des gains.

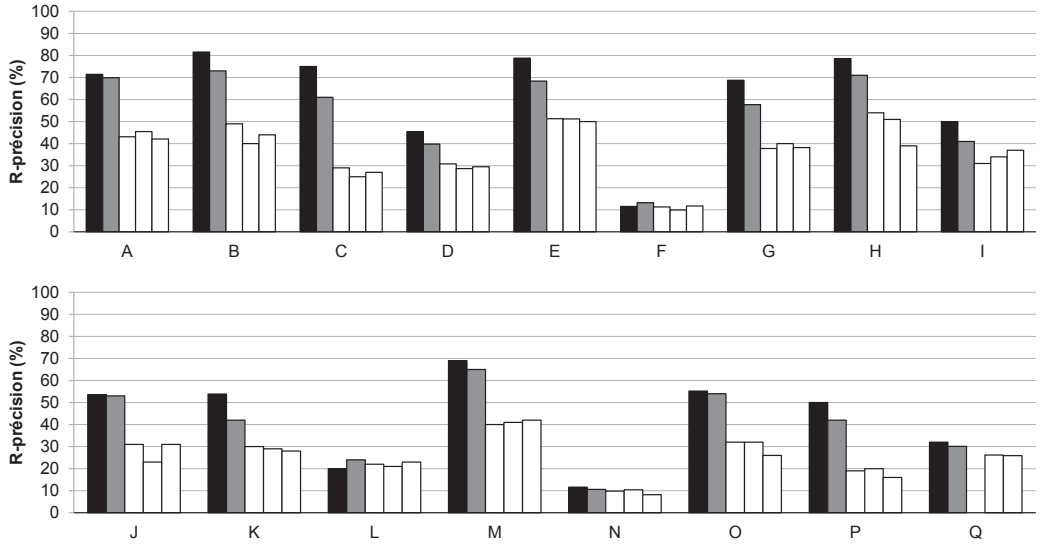


FIG. 4.17 – *Distribution des R-précisions avec différentes indexations calculées pour chaque classe de reprises. Les lettres en abscisse désignent la classe considérée (voir Tableau 3.2). Noir : aucune indexation, gris : indexation par la répétition majeure, blanc : de gauche à droite, indexations par les méthodes arbitraires I1, I2 et I3.*

gain de l'ordre de $k^2$, soit d'environ 8.58 [1] dans notre cas.

Les expériences présentées dans cette section nécessitent sur notre configuration matérielle [2] un temps de calcul de 5 heures et 45 minutes, en moyenne, pour chaque classe de reprises et sans indexation. Le temps nécessaire à la même évaluation avec des données indexées par la répétition majeure est de 41 minutes en moyenne pour chaque classe de reprises. En d'autres termes, grâce à l'étape d'indexation, le temps de calcul pratique est divisé par 8.4, soit d'un facteur proche de celui escompté. Les tâches d'indexation I1, I2 et I3 réduisant la quantité d'information dans le même rapport, leur utilisation dans un système de détection des reprises engendre un temps d'exécution identique.

La durée totale de la tâche d'identification des reprises inclut le temps nécessaire à l'indexation des données. Le calcul des répétitions majeures nécessite un temps de 23 minutes avec notre configuration. Par conséquent, la durée totale d'exécution du système est réduite d'un facteur 5.4 grâce à l'indexation proposée, avec une exécution par classe de 345 minutes sans indexation et de 64 minutes avec indexation par la répétition majeure.

### Bilan expérimental

L'évaluation présentée dans cette section souligne la pertinence de la répétition majeure pour indexer les séquences musicales d'un système d'identification des reprises. Une telle indexation permet de conserver une précision de la détection statistiquement proche de celle obtenue sans aucune indexation, avec une perte moyenne sur la précision globale du système de 3.5% pour les versions et 5.3% pour

1. Cette valeur correspondant à la moyenne des carrés des gains
2. Spécifications : Processeur Intel Core i5 750 2.66 GHz, Mémoire 4Go

les reprises. Cependant, cette technique permet de réduire sensiblement le temps de calcul requis, avec un facteur constaté de 8.4 fois plus rapide sans compter l'étape d'indexation, et 5.4 fois plus rapide en comptant cette étape.

## 4.3 Conclusion du chapitre

Dans ce chapitre, nous avons décrit l'étude de plusieurs structures répétitives simples dans un morceau de musique.

En choisissant un segment dans un morceau de musique, nous avons proposé un algorithme permettant d'identifier une répétition significative de ce segment. Cet algorithme est construit dans le cadre applicatif de la reconstruction de données audio manquantes, et évalué par des tests subjectifs. Les résultats mettent en valeur la précision et la pertinence musicale de la répétition identifiée.

En relâchant la contrainte de choix d'un segment pour l'étude de la répétitivité dans un morceau, nous avons défini la répétition majeure comme une répétition optimale d'un morceau sur un critère lié à la similarité et à la longueur des segments identifiés. Un algorithme d'extraction de cette répétition a été proposé, puis une comparaison avec des annotations manuelles de la structure répétitive a permis de montrer la concordance de la répétition majeure avec la structure perçue. Enfin, une utilisation concrète de cette répétition a été proposée dans un but d'identification automatique des reprises. L'évaluation de cette dernière a mis en avant la conservation d'une bonne précision du système d'identification grâce à l'indexation proposée, qui permet également d'accélérer la recherche des reprises dans une base de données.

Ce chapitre constitue une première approche de l'étude de l'inférence de la structuration répétitive des morceaux de musique. Ainsi, il convient de noter que de nombreuses autres structures répétitives simples et pourvues d'une signification musicale forte pourraient être étudiées. En particulier, on pourrait relâcher la contrainte de disjonction afin de prendre en compte un plus large ensemble de formes musicales.

Plus généralement, les solutions aux problèmes applicatifs posés dans ce chapitre sont susceptibles d'être améliorées en combinant un ensemble de quelques structures répétitives simples. Par exemple, dans le cas de la reconstruction audio, une première étude [BJM12] montre que la prise en compte de plusieurs répétitions combinées permet d'optimiser la similarité du segment reconstitué. Dans le cas de l'indexation par la répétition majeure, des tests préliminaires suggèrent également une plus grande précision du système d'identification des reprises en représentant chaque morceau par un ensemble de répétitions. Le problème de combiner plusieurs répétitions simples pour former une structuration plus complète des morceaux de musique constitue l'objet du Chapitre 5.

Chapitre 5

# Inférence de structures répétitives

La musique se compose de nombreuses répétitions à différentes échelles temporelles et sur différentes caractéristiques musicales. Dans ce chapitre, nous étudions le problème de l'identification d'un ensemble de répétitions qui structurent les morceaux de musique occidentale.

Si l'ensemble des travaux d'analyse de la structure en musique s'accordent sur son utilité dans le cadre de nombreux champs d'application [PMK10], la définition du problème d'inférence structurelle n'est pas universelle [PD09]. L'organisation de multiples répétitions dans un morceau de musique forme en effet une structuration complexe, dont l'analyse a été abordée avec différents objectifs [Pau10].

Dans la suite de ce chapitre, nous introduisons la problématique d'analyse de la structure musicale en Section 5.1 en passant en revue les travaux existants. Face à de nombreuses limitations et difficultés de représentation, nous présentons ensuite en Section 5.2 notre propre formalisation du problème sous la forme d'une structuration hiérarchique en nous basant sur le bien-fondé d'une telle description d'un point de vue musical. En Section 5.3, nous décrivons notre algorithme et les propriétés des structures répétitives qu'il identifie. Nous détaillons alors une évaluation des résultats d'inférence obtenus en Section 5.3.4 sur des données musicales.

## 5.1 Segmentation structurelle

De nombreuses approches pour l'inférence de la structuration répétitive de la musique depuis le signal audio ont été proposées ces dernières années sous le terme d'algorithmes de *segmentation structurelle*. La formalisation précise et l'identification de la structuration musicale reste un problème ouvert dont la résolution constitue un enjeu majeur de la recherche d'informations musicales [Mül11]. Dans cette section, nous décrivons les différentes approches existantes en détaillant leur représentation de la structuration musicale. Le Tableau 5.1 liste les principaux travaux antérieurs pour l'analyse de la structure répétitive à partir de données audio.

### 5.1.1 Matrice d'auto-distance

Foote introduit en 1999 [Foo99] un outil de représentation des répétitions au sein d'un morceau : la *matrice d'auto-distance*[1] pour l'analyse de la musique. Construite

1. L'expression *auto-similarité* est fréquemment employée pour désigner ce type de matrices. Cependant, ce terme fait référence à un concept mathématique bien connu et désigne un concept différent [Man83]. Pour éviter toute ambiguïté dans ce document, le terme d'*auto-distance* est préféré. Notez que même si la matrice d'auto-distance présente probablement certaines propriétés d'auto-similarité au sens de Mandelbrot, celles-ci n'ont pas été étudiées.

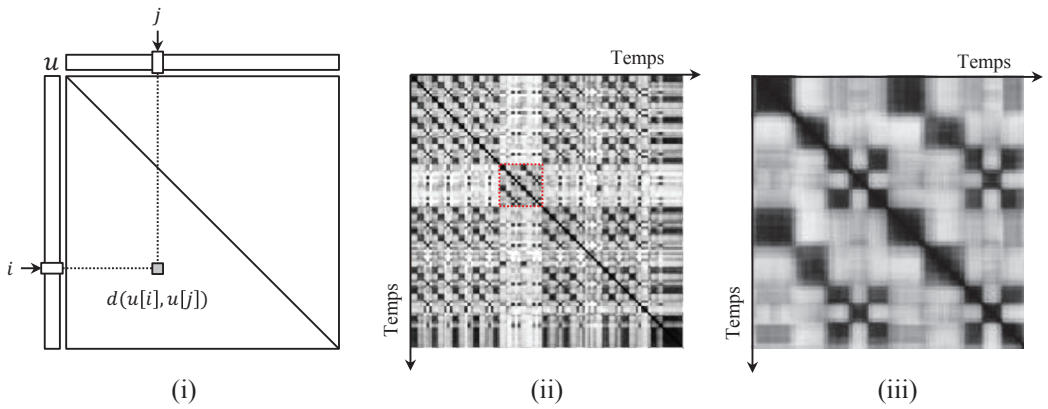


FIG. 5.1 – *Illustration de la matrice d'auto-distance. (i) : Principe d'obtention, (ii) : Exemple sur le morceau The Show Must Go On du groupe Queen, (iii) Détail d'un motif répété.*

à partir d'une séquence $u[1]u[2]\dots u[n]$, chaque coefficient $(i,j)$ de cette matrice représente la distance $d(u[i],u[j])$ entre les symboles $u[i]$ et $u[j]$ de cette séquence. Une telle représentation de l'auto-distance est utilisée dans de nombreux domaines applicatifs, et peut être vue comme un cas particulier des graphes de récurrence [EKR87]. Les plus anciennes références à l'auto-distance pour l'analyse de répétitions apparaissent dans la littérature bio-informatique [Lab], la matrice étant utilisée pour mettre en valeur différents aspects structurels des séquences moléculaires [ML81].

La Figure 5.1-(i) décrit le principe de calcul de cette matrice à partir d'une séquence : on calcule pour chaque couple de symboles de $u$ une distance, représentée sur une échelle de niveaux de gris dans la matrice. Celle-ci représente les distances les plus élevées par une couleur blanche, et les distances les plus faibles par une couleur noire. L'illustration 5.1-(ii) montre un exemple d'une telle matrice calculée à partir d'une représentation tonale du morceau *The show must go on* du groupe *Queen*. Les structures répétitives apparaissent sous la forme de motifs récurrents dans la matrice. La Figure 5.1-(iii) montre un détail de la figure précédente correspondant à un exemple de motif récurrent dans la matrice d'auto-distance.

Une grande partie des méthodes de la littérature (voir colonne 3 du Tableau 5.1) déduisent les structures répétitives de morceaux de musique en analysant cette matrice d'auto-distance. Deux motifs en particulier sont pertinents pour l'analyse de la répétition [Foo99]:

- D'une part, les *lignes* de coefficients faibles représentent la ressemblance entre deux facteurs ;
- D'autre part, les *blocs* de coefficients faibles représentent la ressemblance entre deux facteurs, chacun d'eux étant de plus composé de symboles se ressemblant entre eux.

La matrice d'auto-distance est parfois transformée en une structure de données équivalente, appelée *représentation temps-retard* [Got03, Got06, LWZ04a, BW05, LWZ04a, Ong07], qui correspond à une déformation de la matrice rendant plus facilement identifiables les lignes diagonales caractéristiques de celles-ci, mais moins facilement identifiables ses blocs [Pau10].

Les motifs de la matrice d'auto-distance peuvent être extraits par différentes

techniques. Cependant, puisque chaque coefficient de la matrice représente une valeur de similarité calculée entre deux trames de signal, seule une information correspondant au niveau de la structuration des trames peut directement être déduite de celle-ci. En d'autres termes, la matrice peut être vue comme une représentation *bas niveau* de l'information structurelle. Les méthodes d'identification de structures répétitives de plus haut niveau hiérarchique doivent alors utiliser des techniques adaptées à la détection de motifs approchés dans cette matrice afin d'assurer une robustesse aux variations musicales. Une approche consiste à utiliser des techniques de traitement d'images afin d'effacer les variations à court terme des éléments de la matrice, et ainsi d'adapter la détection au niveau de répétitions souhaité [MK07, CF04]. Des méthodes adaptées à la segmentation approchée d'images sont également utilisées afin d'identifier directement des motifs récurrents [AS02, ET07, Ong07, WCB10, KAS11]. Les structures répétitives peuvent en outre être repérées à l'aide d'heuristiques sur leur disposition dans la matrice [Mar06, Pee07, WH03] ou dans la représentation temps-retard équivalente [Got06, BW05, LWZ04a, Pee04]. En particulier, des relations de transitivité sur les répétitions permettent d'assurer une cohérence des éléments structurels identifiés [Pee07, Cha05]. Par ailleurs, plusieurs études identifient un ensemble de motifs récurrents en optimisant des algorithmes de parcours dans la matrice [DH02, MK07, Jen07] ou des critères de chevauchement des répétitions identifiées [MND09]. Enfin, une autre approche consiste à définir des variantes de la matrice d'auto-distance qui permettent notamment de préciser les frontières des répétitions identifiées [WCL09, TLX+09].

### 5.1.2 Détection des structures

De nombreuses méthodes sont proposées dans la littérature pour la détection de structures répétitives, cette matrice d'auto-distance étant directement utilisée dans une majorité d'entre elles. D'une manière générale, les méthodes existantes peuvent être regroupées en fonction des outils utilisés et de la logique de l'analyse qu'elles proposent. En particulier, une catégorisation proposée par Peeters [Pee04] divise ces méthodes en deux grandes approches : l'approche *état* et l'approche *séquence*.

#### 5.1.2.1 Approche *état*

L'approche *état* considère un morceau de musique comme généré par un automate d'états finis, chaque état produisant une section du signal [Pee04]. Les structures identifiées correspondent alors à des segments au contenu acoustique homogène, pouvant être vus comme des blocs dans la matrice d'auto-distance [PMK10]. Cette approche se décompose généralement en deux étapes complémentaires pour l'analyse des structures répétitives :

1. Une étape de *segmentation* du morceau, consacrée à l'identification des instants de contrastes importants dans le contenu musical ;
2. Une étape de *regroupement*, où tout ou partie des segments identifiés sont regroupés en classes *homogènes* en fonction de la distribution de leurs descripteurs.

L'étape de segmentation peut être réalisée sur un morceau de musique en analysant l'évolution de la *nouveauté audio* [Foo00] du morceau. Cette information est déduite de la matrice d'auto-distance $M$ en calculant pour chaque descripteur audio

| Contribution | Approche | Matrice d'auto-distance | Approche hiérarchique | Nombre de descripteurs | Méthode de détection des structures répétitives | MIREX |
|---|---|---|---|---|---|---|
| Abdallah *et al.* 2005 [ANS+05] | État | Non | Non | 1 | HMM [1] | |
| Aucouturier et Sandler 2002 [AS02] | Séquence | Oui | Oui | 1 | Image | |
| Aucouturier *et al.* 2005 [APS05] | État | Non | Non | 1 | HMM [1] | |
| Barrington *et al.* 2010 [BCL10] | État | Non | Non | 2 | DTM [2] | |
| Bartsch et Wakefield 2005 [BW05] | Séquence | Oui | Non | 1 | Heuristiques | |
| Chai 2003 [Cha03] | Séquence | Non | Non | 1 | Pattern matching | |
| Chen et Li 2011 [CL11] | État | Non | Non | 2 | Clustering, NMF [3] | 2011 |
| Cooper et Foote 2004 [CF04] | État | Oui | Non | 1 | Image | |
| Dannenberg et Hu 2002 [DH02] | Séquence | Non | Non | 1 | Algorithmique du texte | |
| Eronen et Tampere 2007 [ET07] | Séquence | Oui | Non | 2 | Image | |
| Foote 2000 [Foo00] | État | Oui | Non | 1 | Nouveauté | |
| Goto 2006 [Got06] | Séquence | Oui | Non | 1 | Heuristiques | |
| Jehan 2005 [Jeh05a] | État | Oui | Oui | 3 | Matrices hiérarchiques | |
| Jensen 2007 [Jen07] | État | Oui | Non | 3 | Chemin matrice | |
| Kaiser et Sikora 2010 [KS10] | État | Oui | Non | 2 | NMF [3] | |
| Levy et Sandler 2008 [LS08] | État | Non | Non | 1 | Clustering, HMM [1] | |
| Logan et Chu 2000 [LC00] | État | Non | Non | 1 | Clustering, HMM [1] | |
| Lu *et al.* 2004 [LWZ04a] | Séquence | Oui | Non | 1 | Image, heuristiques | |
| Maddage *et al.* 2006 [MXK06] | État | Non | Non | 1 | Heuristiques | |
| Marolt 2006 [Mar06] | Séquence | Oui | Non | 1 | Heuristiques | |
| Mauch *et al.* 2009 [MND09] | Séquence | Oui | Non | 1 | Heuristiques | 2010 |
| Muller et Kurth 2007 [MK07] | Séquence | Oui | Non | 1 | Image, chemin matrice, heuristiques | |
| Ong 2007 [Ong07] | Séquence | Oui | Non | 1 | Heuristiques | |
| Paulus et Klapuri 2006 [PK06] | Séquence | Non | Non | 2 | Fonction de coût | |
| Paulus et Klapuri 2009 [PK09] | Séquence/État | Oui | Non | 3 | Fonction de coût | 2009 |
| Peeters 2004 [Pee04] | État | Non | Non | 1 | HMM [1] | |
| Peeters 2007 [Pee07] | Séquence | Oui | Non | 3 | Heuristiques | 2009, 2010, 2011 |
| Rhodes et Casey 2007 [RC07] | Séquence | Non | Oui | 1 | Algorithmique du texte | |
| Sargent *et al.* 2010 [SBV10] | Séquence/État | Non | Non | 2 | Fonction de coût | 2010 |
| Sargent *et al.* 2011 [SBV+11] | Séquence/État | Non | Non | 2 | HMM [1], Fonction de coût | 2011 |
| Shiu *et al.* 2005 [SJK05] | Séquence | Oui | Non | 1 | Chemin matrice | |
| Tian *et al.* 2009 [TLX+09] | Séquence | Oui | Non | 1 | Heuristiques | |
| Wang *et al.* 2009 [WCL09] | Séquence | Oui | Non | 1 | Fonction de coût | |
| Wankhammer *et al.* 2010 [WCB10] | Séquence | Oui | Non | 2 | Image, heuristiques | |
| Wellhausen et Hoeynck 2003 [WH03] | Séquence | Oui | Non | 1 | Heuristiques | |
| Weiss et Bello 2010 [WB10] | État | Non | Non | 1 | NMF [3] | 2010 |
| Martin *et al.* 2010 [MHR+10] | Séquence | Non | Oui | 1 | Algorithmique du texte | 2010, 2011 |

TAB. 5.1 – *Vue d'ensemble des méthodes d'analyse de structures répétitives.* [1]*Modèle de Markov caché [Rab89].* [2]*Modèle de texture dynamique [BCL10].* [3]*Factorisation en matrices non-négatives [SL01].*

une mesure locale obtenue par corrélation le long de la diagonale avec une matrice $K$ de dimensions inférieures à celles de $M$ [Foo00]. $K$ agit alors comme un facteur de pondération et met en avant dans $M$ tout contraste important entre des blocs homogènes. Le choix d'une pondération imposée à $K$ dépend du type de segmentation recherchée, et peut par exemple être définie sur un modèle binaire ou Gaussien [Foo00]. La *courbe de nouveauté* correspond alors au résultat de la corrélation de $M$ avec $K$ en fonction du point d'application de $K$ sur la diagonale de $M$. Cette courbe est souvent utilisée comme une première étape de segmentation [Foo00, CF04, KS10, PK09, Rao04, Jen07, WCB10].

L'étape de *regroupement* consiste ensuite à identifier parmi les segments déterminés par la nouveauté audio des groupes homogènes. Cette étape peut être guidée par un modèle probabiliste permettant de classer les segments identifiés en fonction de la distribution de leurs descripteurs. En particulier, l'utilisation de modèles de Markov cachés [Rab89] permet de représenter chaque partie musicale caractéristique (couplet, refrain, *etc.*) par un état, le modèle produisant des segments annotés dont l'observation est liée à la distribution des descripteurs [GML03]. Après une phase d'apprentissage du modèle, l'algorithme de Viterbi [Vit67] permet de déterminer la distribution la plus probable des états pour chaque segment identifié [LC00, ANS+05, PBR02, SBV+11]. L'utilisation d'un modèle Gaussien pour les probabilités d'émission [Rab89] est souvent privilégiée pour représenter la distribution statistique des descripteurs audio [LC00, PBR02, APS05, SBV10, PQ11].

Une autre approche du problème consiste à employer des outils de segmentation d'images afin d'identifier des blocs homogènes à partir de la matrice d'auto-distance. Par exemple, Cooper et Foote [CF04] utilisent une technique de décomposition en valeurs singulières [Wei99] afin de rapprocher les motifs similaires de cette matrice. Plus récemment, Barrington *et al.* [BCL10] proposent des modèles de textures dynamiques [DCWS03] pour le signal audio permettant de représenter de manière précise les états possibles. La détermination des états produisant les sections de signal peut également être effectuée en utilisant une factorisation en matrices non-négatives [LS+99] appliquée sur la matrice d'auto-distance [KS10, CL11], ou appliquée directement sur la séquence de descripteurs via un modèle de factorisation probabiliste [WB10, WB11].

#### 5.1.2.2 Approche *séquence*

Dans l'approche *séquence*, une œuvre musicale est vue comme comprenant des séquences d'événements musicaux répétés. La répétition de ces séquences forme ainsi des lignes dans la matrice d'auto-distance.

De nombreuses méthodes de la littérature sont dédiées à l'identification de ces lignes. Face aux distorsions susceptibles d'apparaître entre plusieurs répétitions d'un motif, ces méthodes emploient différentes approches assimilables à un moyennage temporel permettant d'atténuer les variations à court terme [PMK10]. Cette opération peut ainsi être effectuée en appliquant un effet de filtrage, affectant toute la matrice d'auto-distance [Ong07, Got06, ET07, Pee07] ou uniquement ses diagonales [WH03, BW05, LWZ04a, MK07].

La détection des lignes de répétition dans la matrice d'auto-distance peut alors être effectuée par un seuillage de la matrice (ou de la représentation temps-retard équivalente) ainsi filtrée. Par exemple, dans sa méthode *RefraiD*, Goto [Got06]

sélectionne à partir d'un seuillage de la représentation temps-retard les sections les plus souvent répétées, puis utilise une série de mesures moyennes permettant d'identifier la partie la plus significative au sens de ces mesures, assimilée au refrain [Got06, Mar06, Ong07].

Les répétitions peuvent également être analysées à l'aide d'un algorithme de détection de lignes significatives approchées, afin de prendre en compte les nombreuses distorsions temporelles pouvant apparaître sur la matrice d'auto-distance [MK07, MND09, SJK05]. Par exemple, Müller et Kurth [MK07] proposent un algorithme glouton permettant de déduire, à partir de choix locaux sur les coefficients de la matrice d'auto-distance[1], le chemin le plus probable représentant une répétition importante.

L'analyse peut être menée de manière itérative en caractérisant la structure par étapes successives, précisant de plus en plus la détection des répétitions [Pee07, Jeh05a, Cha03]. Ainsi, Peeters [Pee07] propose le calcul de matrices d'auto-distance de plus en plus informatives en appliquant des relations de transitivité sur les répétitions apparaissant à plus de deux occurrences dans chaque morceau, et parvient ainsi à propager la similarité entre deux motifs à l'ensemble des motifs similaires pour faciliter l'extraction de lignes caractéristiques.

Plus généralement, une telle logique de transitivité est souvent mise en place dans les méthodes qui suivent une approche *séquence*. En effet, celles-ci identifient des paires de motifs similaires en présence de variations musicales ; en conséquence, si un motif est répété de manière approchée plus de deux fois dans un morceau, il est probable que les mesures de similarité entre les motifs deux-à-deux ne soient pas exactement identiques. Afin de conserver une description homogène des motifs structurels, la transitivité entre les similarités identifiées permet alors de regrouper toutes les mesures proches en attribuant aux motifs une même étiquette [Mül07]. Cette logique peut notamment être appliquée en ajoutant ou retirant de manière itérative des lignes caractéristiques de la matrice d'auto-distance [Got06, Mar06, Ong07, Pee07]. La logique de transitivité est parfois directement utilisée afin d'identifier un extrait caractéristique [ET07, WH03].

Certaines études correspondent à une approche *séquence* mais n'emploient pas la matrice d'auto-distance. En particulier, l'identification de répétitions peut être effectuée à l'aide d'outils d'algorithmique du texte [DH02, RC07, Smi10, AAF+09]. Par exemple, Rhodes et Casey [RC07] définissent plusieurs algorithmes du texte permettant d'identifier des facteurs similaires de mêmes longueurs, en autorisant un nombre limité de remplacements de symboles. Allali *et al.* [AAF+09] formalisent plusieurs problèmes liés à l'optimisation du recouvrement de la sous-séquence décrivant la structure répétitive musicale, et présentent les premières approches à leur résolution.

Il convient de noter que quelques études parviennent à combiner les approches *état* et *séquence* en optimisant des fonctions de score prenant en compte la répétitivité et l'homogénéité des motifs structurels identifiés [PK09, PLR08]. Dans leur approche, Paulus et Klapuri [PK09] définissent une mesure de coût de segmentation prenant en compte des propriétés musicales distinctes ; ainsi, leur méthode permet

1. L'algorithme glouton se différencie du raisonnement par programmation dynamique par la non optimalité du résultat. Le premier prend des décisions locales figées et non remises en question par la suite, tandis que le second optimise chaque décision sur l'ensemble des décisions passées.

d'optimiser à la fois la similarité entre tous les motifs associés à une même étiquette et la dissimilarité entre des groupes distincts [PK09, PMK10].

Pour plus de détails sur ces différentes méthodes d'inférence des structures répétitives, nous invitons le lecteur à se référer aux rapports d'état de l'art, tels que proposés par Dannenberg et Goto [DG09], Müller [Mül07] ou plus récemment Paulus *et al.* [PMK10].

### 5.1.3 Limitations et subjectivité

Les approches existantes pour l'analyse de la structure musicale souffrent de plusieurs limitations, liées à la fois aux outils utilisés et à la définition peu précise du problème.

La matrice d'auto-distance présente plusieurs limitations pour l'analyse des structures répétitives, comme le souligne notamment Chai [Cha05, p.40–42]. Un premier écueil apparaît dans le cas où une variation musicale modifie l'ensemble de la répétition musicale d'une section. Par exemple, dans le cas où cette matrice est calculée à partir d'une séquence de descripteurs tonaux, la présence d'une transposition entre deux sections répétées peut s'avérer problématique. Ainsi, le couplet représenté par une zone de pointillés rouges en Figure 5.1(ii) est transposé par rapport à ses autres occurrences dans le morceau. La structure répétitive *interne* de ce couplet est alors respectée, comme représenté par le motif de la Figure 5.1-(iii), récurrent dans la Figure 5.1-(ii). En revanche, la distance entre ce couplet et le reste du morceau est importante, comme matérialisé par des bandes horizontales et verticales blanches en Figure 5.1-(ii).

En outre, identifier les motifs répétés s'avère complexe en présence de forts changements de tempo entre plusieurs occurrences d'une répétition, comme souligné notamment par Müller [Mül07]. Dans le cas d'une telle variation, les lignes caractéristiques des répétitions dans la matrice d'auto-distance ne correspondent plus à des sous-diagonales et peuvent prendre une forme courbée qui témoigne d'un changement non linéaire du tempo entre deux motifs. Dans le cas de l'analyse de la musique classique occidentale, Müller [Mül07] affirme que les changements de tempo importants mêlés aux fortes variations dans l'instrumentation rendent la majorité des techniques basées sur l'auto-distance inefficaces pour identifier les structures répétitives.

Les approches *état* basées sur la nouveauté du signal audio présentent également une limitation due à la forte variabilité des frontières structurelles perçues. En effet, comme le soulignent notamment Bruderer *et al.* [BMK06], la définition des frontières entre motifs structurels est associée à des éléments subjectifs multiples. Les auteurs mettent en avant par une série d'expériences perceptives la présence de frontières communes, identifiées par tous les sujets, mais liées à des critères musicaux variables. Ainsi, la tâche d'identifier des points de nouveauté depuis le signal audio est complexe, et source de nombreuses erreurs pour l'analyse des structures répétitives.

Une limitation majeure de l'analyse de la structuration de morceaux de musique, notamment mise en valeur dans des études méthodologiques récentes [BLBSV10, BDSV11, PD09, PK12], est liée au manque d'une formalisation unique du problème étudié. L'utilité d'un effort commun vers une définition plus formelle du problème

de la structuration musicale est communément admise dans la littérature récente. Néanmoins, l'utilisation des différentes méthodologies proposées pour l'analyse et l'annotation de structures musicales [BLBSV10, BDSV11, PD09, SBF+11] reste peu employée à l'heure actuelle.

Bimbot *et al.* [BLBSV10] soulignent ainsi que l'absence d'une définition formelle de la structure musicale soulève un problème méthodologique pour son analyse, qui limite notamment les possibilités d'évaluation de telles méthodes. Les auteurs proposent alors une méthodologie présentant les concepts d'un découpage structurel perceptif en "blocs", caractérisés en particulier par des propriétés d'*autonomie*, de *comparaison* avec les autres blocs ou encore de *régularité* tout au long du morceau [BLBSV10]. Pour faciliter ce découpage, les auteurs [BDSV11] définissent ensuite plusieurs niveaux de décomposition à examiner d'une manière conjointe afin de déterminer le meilleur compromis d'annotation structurelle.

L'annotation structurelle sur différents niveaux est également exposée comme un procédé plus rigoureux par Peeters et Deruty [PD09]. Ces derniers proposent ainsi une représentation des structures répétitives selon plusieurs dimensions traduisant des points de vue disjoints, tels que la *similarité acoustique*, le *rôle musical* ou encore le *rôle instrumental*. Si l'espace de description de la structure ainsi recherchée est plus complexe que celle correspondant au problème de segmentation structurelle, son obtention est fondée sur des critères de comparaisons subjectives simples et plus aisément reproductibles entre différents sujets.

Ces derniers travaux suggèrent qu'une représentation de la structure musicale sur plusieurs niveaux de description permet de modéliser plus précisément les structures répétitives.

### 5.1.4 Vers une approche hiérarchique

La détection d'un ensemble de répétitions perçues à différents niveaux de description est considérée comme un enjeu important de l'analyse de la structuration musicale [Jeh05a, Pau10, Cha05]. En particulier, Müller [Mül07] suggère comme perspective majeure de son étude la description des motifs imbriqués sur plusieurs niveaux. Il cite alors comme problème non résolu « *l'intégration des similarités à toutes les résolutions temporelles dans un unique modèle hiérarchique pour décrire au mieux la structure musicale* ». Plusieurs autres études récentes citent également la prise en compte d'une décomposition structurelle *hiérarchique* comme perspective majeure (voir notamment [Pau10, KS10, WB10, AS02, MK07]).

Bien que cet aspect soit considéré comme une amélioration significative, il n'existe que quelques approches proposant l'identification d'une hiérarchie des répétitions [PMK10]. Jehan [Jeh05b, Jeh05a] définit une construction itérative de matrices d'auto-distance en déduisant la matrice d'un niveau donné à partir d'un calcul par programmation dynamique dans les motifs de la matrice du niveau précédent. Les frontières de chaque matrice étant obtenues par un descripteur différent, cette technique permet alors d'identifier la structure répétitive sur différents critères musicaux simultanément. Chai [Cha05] définit une procédure permettant de déduire un ensemble de hiérarchies structurelles possibles à partir d'une description séquentielle. Malgré de possibles conflits de cette méthode pour l'assignation d'étiquettes hiérarchiques (notamment des problèmes de recouvrement), l'étude met alors en avant la pertinence d'un modèle arborescent pour une évaluation des répé-

titions analysées, plus fidèle à la représentation de structurations musicales qu'en considérant un modèle séquentiel. Rhodes et Casey [RC07] proposent un ensemble d'algorithmes de séquences musicales permettant d'identifier de manière itérative une construction hiérarchique des répétitions prépondérantes. Leur méthode, évaluée sur des données symboliques, est basée sur l'inférence approchée de motifs de même taille en utilisant une distance de Hamming [Ham50]. Enfin, quelques publications proposent une visualisation de la hiérarchie structurelle, sous la forme d'un espace instant/durée [MG12] ou d'un diagramme de convergence radiale [SHOB11].

Ces différentes approches permettent d'identifier une représentation de la structuration de la musique sur différentes échelles. Cependant, aucune de ces études n'explicite la structuration et le modèle hiérarchique sous-jacent, rendant leur évaluation complexe.

Dans la suite de ce chapitre, nous introduisons un modèle hiérarchique, définissons formellement le problème d'identification de structures répétitives par un critère d'optimisation dans la hiérarchie, et nous proposons un algorithme d'inférence répondant au problème.

## 5.2 Modèle hiérarchique des répétitions

Une perspective majeure des travaux existants en analyse de structures répétitives dans la musique est la prise en compte d'une hiérarchie de répétitions, comme expliqué en section précédente. Dans cette section, nous examinons les caractéristiques de la structuration hiérarchique de la musique à partir d'études existantes avant d'introduire une formalisation de ce problème.

### 5.2.1 Caractérisation musicale

Considérer un modèle hiérarchique présente une grande pertinence pour décrire l'information musicale.

Les répétitions d'un morceau de musique apparaissent d'abord à de nombreux niveaux temporels de composition, formant une structuration hiérarchique. Par exemple, un ensemble de notes de musique disposées selon des motifs plus ou moins complexes forme des accords et progressions harmoniques, eux-mêmes décrivant des phrases musicales sur de plus longues périodes, ces dernières pouvant être à nouveau regroupées en sections caractéristiques. Des contrastes et ressemblances apparaissent alors à tous les niveaux de description, la musique étant construite par les relations entre ces niveaux [Esc88]. Par exemple, le *Boléro* de Ravel est fait de neuf répétitions du même thème, chaque thème étant lui-même composé de neuf phrases de seize mesures chacune. Cet agencement particulier témoigne d'une volonté du compositeur, comme indiqué en Section 1.2.1.

La structuration hiérarchique n'est pas liée uniquement à la *composition* musicale, mais également à sa *perception* : les répétitions musicales sont perçues d'une manière arborescente. Cette nature de la musique ne dépend pas de considérations esthétiques ; au contraire, elle fonctionne dans tous les styles de musique ou même toute succession de signaux car elle est inhérente à la façon dont nous percevons le son [Eri75, p.80–82]. Comme souligné par Lerdahl et Jackendoff [LJ96], l'imbrication hiérarchique de motifs musicaux dans notre perception de la musique est alors

essentielle :

> *« La caractéristique la plus fondamentale des regroupements musicaux est qu'ils sont perçus d'une manière hiérarchique. Un motif est perçu comme partie d'un thème, un thème comme partie d'un groupe de thèmes, et une section comme partie d'une œuvre. »*

Dans leur étude, Lerdahl et Jackendoff [LJ96, p.16] affirment en outre qu'un aspect important des répétitions musicales est leur propriété de *non chevauchement*[1]. Chaque niveau de la hiérarchie répétitive perçue pour un morceau correspond ainsi à des extraits disjoints. La théorie des auteurs est applicable à un ensemble de styles musicaux aussi large que possible. Néanmoins, il convient de noter que cette considération peut s'avérer non pertinente pour certaines formes musicales particulières ; par exemple, la forme fugue [Esc88] est notamment composée de thèmes se chevauchant, et ne semble pas correspondre pas à ce modèle.

Une autre propriété importante de la hiérarchie musicale est son aspect dit *récursif*, défini comme le fait que chaque niveau hiérarchique « *peut être élaboré par les mêmes règles* » [LJ96, p.14]. Par exemple, la contrainte de non chevauchement est identique quel que soit le niveau considéré. Enfin, Lerdahl et Jackendoff [LJ96, p.16] définissent la structure hiérarchique comme constituée de groupes d'éléments *contigüs*, indiquant ainsi qu'à tout motif identifié dans une séquence $u$ doit correspondre un facteur de $u$ (et non une sous-séquence).

Les auteurs [LJ96, p.17] résument la structure musicale de regroupement de la manière suivante :

> *« La structure de regroupement est hiérarchique d'une façon non chevauchante, elle est récursive et chaque groupe doit être composé d'éléments contigüs. »*

Cette assertion est utilisée dans la suite comme axiome pour la modélisation et l'inférence des structures musicales répétitives.

### 5.2.2 Modélisation hiérarchique

Nous introduisons dans cette section un formalisme de l'identification de structures répétitives. Celui-ci expose d'abord une formalisation de la segmentation structurelle avant de présenter notre modèle de structuration des morceaux de musique. Celui-ci prend en compte les principes d'organisation hiérarchique justifiés dans la section précédente. Les notations introduites dans cette section sont notamment adaptées à notre problème notamment à partir des travaux d'Allali *et al.* [AAF+09].

#### 5.2.2.1 Définitions

Dans la suite, on considère deux alphabets distincts notés $\Sigma$ et $\Lambda$. Le premier désigne l'ensemble des symboles sur lesquels sont définies les séquences comparées.

1. Les auteurs parlent plus précisément d'une « condition de *disjonction*, avec pour exception l'inclusion d'un motif dans un autre » ; cela correspond ainsi à notre définition du *non chevauchement*.

Le second est utilisé pour nommer les répétitions identifiées en leur donnant une *étiquette*, ces notions étant définies formellement dans la suite.

**Définition 12 (Motif, occurrence, étiquette)** *Soient $\Sigma$ et $\Lambda$ deux alphabets, et soit $u$ une séquence de $n$ symboles sur $\Sigma$. On appelle* motif *$m$ de $u$ tout triplet de la forme $(b,e,\alpha) \in [\![1,n]\!] \times [\![1,n]\!] \times \Lambda$ tel que $b \leq e \leq n$. La séquence $u[b \dots e]$ est appelée l'*occurrence *de $m$ dans $u$ ; le symbole $\alpha$ est appelé* étiquette *de $m$. Par commodité, la taille d'un motif $m$ de $u$ est notée $|m| = e - b + 1$ et correspond à la taille de son occurrence dans $u$. L'ensemble des motifs possibles dans $u$ est noté $\mathcal{M}(u) \subset [\![1,n]\!] \times [\![1,n]\!] \times \Lambda$.*

En d'autres termes, un *motif* d'une séquence $u$ contient les positions de début et de fin d'un facteur de $u$ ainsi qu'une étiquette associée à celui-ci.

On introduit un ensemble $\Theta \subset \Lambda$ d'étiquettes particulières. Chaque étiquette $\Theta_i \in \Theta$ est appelée *étiquette nulle*. La notion d'étiquettes est utilisée afin de modéliser un ensemble de *classes d'équivalence*, qui caractérisent la similarité ou la dissimilarité entre motifs. Ainsi, deux motifs de même étiquette non nulle sont dits *similaires* et font partie de la même classe d'équivalence, tandis que deux motifs d'étiquettes différentes et non nulles sont dits *dissimilaires* et ne font pas partie de la même classe d'équivalence. Une étiquette nulle ne caractérise pas de similarité entre motifs. En particulier, deux motifs de même étiquette nulle ne font pas nécessairement partie de la même classe d'équivalence.

Pour une séquence $u$ de taille $n$, le motif $(1,n,\Theta_0)$ est appelé *motif initial* de $u$.

**Définition 13 (Sous-séquence de motifs)** *Soit $\mathcal{S} = \{(b_i,e_i,\alpha_i) | 1 \leq i \leq k\}$ un ensemble de motifs d'une séquence $u$. On dit que $\mathcal{S}$ est une sous-séquence de $u$ si et seulement si les occurrences de ses motifs sont disjointes dans $u$ :*

$$\forall (i,j) \in [\![1,k]\!], b_i < b_j \Leftrightarrow e_i < b_j$$

**Définition 14 (Taille de sous-séquence de motifs)** *Soit $\mathcal{S} = \{(b_i,e_i,\alpha_i) | 1 \leq i \leq k\}$ une sous-séquence de motifs d'une séquence $u$. La taille de $\mathcal{S}$, notée $|\mathcal{S}|$, correspond à la somme des tailles des occurrences de ses motifs :*

$$|\mathcal{S}| = \sum_{i=1}^{k} e_i - b_i + 1.$$

En fonction des étiquettes de ses motifs, une sous-séquence de motifs peut être étiquetée *complètement* ou *partiellement* :

**Définition 15 (Étiquetages complet et partiel)** *Soit $\mathcal{S}$ une sous-séquence de motifs d'une séquence $u$. $\mathcal{S}$ est dite complètement étiquetée si elle ne contient aucun motif d'étiquette nulle. À l'inverse, $\mathcal{S}$ est dite partiellement étiquetée si elle contient au moins un motif d'étiquette nulle.*

**Définition 16 (Recouvrement)** *Soit $\mathcal{S} = \{(b_i,e_i,\alpha_i) | 1 \leq i \leq k\}$ une sous-séquence de motifs d'une séquence $u$. On dit que $\mathcal{S}$ recouvre $u$ si $u$ correspond à la concaténation des occurrences de tous les motifs de $\mathcal{S}$. Formellement, $\mathcal{S}$ recouvre $u$ si*

$$\bigcup_{i=1}^{k} [b_i,e_i] = [1,n].$$

La notion de *segmentation structurelle*, telle qu'elle est analysée dans les méthodes existantes, peut alors être formalisée de la manière suivante :

**Définition 17 (Segmentation structurelle)** *Soit $u$ une séquence. Toute sous-séquence de motifs $\mathcal{S}$ de $u$ complètement étiquetée et recouvrant $u$ est appelée segmentation structurelle.*

Notre définition de la structuration répétitive correspond à la généralisation de la segmentation structurelle sous la forme d'une *hiérarchie*.

**Définition 18 (Hiérarchie)** *Soit $u$ une séquence de longueur $n$. On appelle hiérarchie de $u$ tout ensemble non vide de motifs aux occurrences non chevauchantes dans $u$. Par convention, on impose que toute hiérarchie de $u$ contienne le motif initial $(1,n,\Theta_0)$.*

**Définition 19 (Domination)** *Soient $m_1 = (b_1,e_1,\alpha_1)$ et $m_2 = (b_2,e_2,\alpha_2)$ deux motifs distincts. On dit que $m_1$ domine $m_2$ si l'occurrence de $m_2$ dans $u$ est un facteur propre de l'occurrence de $m_1$ dans $u$, c'est-à-dire si $b_1 < b_2 \leq e_2 < e_1$. Par convention, un motif n'est pas dominé par lui-même.*

Pour une séquence $u$, on peut remarquer que la domination est une relation d'ordre partiel sur l'ensemble $\mathcal{M}(u)$ des motifs de $u$.

Chaque motif d'une hiérarchie est ainsi dominé par un certain nombre d'autres motifs de la même hiérarchie.

**Définition 20 (Niveau)** *Soient $\mathcal{H}$ une hiérarchie et $i$ un entier. On appelle* niveau *hiérarchique $i$ de $\mathcal{H}$ l'ensemble des motifs de $\mathcal{H}$ dominés par exactement $i$ autres motifs dans $\mathcal{H}$.*

Avec une représentation arborescente, un *niveau* peut être vu comme une *hauteur* spécifique dans l'arbre. La notion de domination peut être étendue à un niveau entier de la manière suivante :

**Définition 21 (Hiérarchie dominante)** *Soient $\mathcal{H}$ une hiérarchie et $i$ un entier. On appelle hiérarchie dominante du niveau $i$ de $\mathcal{H}$ la hiérarchie composée de l'ensemble des niveaux de $\mathcal{H}$ strictement inférieurs à $i$.*

Avec une représentation arborescente, la hiérarchie dominante d'un niveau $i$ correspond au sous-arbre de profondeur $i-1$.

Chaque niveau d'une hiérarchie possède la propriété suivante :

**Proposition 10 (Niveau et sous-séquence de motifs)** *Soit $u$ une séquence, et soit $\mathcal{H}$ une hiérarchie de $u$. Tout niveau hiérarchique de $\mathcal{H}$ est une sous-séquence de motifs de $u$.*

**Preuve 2** *Soient $\mathcal{H}$ une hiérarchie et $\mathcal{N}$ un niveau de $\mathcal{H}$. Par définition, tous les éléments de $\mathcal{H}$ sont non chevauchants. On raisonne par l'absurde : supposons que $\mathcal{N}$ ne soit pas une sous-séquence de motifs de $u$. Il existe donc au moins deux éléments de $\mathcal{N}$ non disjoints dans $u$. Puisque ces deux motifs sont non disjoints mais non chevauchants, l'un, noté $m_1$, domine l'autre, noté $m_2$. Par conséquent, $m_1$ est de niveau hiérarchique supérieur à $m_2$. Cette assertion entre en contradiction avec l'appartenance des deux motifs au même niveau $\mathcal{N}$, ce qui prouve la proposition.*

Cette propriété permet de passer aisément d'une représentation hiérarchique vers une représentation sous forme de segmentation structurelle, notamment à des fins d'évaluation, comme expliqué ci-après.

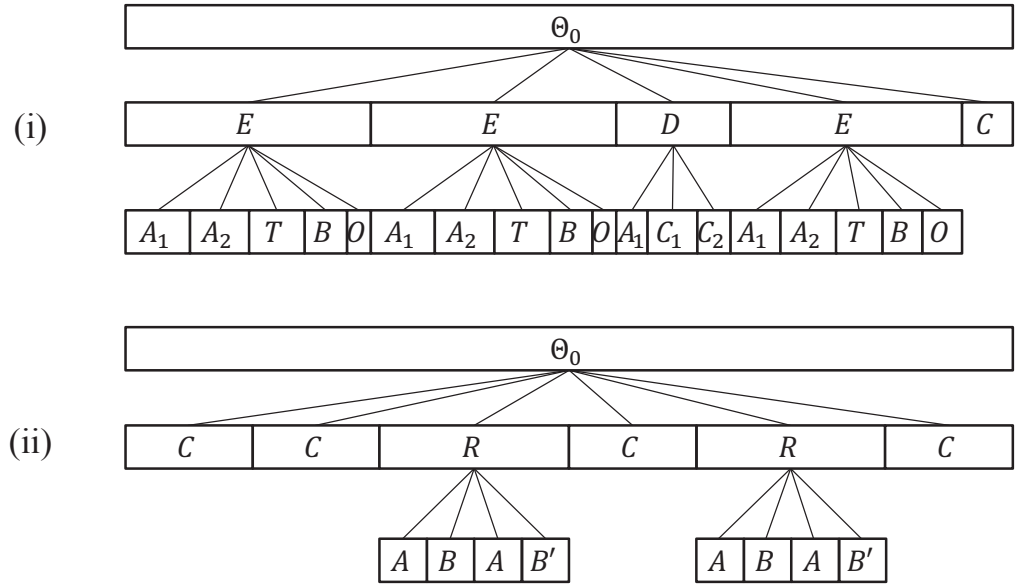


FIG. 5.2 – *Exemples de représentations arborescentes de structures répétitives : (i) Forme sonate schématisée ; (ii) :* Yesterday *du groupe The Beatles, d'après [Cha05, p.58].*

#### 5.2.2.2 Représentation et exemples

Pour afficher la nature hiérarchique et non chevauchante des motifs, une hiérarchie peut être représentée sous la forme d'une structure arborescente, définie de la manière suivante :

- La racine représente le motif initial
- Chaque nœud représente un motif de $\mathcal{H}$ ;
- Chaque arc représente une relation de domination entre deux motifs : $m_1$ est père de $m_2$ si et seulement si $m_1$ domine $m_2$.

Sur la Figure 5.2, par commodité de lecture, chaque nœud de l'arbre est représenté sous la forme d'un rectangle dont la largeur est une proportion de la taille de l'occurrence du motif. De cette manière, l'échelle temporelle des motifs représentés est facilement lisible sur l'arbre.

La Figure 5.2-(i) schématise l'organisation générale des répétitions d'un morceau suivant une structure dite *forme sonate*, très courante en musique classique [BS09]. Elle est organisée de manière hiérarchique, deux niveaux caractéristiques étant présentés ici. Le niveau 1 de la hiérarchie correspond aux quatre sections principales de la forme sonate : l'*exposition* $E$, jouée deux fois, le *développement* $D$, puis la *réexposition* correspondant à une répétition de $E$ (éventuellement avec variations) suivie de la *coda* $C$ qui conclut le morceau [BS09]. Le niveau 2 indique la présence des *thèmes* $A_1$, $A_2$, $B$ et $C$ apparaissant à plusieurs reprises et séparés par des *transitions* $T$ et des *résolutions* $O$.

La Figure 5.2-(ii) schématise la hiérarchie des répétitions dans un morceau de musique populaire, *Yesterday* du groupe *The Beatles*, d'après Chai [Cha05, p.58]. Le premier niveau correspond au niveau structurel des couplets (C) et refrains (R), tandis que le second décrit les phrases musicales des refrains.

Dans ces deux exemples, la hiérarchie structurelle est perçue par tout auditeur et correspond à une volonté de composition. Dans le cas de la forme sonate, chacun des deux niveaux a un sens musical fort, et décrire la structure du morceau sur un unique niveau semble être une simplification conséquente d'un problème plus général.

Ces exemples suggèrent que la définition d'un algorithme d'inférence de répétitions sous une forme hiérarchique peut permettre d'identifier une structure riche en informations sur la composition musicale d'un morceau.

## 5.3 Inférence hiérarchique de répétitions

Dans cette section, nous proposons un algorithme d'inférence de la structuration répétitive d'un morceau de musique. Cet algorithme est construit d'une manière hiérarchique, et correspond au modèle exposé dans la section précédente.

### 5.3.1 Préliminaires

Avant de décrire l'algorithme d'inférence, nous introduisons plusieurs critères sur la structuration des séquences.

En premier lieu, on introduit un critère de décision sur la proximité entre scores de similarité locale. Celui-ci est utile à l'identification de multiples répétitions d'un même motif.

**Critère 1 ($\Delta$-Proximité de scores)** *Soient $s_1$ et $s_2$ deux scores de similarité, et $\Delta$ un seuil décimal. $s_1$ est dit $\Delta$-proche de $s_2$ si et seulement si $|s_1 - s_2| \leq \Delta$.*

$\Delta$ correspond à un seuil de similarité entre motifs qui permet de regrouper ces derniers en classes d'équivalence. Plus précisément, deux motifs qui sont $\Delta$-proches sont considérés comme appartenant à la même classe d'équivalence.

En second lieu, on introduit un critère de décision sur la taille des motifs identifiés. Celui-ci est utile à la terminaison de l'algorithme présenté ci-après.

**Critère 2 ($\Gamma$-Insuffisance de taille)** *Soient $u$ une séquence, $m$ un motif de $u$ et $\Gamma$ un seuil décimal. On dit que $m$ est de taille $\Gamma$-insuffisante si et seulement si $|m - u| > \Gamma$. À l'inverse, on dit que $m$ est de taille $\Gamma$-suffisante si et seulement si $|m - u| \leq \Gamma$*

### 5.3.2 Problème

Cette section formalise le problème de structuration hiérarchique. On commence par définir la *sous-structure optimale* de la manière suivante :

**Définition 22 (Sous-structure optimale)** *Soient $u$ une séquence et $\mathcal{S}$ une sous-séquence de motifs de $u$. On note $E$ l'ensemble des facteurs de $u$ non chevauchants dans $u$ avec les occurrences des motifs de $\mathcal{S}$. On appelle sous-structure optimale de $\mathcal{S}$, et on note $R(\mathcal{S})$, la paire de motifs $\{(b_1,e_1,\alpha_1),(b_2,e_2,\alpha_2)\} \in \mathcal{M}(u)^2$ vérifiant :*

$$\begin{cases} (b_1,e_1,\alpha_1) \notin \mathcal{S} \\ (b_2,e_2,\alpha_2) \notin \mathcal{S} \\ s(u[b_1 \dots e_1],u[b_2 \dots e_2]) = \max\limits_{(v_1,v_2)\in E^2} s(v_1,v_2) \\ \alpha_1 = \alpha_2 \end{cases} .$$

En d'autres termes, $R(\mathcal{S})$ correspond à un ensemble de deux motifs de même étiquette, dont la similarité entre les occurrences est maximale, et qui ne chevauchent pas dans $u$ les motifs de la sous-séquence de motifs $\mathcal{S}$.

On définit alors de manière plus générale un ensemble identifiant tous les motifs de la même classe d'équivalence, dans une *sous-structure répétitive* $\Delta$*-optimale* de la manière suivante :

**Définition 23 (Sous-structure $k$,$\Delta$-répétitive)** *Soient $u$ une séquence, $\mathcal{S}$ une sous-séquence de motifs de $u$, $\Delta \geq 0$ un seuil décimal et $k$ un entier positif. On appelle sous-structure $k$,$\Delta$-répétitive de $\mathcal{S}$, et on note $R^k_\Delta(\mathcal{S})$ toute sous-séquence de $k$ motifs $\{(b_i,e_i,\alpha_i) \in \mathcal{M}(u) | 1 \leq i \leq k\}$ vérifiant :*

$$\exists (i,j) \in [\![1,k]\!]^2 : \begin{cases} \{(b_i,e_i,\alpha_i),(b_j,e_j,\alpha_j)\} = R(\mathcal{S}) \\ \forall (i',j') \in [\![1,k]\!]^2, s(u[b_{i'} \dots e_{i'}],u[b_{j'} \dots e_{j'}]) \\ \quad \textit{est } \Delta\textit{-proche de } s(u[b_i \dots e_i],u[b_j \dots e_j]) \\ \forall (i',j') \in [\![1,k]\!]^2, \alpha_i = \alpha_j = \alpha_1 \end{cases}.$$

En d'autres termes, $R^k_\Delta(\mathcal{S})$ correspond à l'ensemble incluant la sous-structure optimale de la sous-séquence de motifs $\mathcal{S}$ ainsi qu'un ou plusieurs motifs de scores de similarité $\Delta$-proches.

**Définition 24 (Sous-structure $\Delta$-répétitive optimale)** *Soient $u$ une séquence, $\mathcal{S}$ une sous-séquence de motifs de $u$, $\Delta \geq 0$ un seuil décimal et $k$ un entier positif. On appelle sous-structure $\Delta$-répétitive optimale, et on note $R_\Delta(\mathcal{S})$, une sous-structure $k$,$\Delta$-répétitive dont la taille est maximale pour toutes les valeurs de $k$ possibles. Formellement, $R_\Delta(\mathcal{S})$ vérifie :*

$$|R_\Delta(\mathcal{S})| = \max_{k \in \mathbb{N}^+} |R^k_\Delta(\mathcal{S})|.$$

Le problème de structuration hiérarchique est alors posé de la manière suivante :

**Problème 4 (Structuration hiérarchique)** *Soient $u$ une séquence, $\Delta$ et $\Gamma$ deux seuils décimaux et $N$ un entier strictement positif. Identifier une hiérarchie $\mathcal{H}$ qui respecte les critères suivants :*

*(i) Tous les motifs de $\mathcal{H}$ sont de taille $\Gamma$-suffisante ;*

*(ii) Le nombre de niveaux de $\mathcal{H}$ est $N$ ;*

*(iii) Chaque niveau $1 \leq i \leq N$ de $\mathcal{H}$ contient la sous-structure $\Delta$-répétitive optimale du niveau $i-1$.*

Chaque niveau d'une hiérarchie-solution du problème est donc caractérisé par un critère d'optimalité sur un ensemble de répétitions qu'il contient.

### 5.3.3 Algorithme

Soit $u$ une séquence de symboles. L'objectif de l'algorithme d'inférence de structures répétitives est d'identifier une hiérarchie de répétitions dans $u$ solution du Problème 4. Cette identification est effectuée par itérations successives, chacune de ces itérations introduisant une nouvelle répétition caractéristique. On pose $\mathcal{H}$ une hiérarchie de $u$. Notre algorithme est décrit sous la forme d'une récurrence sur le niveau $i$ des motifs de la hiérarchie $\mathcal{H}$.

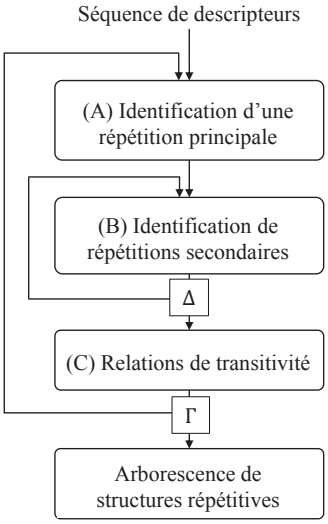


FIG. 5.3 – *Vue d'ensemble de l'algorithme d'inférence de structures répétitives. A, B et C désignent les différentes étapes de l'algorithme. Les flèches remontantes représentent les processus de bouclage, $\Delta$ et $\Gamma$ désignant les paramètres sur lesquels sont basées les décisions de fin de boucle.*

#### 5.3.3.1 Récurrence

**Définition 25** *Soient $\mathcal{H}$ une hiérarchie et $i$ un entier positif. Pour chaque niveau $i$ de $\mathcal{H}$, on définit $\overline{\mathcal{H}_i}$ comme l'ensemble de $u$ contenant les motifs de $\mathcal{H}$ munis de l'étiquette nulle $\Theta_i$ :*

$$\overline{\mathcal{H}_i} = \{(b,e,\alpha) \in \mathcal{H} | \alpha = \Theta_i\}.$$

Il convient de noter que puisque pour un $i$ donné, $\overline{\mathcal{H}_i}$ contient les motifs non étiquetés du niveau $i$, $\overline{\mathcal{H}_i}$ est une sous-séquence de motifs de $u$.

À l'initialisation de la récurrence, aucun facteur n'est étiqueté, d'où $\mathcal{H} = \overline{\mathcal{H}_0} = \{(1,n,\Theta_0)\}$.

Le calcul de l'étape $i$ de la récurrence est divisé en trois étapes complémentaires. La Figure 5.3 schématise le déroulement de ces différentes étapes, décrites en détail ci-dessous.

Dans la suite de cette section, on pose $i$ l'étape de la récurrence en cours, et $\mathcal{H}_i$ l'ensemble des motifs de $\mathcal{H}$ munis d'une étiquette nulle $\Theta_i$.

**Étape A : Calcul d'une sous-structure optimale**

Cette étape identifie une paire de motifs et leur attribue une nouvelle étiquette.

**Calcul A**

| | |
|---|---|
| $\forall(b_1,e_1,\alpha_1) \in \overline{\mathcal{H}_i}$, calculer $s^{\circledast}(u[b_1 \dots e_1])$ | Tâche (i) |
| $\forall((b_1,e_1,\alpha_1),(b_2,e_2,\alpha_2)) \in \overline{\mathcal{H}_i} \times \mathcal{H}$, calculer $s^*(u[b_1 \dots e_1],u[b_2 \dots e_2])$ | Tâche (ii) |

Chacune des deux tâches fournit un ensemble de scores de similarité locale ; à la fin de cette étape, on identifie le score de similarité maximal $s_{\max}$ de cet ensemble.

Formellement,

$$s_{\max} = \max(\max_{(b_1,e_1,\alpha_1)\in\overline{\mathcal{H}_i}} s^{\circledast}(u[b_1 \dots e_1]), \max_{((b_1,e_1,\alpha_1),(b_2,e_2,\alpha_2))\in\overline{\mathcal{H}_i}\times\mathcal{H}} s^*(u[b_1 \dots e_1], u[b_2 \dots e_2])) \quad (35)$$

On note $(b_1^*,e_1^*)$ et $(b_2^*,e_2^*)$ les indices dans $u$ des deux facteurs effectivement alignés $v_1^* = u[b_1^* \dots e_1^*]$ et $v_2^* = u[b_2^* \dots e_2^*]$.

L'algorithme correspond alors à la suite d'opérations suivante :

- On choisit une nouvelle étiquette, notée $\alpha$ ;
- Les deux motifs $m_1 = (b_1^*,e_1^*,\alpha)$ et $m_2 = (b_2^*,e_2^*,\alpha)$ sont ajoutés à $\mathcal{H}$ ;
- On retire de $\mathcal{H}$ les motifs d'étiquettes $\Theta_i$ ;
- On ajoute à $\mathcal{H}$ le nombre minimum de motifs d'étiquettes $\Theta_i$ afin que le niveau $i$ de $\mathcal{H}$ recouvre $u$ ;
- On modifie $\overline{\mathcal{H}_i}$ afin qu'il contienne exactement tous les motifs d'étiquettes nulles du niveau $i$ de $\mathcal{H}$.

### Étape B : Calcul d'une sous-structure $k$,$\Delta$-répétitive

Cette étape est dédiée à l'identification des autres répétitions dans $u$ de l'occurrence du motif identifié dans l'étape A.

**Calcul B**

$\forall(b_3,e_3,\alpha_3) \in \overline{\mathcal{H}_i}$, calculer $s^*(u[b_1^* \dots e_1^*], u[b_3 \dots e_3])$

On identifie tous les facteurs dont le score de similarité locale avec $v_1^*$ est $\Delta$-proche du score $s_{\max}$. En utilisant la définition 1, un facteur $v = u[b_3 \dots e_3]$ est conservé si et seulement si $s^*(u[b_1^* \dots e_1^*], u[b_3 \dots e_3]) \geq \Delta \cdot s_{\max}$. Dans ce cas, on note $(b_3^*,e_3^*)$ les indices dans $u$ du facteur de $v$ effectivement aligné.

L'algorithme correspond alors à la suite d'opérations suivante :

- Le motif $(b_3^*,e_3^*,\alpha)$ est ajouté à $\mathcal{H}$ ;
- On retire de $\mathcal{H}$ les motifs d'étiquettes $\Theta_i$ ;
- On ajoute à $\mathcal{H}$ le nombre minimum de motifs d'étiquettes nulles $\Theta_i$ afin que le niveau $i$ de $\mathcal{H}$ recouvre $u$ ;
- $\overline{\mathcal{H}_i}$ est modifié afin qu'il contienne exactement les motifs d'étiquettes nulles du niveau $i$ de $\mathcal{H}$.

### Étape C : Calcul d'une sous-structure $\Delta$-répétitive optimale

On note $M_1$, $M_2$ et $M_3$ les motifs de niveau $i-1$ d'étiquettes respectives $\beta_1$, $\beta_2$ et $\beta_3$ qui dominent respectivement $m_1$, $m_2$ et $m_3$. On désigne alors par $\mathcal{M}$ l'ensemble des motifs de niveau $i-1$ de même étiquette que l'un des motifs $M_1$, $M_2$ ou $M_3$ et distincts de ces derniers :

$$\mathcal{M} = \{(b_4,e_4,\alpha_4) \in \mathcal{H} \backslash \{M_1,M_2,M_3\} | \alpha_4 \in \{\beta_1,\beta_2,\beta_3\}\}.$$

**Calcul C**

$\forall(b_4,e_4,\alpha_4) \in \mathcal{M}$, calculer $s^*(u[b_1^* \dots e_1^*],u[b_4 \dots e_4])$

Ce processus est effectué pour chaque motif $M$ de $\mathcal{M}$. Quel que soit le score de similarité obtenu, le facteur de l'occurrence de $M$ effectivement aligné, noté $v = u[b_4 \dots e_4]$, est conservé.

L'algorithme correspond alors à la suite d'opérations suivante :

- Pour chaque motif de $\mathcal{M}$, le motif $(b_4,e_4,\alpha)$ obtenu par le calcul ci-dessus est ajouté à $\mathcal{H}$ ;
- On retire de $\mathcal{H}$ les motifs d'étiquettes $\Theta_i$ ;
- On ajoute à $\mathcal{H}$ le nombre minimum de motifs d'étiquettes nulles $\Theta_i$ afin que le niveau $i$ de $\mathcal{H}$ recouvre $u$ ;
- $\overline{\mathcal{H}_i}$ est modifié afin qu'il contienne exactement les motifs d'étiquettes nulles du niveau $i$ de $\mathcal{H}$.

**Conditions d'arrêt**

Les trois étapes introduites ci-dessus sont répétées jusqu'à ce que l'une des conditions d'arrêt sur $\overline{\mathcal{H}_i}$ suivantes soit vérifiée :

(i) L'ensemble des motifs d'étiquettes non nulles du niveau $i$ de $\mathcal{H}$ recouvre $u$ : $\overline{\mathcal{H}_i} = \emptyset$ ;

(ii) Toutes les occurrences de motifs d'étiquettes nulles du niveau $i$ de $\mathcal{H}$ sont de taille $\Gamma$-insuffisante. Avec le critère défini en 2, on a alors :

$$\forall(b,e,\alpha) \in \overline{\mathcal{H}_i},\ e - b < \Gamma|u|.$$

#### 5.3.3.2 Propriétés

Dans cette section, nous détaillons les propriétés des motifs identifiés par l'algorithme exposé ci-dessus. Nous exposons formellement ces différentes propriétés et proposons une interprétation musicale, lorsqu'adaptée.

**Caractérisation de l'étape A**

L'étape A est divisée en deux tâches. La tâche (i) consiste à calculer les répétitions majeures pour toutes les occurrences de motifs. La tâche (ii) permet de calculer la meilleure répétition d'une occurrence de motif non étiqueté. Par conséquent, cette étape permet de comparer toutes les occurrences de motifs non étiquetés avec le reste des facteurs de $u$, en préservant la contrainte de non chevauchement avec une section déjà étiquetée.

**Proposition 11 (Étape A)** *Le résultat de l'étape A correspond à la répétition non chevauchante la plus significative entre l'occurrence d'un motif d'étiquette nulle au niveau $i$ et un autre facteur de $u$.*

L'étape A introduit ainsi un nouveau motif dans la hiérarchie. Ce nouveau motif est choisi par un critère d'optimalité du score de similarité locale : il correspond à un compromis entre la répétition de plus fort degré de similarité et de plus longue taille. L'ordre des répétitions musicales introduites par l'algorithme d'inférence est donc directement lié à l'aspect significatif des structures répétitives détectées : si une première répétition est introduite avant une deuxième par l'algorithme d'inférence, alors la celle-ci est plus significative que la deuxième.

#### Caractérisation de l'étape B

L'étape B recherche les motifs correspondant à des répétitions multiples des occurrences des motifs identifiés en étape A.

**Proposition 12 (Étape B)** *À l'issue de l'étape B, toutes les répétitions de l'occurrence du motif identifié en A dans des motifs non étiquetés ont été identifiées.*

Cette étape est construite afin de détecter les répétitions musicales multiples. Le seuil $\Delta$ est nécessaire pour assurer une robustesse à des variations musicales légères. Par exemple, afin d'identifier plusieurs couplets dans un morceau, il est important de considérer une légère tolérance sur le score de similarité calculé, ces couplets pouvant présenter quelques différences sur les paroles, les instruments, la mélodie *etc.*

#### Caractérisation de l'étape C

La tâche (ii) de l'étape A peut introduire un nouveau motif, étiqueté $\alpha$, susceptible d'être dominé par un motif identifié lors d'une itération précédente de la récurrence, étiqueté $\beta$. L'étape C assure que les relations de transitivité hiérarchiques soient appliquées.

**Proposition 13 (Étape C)** *À l'issue de l'étape C, tous les motifs d'étiquette $\beta$ dominent un motif d'étiquette $\alpha$.*

Cette étape est cruciale pour assurer une bonne représentation des niveaux hiérarchiques de la structure répétitive. En effet, grâce à l'application de la transitivité, toute création d'un lien de filiation, sous-jacente à la domination d'un motif par un autre, donne lieu à la création de tous les liens de filiation des motifs similaires.

#### Terminaison

La terminaison de l'algorithme est assurée par la proposition suivante :

**Proposition 14 (Terminaison)** *La taille de la sous-séquence de motifs $\overline{\mathcal{H}_i}$ est décroissante suivant $i$.*

**Preuve 3** *Soit $i$ une étape de la récurrence. La proposition 11 indique que l'étape A de l'algorithme introduit nécessairement une répétition correspondant à l'occurrence d'un motif d'étiquette nulle appartenant au niveau $i$, donc à l'un des éléments de $\overline{\mathcal{H}_i}$, noté $m_\Theta$. En conséquence, $m_\Theta$ domine ce nouveau motif $m$. La redéfinition dans l'étape A des motifs d'étiquettes nulles supprime alors le motif $m_\Theta$, et introduit*

*un ensemble $E$ de motifs dont les occurrences sont non chevauchantes avec l'occurrence de $m$. Or, $E$ est défini par le nombre minimum d'éléments nécessaires pour compléter le niveau $i$ ; on en déduit que l'occurrence de $m$ est disjointe de toute occurrence d'un motif introduit. En conséquence, la somme des tailles de motifs de $E$ est inférieure à la taille de $m_\Theta$, d'où la taille de $\overline{\mathcal{H}_i}$ décroît à l'issue de l'application de l'étape A.*

Puisque l'une des conditions d'arrêt de la récurrence porte sur la taille des éléments de $\overline{\mathcal{H}_i}$, la décroissance de celle-ci assure la terminaison de l'algorithme.

**Propriétés globales**

Soient $u$ une séquence, et $\mathcal{H}$ une hiérarchie sur $u$ calculée par l'algorithme ci-dessus.

**Proposition 15 (Recouvrement de $\mathcal{H}$)** *Chaque niveau de $\mathcal{H}$ recouvre $u$.*

**Proposition 16 (Niveau d'une itération)** *À chaque itération de la récurrence, les motifs introduits par les étapes A, B et C sont tous sur le même niveau hiérarchique dans $\mathcal{H}$.*

**Preuve 4** *Notons $M_A$ l'ensemble des motifs introduits par l'étape A. Par construction, ces motifs correspondent à la détection d'une nouvelle répétition à partir de la hiérarchie à l'étape $i$, donc au niveau $i$ de $\mathcal{H}$. L'étape B recherche des répétitions entre le motif identifié par l'étape A et un motif de $\overline{\mathcal{H}_i}$ ; par conséquent, l'étape B ne peut introduire un motif dominé par un motif de l'étape A. D'une manière similaire, l'étape C n'introduit pas de motif dominé par un motif des deux étapes précédentes. On en déduit que toutes les occurrences des motifs introduits dans ces trois étapes sont sur le même niveau hiérarchique dans $\mathcal{H}$.*

**Proposition 17 (Étiquettes et niveau hiérarchique)** *Deux motifs de $\mathcal{H}$ possédant la même étiquette appartiennent au même niveau hiérarchique.*

**Preuve 5** *Pour une itération de la récurrence, l'étape A n'attribue à des motifs que des étiquettes nouvellement choisies. Les étapes B et C, quant à elles, ne peuvent attribuer que la même étiquette que l'étape A, ou des nouvelles étiquettes. Par construction, A, B et C n'attribuent donc jamais d'étiquettes déjà attribuées dans des itérations précédentes : un ensemble d'étiquettes est propre à chaque itération de la récurrence. Puisqu'à une itération correspond un seul niveau hiérarchique (Proposition 16), on en déduit la propriété souhaitée.*

#### 5.3.3.3 Exemple d'exécution

La Figure 5.4 décrit un exemple de l'exécution de l'algorithme défini précédemment. À chaque étape, l'ensemble $\overline{\mathcal{H}_i}$ des motifs d'étiquettes nulles au niveau $i$ est représenté par des sections blanches. Les sections noires indiquent les motifs identifiés à chaque étape.

(i) présente le motif initial $(1,n,\Theta_0)$ ;

(ii) montre le résultat de l'étape A de la récurrence. Puisque $\overline{\mathcal{H}_1}$ ne contient que le motif initial, le score $s_1$ de répétition majeure est calculé (tâche (i) de A), et deux motifs à la même étiquette $\alpha$ sont définis ;

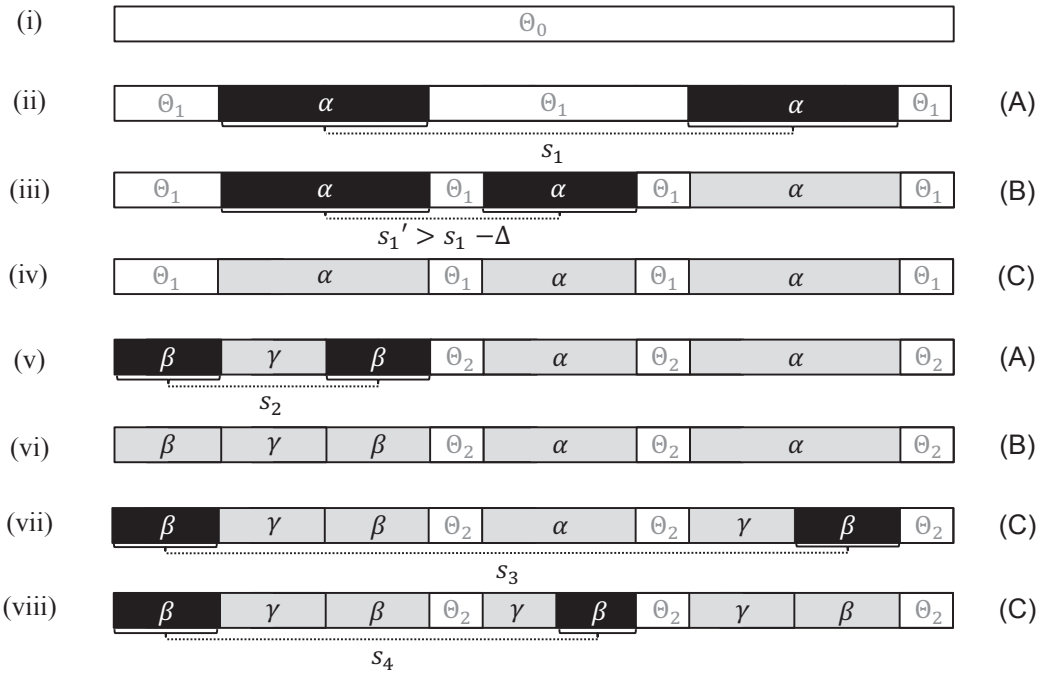


FIG. 5.4 – *Exemple d'exécution étape par étape de l'algorithme d'inférence de structures répétitives (voir texte).*

(iii) l'étape B de la récurrence permet d'identifier un nouveau facteur de score de similarité $s_1'$ avec la section identifiée précédemment. Ce score étant $\Delta$-proche de $s_1$, un nouveau motif étiqueté $\alpha$ est défini ;

(iv) prend en compte des possibles relations de transitivité, aucune n'étant applicable à ce point de l'algorithme ;

(v) présente la première étape de l'itération suivante de la récurrence. Une nouvelle répétition est identifiée comme principale, et deux motifs à la nouvelle étiquette $\beta$ sont définis. Cette répétition correspond à l'alignement entre l'occurrence d'un motif d'étiquette nulle et l'occurrence d'un motif déjà étiqueté $\alpha$. Puisque le second motif est dominé par un motif déjà étiqueté, une nouvelle étiquette $\gamma$ est ajoutée ;

(vi) montre l'étape B de recherche de répétitions secondaires, qui ne détecte aucune nouvelle répétition pour cette itération ;

(vii) et (viii) prennent en compte des relations de transitivité pour insérer de nouveaux motifs étiquetés $\beta$ pour chaque motif étiqueté $\alpha$ identifié précédemment.

À l'issue de l'étape (viii), les motifs restants dans $\overline{\mathcal{H}_2}$ sont jugés de tailles $\Gamma$-insuffisantes, et l'algorithme termine.

La Figure 5.5 représente la structure arborescente ainsi calculée. (ii) représente le résultat de la première itération de la récurrence, à laquelle l'étiquette $\alpha$ est introduite, et (iii) représente le résultat de la seconde itération, à laquelle l'étiquette $\beta$ est introduite.

La Figure 5.5-(iv) illustre une représentation extraite de l'arbre, définie comme la sous-séquence de motifs du deuxième niveau hiérarchique dont les étiquettes nulles ont été remplacées par autant de nouvelles étiquettes.

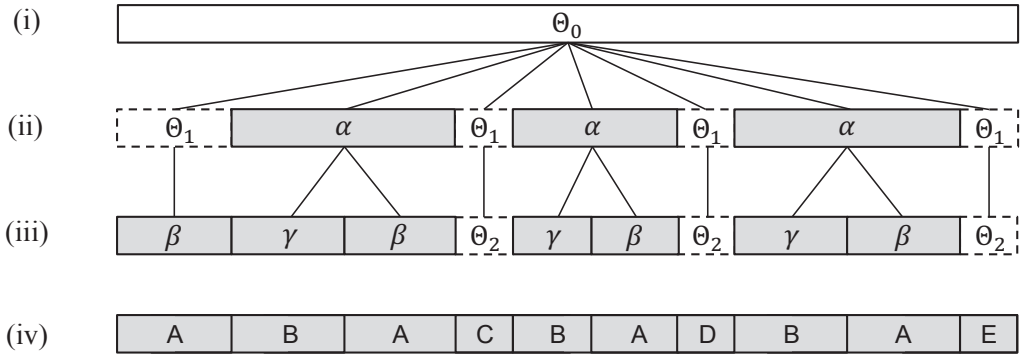


FIG. 5.5 – *Exemple de résultat de l'algorithme d'inférence de structures répétitives. En (i)-(ii)-(iii) sont représentés les trois niveaux de la structure arborescente calculée, (ii) correspondant au résultat de la première itération de la récurrence et (iii) au résultat de la seconde. (iv) montre la représentation séquentielle du niveau 2 de l'arbre.*

### 5.3.4 Expériences et résultats

La Figure 5.6-(ii) représente un exemple de résultat de la hiérarchie structurelle calculée par notre algorithme d'inférence sur le morceau *Eine Kleine Nachtmusik* de *Mozart*. Afin de juger du résultat, l'annotation de ce morceau [Wri07] a été représentée en Figure 5.6-(i) [1]. Ce morceau suit une forme sonate, dont le schéma général est représenté en Figure 5.2-(i). Seuls les quatre premiers niveaux de la hiérarchie sont représentés. Le premier niveau identifie la structure correspondant à la répétition de l'exposition (motifs d'étiquette $E$). La répétition secondaire correspondant à la réexposition n'est pas identifiée par notre algorithme, probablement à cause d'une trop forte variation des séquences tonales dans sa seconde partie (représentée par les motifs $T'_2$ et $T'_3$ dans l'annotation). Le deuxième niveau représente l'identification du premier thème $T_1$ de l'exposition. On remarque qu'une répétition de ce motif $T_1$ est également identifiée dans la réexposition. Ce motif témoigne de la forte similarité entre les débuts des expositions et réexposition. Le niveau (iii) introduit le motif $A_1$ correspondant notamment au premier thème de l'exposition. Ce motif est également identifié au sein du développement (motif $D$ dans l'annotation) et dans la réexposition. Il convient de noter que notre implémentation emploie dans cet exemple des calculs de similarité robustes aux transpositions locales (voir Section 3.1.5), permettant ainsi d'identifier le motif $A_1$ comme répété au sein du développement $D$ malgré une transposition de celui-ci.

Cet exemple met en avant la pertinence de la hiérarchie structurelle identifiée. Bien que celle-ci ne soit pas aussi complète et précise que la vérité-terrain, elle présente de fortes similitudes sur la forme sonate étudiée.

La section suivante évalue plus précisément la qualité des résultats de notre algorithme d'inférence en les comparant à l'état de l'art.

1. Voir http://oyc.yale.edu/music/musi-112/lecture-9 pour une présentation détaillée et commentée de cette structure par C. Wright (Accédé en Septembre 2012).

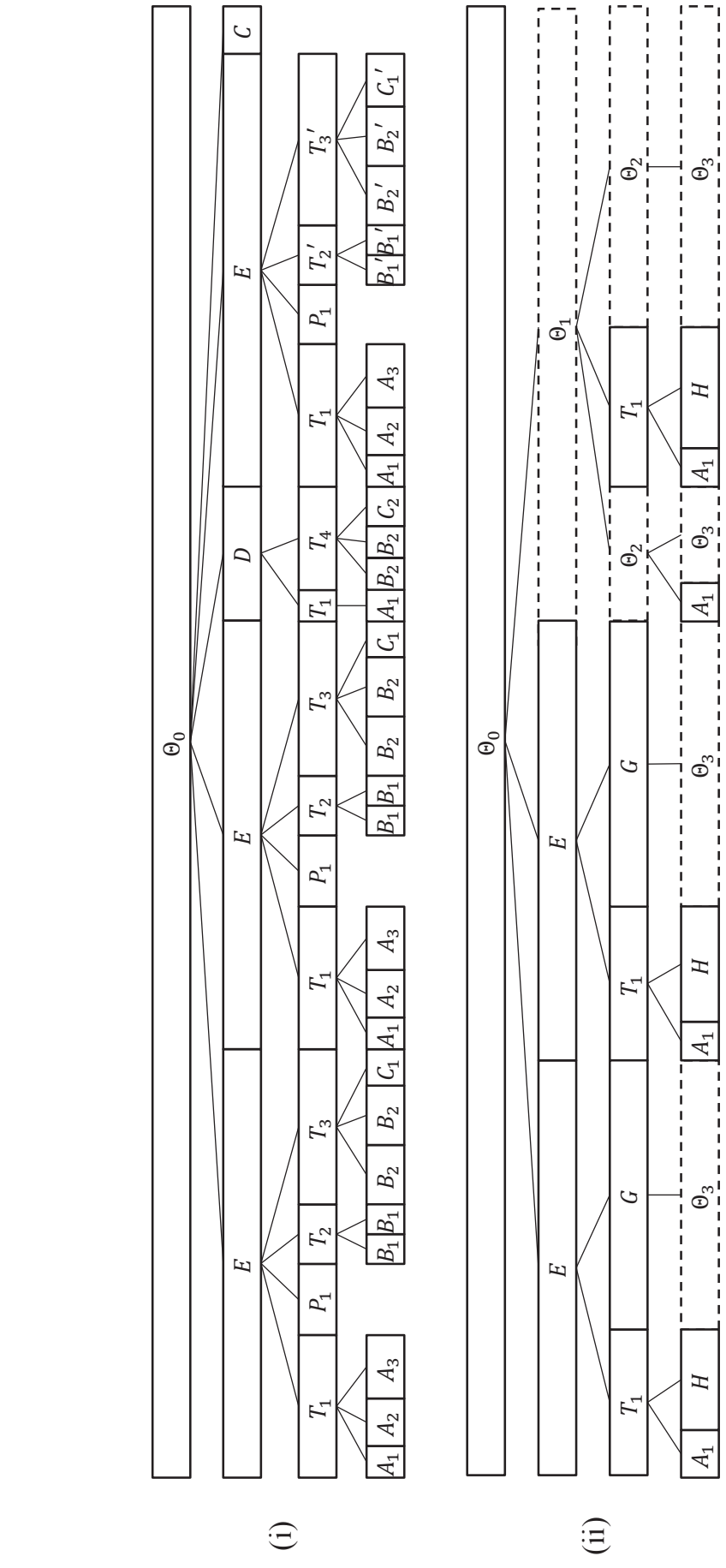


FIG. 5.6 – *Exemple de résultat de l'inférence hiérarchique calculée par notre algorithme sur le morceau Eine Kleine Nachtmusik de Mozart (forme sonate). (i) : Annotation des répétitions sur différents niveaux caractéristiques de la forme sonate [Wri07]. (ii) : Résultat de l'algorithme d'inférence hiérarchique.*

#### 5.3.4.1 Évaluation de segmentations structurelles

Comme expliqué précédemment, la structuration répétitive est systématiquement calculée sous la forme de segmentations structurelles. Il n'existe pas à l'heure actuelle de procédure d'évaluation de hiérarchies de répétitions. Dans cette section, nous proposons donc d'évaluer une segmentation structurelle issue de la représentation hiérarchique obtenue par notre algorithme.

L'obtention d'une segmentation structurelle à partir d'une hiérarchie peut aisément être effectuée en isolant un niveau hiérarchique et en l'étiquetant entièrement, comme l'indique la Définition 17.

Les expériences décrites dans les sections suivantes correspondent à des évaluations de notre algorithme dans le cadre du *Music Information Retrieval Evaluation eXchange* (MIREX)[1] [Dow08, DEBJ10]. L'intérêt scientifique de l'évaluation MIREX est d'assurer des conditions expérimentales identiques entre les méthodes testées, et ainsi de permettre une comparaison fiable des algorithmes existants. Les résultats présentés dans cette section correspondent à la tâche d'évaluation de segmentations structurelles de 2010[2] et au complément d'évaluation de cette tâche, publié par l'équipe du MIREX en 2011 [EMD+11].

#### 5.3.4.2 Bases de données

Trois bases de tests sont utilisées dans le cadre des évaluations de segmentation structurelle MIREX. Puisque l'inférence de structures musicales est étudiée dans la littérature sous une approche séquentielle, consistant à identifier des segmentations structurelles, ces bases de tests comprennent des annotations des répétitions musicales sous la forme de sous-séquences de motifs recouvrantes.

La première base de tests, notée $STRUCT_A$, correspond à l'agrégation d'annotations réalisées par différents laboratoires et universités[3] et est notamment distribuée via le projet OMRAS2[4] [MCD+09]. Elle se compose d'un ensemble de 297 annotations de la structuration des morceaux de musique de différents artistes de musique populaire occidentale, réalisées par des experts.

La deuxième base de tests, notée $STRUCT_B$, est un produit du projet Quæro[5] et comporte un ensemble d'annotations de segments structurels effectuées selon les critères d'annotation décrits dans [BLBSV10]. Celles-ci sont réalisées sur les 100 morceaux de la base de données *RWC Pop Database* [GHNO02] réalisées par des experts. Il convient de noter que cette base de données n'assigne pas d'étiquettes aux motifs identifiés, mais renseigne uniquement sur les frontières entre éléments structurels.

La troisième base de tests, notée $STRUCT_C$, est fournie par le projet *Structural Analysis of Large Amounts of Music Information* (SALAMI)[6] [SBF+11] et comporte un ensemble d'annotations de segmentations structurelles réalisées par

1. http://www.music-ir.org/mirexwiki
2. http://www.music-ir.org/mirex/wiki/2010:Structural_Segmentation
3. Dont notamment *IRCAM Paris*, *Queen Mary University of London*, *Universitat Pompeu Fabra, Barcelona* et *Tampere University of Technology*, voir [PD09] pour un descriptif des contributions
4. http://isophonics.net/content/reference-annotations
5. http://www.quaero.org
6. http://ddmal.music.mcgill.ca/research/salami/annotations

| | | |
|---|---|---|
| (a) | Peeters | [PBR02, Pee04, Pee07] |
| (b) | Weiss, Bello | [WB10] |
| (c) | Méthode proposée | [MHR+10] |
| (d) | Mauch, Noland, Dixon | [MND09] |
| (e) | Sargent et al. 1 | [SBV10] |
| (f) | Sargent et al. 2 | [SBV10] |

TAB. 5.2 – *Nomenclature des méthodes évaluées.*

des experts selon les principes d'annotation exposés dans [PD09]. Cette vérité terrain annote 1383 morceaux de styles occidentaux variés. Elle possède en outre deux particularités par rapport aux autres bases de test :

1. Tous les morceaux sont annotés par chaque expert sur 2 niveaux de description structurelle répétitive ;
2. 1048 morceaux sont annotés par 2 experts différents.

Les méthodes de l'état de l'art auxquelles notre technique est comparée correspondent aux algorithmes soumis à la tâche MIREX 2010 et sont listées dans le Tableau 5.2. Pour plus de détails sur chacune des méthodes, nous invitons le lecteur à se référer à la Section 5.1. Outre l'évaluation sur les bases $STRUCT_A$ et $STRUCT_B$ dans le cadre de cette tâche, ces méthodes ont également été évaluées sur la base de données $STRUCT_C$ dans le contexte d'un complément d'évaluation publié par l'équipe du MIREX [EMD+11]. Il convient de préciser que les résultats d'évaluation effectués dans les campagnes d'évaluation MIREX 2011 et MIREX 2012 ne sont pas rapportés dans notre étude, la première campagne ne considérant pas la base conséquente $STRUCT_C$ comme ensemble de test, et la seconde n'étant pas finalisée à la date d'écriture de ce manuscrit. Les méthodes de l'état de l'art présentées dans ces travaux sont ainsi les seules, à la date de septembre 2012, à avoir été évaluées par l'équipe du MIREX sur les trois bases de données décrites ci-dessus. Si ce choix nous permet de comparer les méthodes de l'état de l'art sous des conditions d'évaluation communes, il est important de nuancer les conclusions expérimentales effectuées dans les sections suivantes par leur relative ancienneté à la date de publication de ce manuscrit.

#### 5.3.4.3 Métriques d'évaluation

Quantifier la similarité entre deux descriptions structurelles est un problème complexe [Luk08], comme exposé dans cette section. Les mesures d'évaluation de segmentations structurelles utilisées dans la littérature peuvent être regroupées en deux catégories :

- Les mesures d'évaluation des frontières structurelles
- Les mesures d'évaluation des motifs

Deux mesures d'évaluation des frontières sont principalement utilisées.

**Rappel et précision des frontières** La première mesure permet d'appliquer un schéma classique de calcul de rappel/précision [MRS08] sur les frontières de deux segmentations structurelles, en permettant de légères variations temporelles de leur position [TL07]. Pour une taille de fenêtre fixée $x$, un *faux négatif* (respectivement *faux positif*) correspond à une frontière présente à la date $t$ dans la vérité terrain

| | $F_{@0.5}$ | $P_{@0.5}$ | $R_{@0.5}$ | $F_{@3}$ | $P_{@3}$ | $R_{@3}$ | $M_{A\|E}$ | $M_{E\|A}$ |
|---|---|---|---|---|---|---|---|---|
| (a) | 0.23 | 0.23 | 0.23 | 0.57 | 0.58 | 0.59 | 1.57 | 1.79 |
| (b) | 0.29 | 0.36 | 0.25 | 0.58 | 0.72 | 0.50 | 3.60 | 1.58 |
| **(c)** | **0.20** | **0.32** | **0.15** | **0.49** | **0.75** | **0.37** | **4.36** | **1.61** |
| (d) | 0.36 | 0.44 | 0.32 | 0.61 | 0.74 | 0.54 | 2.66 | 1.27 |
| (e) | 0.23 | 0.24 | 0.24 | 0.61 | 0.62 | 0.62 | 2.12 | 2.22 |
| (f) | 0.24 | 0.25 | 0.24 | 0.61 | 0.62 | 0.62 | 2.73 | 2.16 |

TAB. 5.3 – *Résultats de l'évaluation MIREX 2010 sur la base de données $STRUCT_B$ pour les 6 méthodes soumises. L'évaluation ne porte que sur les frontières.*

(resp. dans la segmentation estimée), mais absente à une date comprise entre $t-x$ et $t+x$ dans la segmentation estimée (resp. dans la vérité terrain). À partir de cette estimation simple, la mesure fournit alors un score de précision $P_{@x}$, un score de rappel $R_{@x}$ et une F-mesure $F_{@x}$.

**Écarts médians entre frontières** Une alternative à cette mesure pour l'estimation des frontières consiste à calculer la durée médiane entre les frontières de l'annotation et les frontières calculées automatiquement, et vice-versa [TL07]. On obtient alors les deux mesures $M_{A|E}$ et $M_{E|A}$, désignant respectivement la médiane des durées entre les frontières annotées et les frontières estimées les plus proches, et la médiane des durées entre les frontières estimées et les frontières annotées les plus proches. Aux systèmes les plus précis doivent donc correspondre des valeurs faibles.

L'évaluation des motifs identifiés peut être effectuée à l'aide des deux mesures suivantes.

**Rappel et précision de l'étiquetage trame-à-trame** Afin de comparer les motifs de deux segmentations structurelles, Levy et Sandler [LS08] proposent une adaptation du schéma de calcul rappel/précision [MRS08] à l'échelle temporelle la plus petite, la trame de signal. Plus précisément, on note $F_a$ (respectivement $F_e$) l'ensemble des paires de trames dominées par des motifs d'une même étiquette dans la structure annotée (resp. dans la structure estimée). Les scores de précision $P_p$ et rappel $R_p$ trame-à-trame sont alors définis [LS08] par :

$$P_p = \frac{|F_e \cap F_a|}{|F_e|} \text{ et } R_p = \frac{|F_e \cap F_a|}{|F_a|}.$$

Ces mesures de rappel et précision peuvent être résumées en une F-mesure [MRS08], notée $F_p$. Cette métrique permet d'obtenir une estimation précise de la similarité entre deux segmentations structurelles. En revanche, elle ne tient pas compte de l'échelle temporelle de description de la structure. Comme le soulignent notamment Paulus et Klapuri [PK06], Chai [Cha05] ou Lukashevich [Luk08], cette mesure fonctionne par paires de trames mais ne prend pas en compte l'ordre de ces trames. En d'autres termes, elle ne tient pas compte des différences potentielles d'échelles temporelles de description (niveaux hiérarchiques) entre les deux segmentations structurelles comparées [PMK10]. La mesure suivante permet une plus grande robustesse à cette différence de niveau.

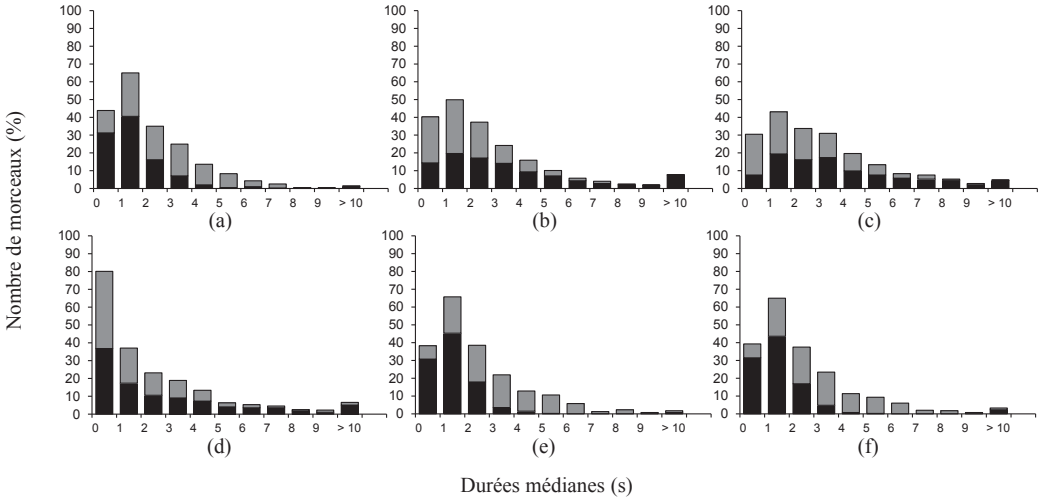


FIG. 5.7 – *Distribution des durées médianes entre frontières identifiées et annotées sur les bases de données $STRUCT_A$ et $STRUCT_B$. La durée $M_{A|E}$ est représentée en noir, et la durée $M_{E|A}$ en gris.*

**Information mutuelle** La structure musicale ayant une nature hiérarchique, sa description sous forme de séquence peut être effectuée à différentes échelles temporelles. Ainsi, il est possible que la segmentation estimée et la segmentation annotée décrivent chacune une structuration plausible, mais ne se trouvent pas sur une même échelle de description. Afin de prendre en compte cette éventualité, Lukashevich [Luk08] introduit deux scores d'évaluation $S_o$ et $S_u$, respectivement *score de sur-segmentation* et *score de sous-segmentation* pour comparer deux segmentations structurelles séquentielles sur leur échelle temporelle de description. Ces mesures sont basées sur le travail d'Abdallah *et al.* [ANS+05] qui estime des entropies conditionnelles dans le cadre d'un estimateur Bayésien et montre que ces grandeurs quantifient de manière précise et distincte la quantité d'information manquante d'une part, et la quantité d'information superflue d'autre part pour la comparaison de segmentation structurelles.

#### 5.3.4.4 Évaluation des frontières

On propose tout d'abord d'évaluer les frontières de la segmentation estimée et leur correspondance à celles de la vérité terrain. Le Tableau 5.3 présente les résultats d'évaluation des frontières pour chaque méthode sur la base de données $STRUCT_B$. Les colonnes 1 à 6 fournissent les taux moyens de F-mesure, précision et rappel des frontières pour des fenêtres de 0.5 et 3 secondes. Dans les deux cas, notre méthode semble relativement peu précise dans son identification des frontières. Sur cet ensemble de données, la F-mesure des frontières identifiées par notre méthode est de 20% avec une fenêtre de tolérance de 0.5s, et 49% avec une fenêtre de tolérance de 3s. Les deux dernières colonnes du Tableau 5.3 indiquent les durées médianes entre les frontières estimées et annotées. Le score élevé $M_{A|E} = 4.36$ indique que la durée médiane entre une frontière annotée et une frontière estimée est de plus de 4 secondes ; le score $M_{E|A} = 1.61$, en revanche, est beaucoup plus faible. Cette forte

| | $S_o$ | $S_u$ | $F_p$ | $P_p$ | $R_p$ |
|---|---|---|---|---|---|
| (a) | 0.60 | 0.68 | 0.54 | 0.63 | 0.51 |
| (b) | 0.67 | 0.54 | 0.54 | 0.53 | 0.64 |
| **(c)** | **0.68** | **0.54** | **0.55** | **0.51** | **0.68** |
| (d) | 0.76 | 0.61 | 0.61 | 0.55 | 0.74 |
| (e) | 0.59 | 0.69 | 0.50 | 0.60 | 0.47 |
| (f) | 0.71 | 0.43 | 0.49 | 0.42 | 0.73 |

| | $F_{@0.5}$ | $P_{@0.5}$ | $R_{@0.5}$ | $F_{@3}$ | $P_{@3}$ | $R_{@3}$ | $M_{A\|E}$ | $M_{E\|A}$ |
|---|---|---|---|---|---|---|---|---|
| (a) | 0.18 | 0.14 | 0.26 | 0.50 | 0.40 | 0.70 | 1.90 | 3.43 |
| (b) | 0.20 | 0.20 | 0.22 | 0.48 | 0.46 | 0.52 | 5.55 | 2.39 |
| **(c)** | **0.19** | **0.20** | **0.18** | **0.51** | **0.55** | **0.50** | **4.05** | **3.05** |
| (d) | 0.32 | 0.34 | 0.33 | 0.61 | 0.63 | 0.63 | 3.04 | 2.43 |
| (e) | 0.22 | 0.18 | 0.29 | 0.57 | 0.47 | 0.76 | 1.83 | 3.92 |
| (f) | 0.22 | 0.19 | 0.29 | 0.56 | 0.47 | 0.74 | 2.80 | 3.86 |

TAB. 5.4 – *Résultats de l'évaluation MIREX 2010 sur la base de données $STRUCT_A$ pour les 6 méthodes soumises. L'évaluation porte sur les motifs (haut) et les frontières (bas).*

différence suggère que les annotations de la base de données $STRUCT_B$ comportent sensiblement plus de frontières. En d'autres termes, notre méthode semble *sous-segmenter* la structure musicale par rapport à ces annotations.

La Figure 5.8 représente un exemple de résultat sur $STRUCT_B$ [1]. Chaque ligne correspond à une méthode, la dernière représentant la vérité terrain. Les lignes pointillées mettent en valeur la concordance entre les frontières estimées et annotées. Cet exemple met en valeur l'aspect sous-segmenté de notre méthode (c) par rapport à l'annotation. Cependant, les frontières identifiées restent proches dans ce cas.

Les colonnes 1 à 3 du Tableau 5.4-bas fournissent les taux moyens d'évaluation des frontières pour la base de données $STRUCT_A$. La détection des frontières semble plus proche des autres méthodes pour cette base de données, avec une F-mesure de 19% pour une tolérance de 0.5 secondes, et une F-mesure de 51% pour une tolérance de 3 secondes. Les valeurs $M_{A|E}$ et $M_{E|A}$ indiquent un écart des frontières estimées et détectées de valeurs médianes respectives 4.05 et 3.05 secondes. La plus grande proximité entre ces deux valeurs suggère une meilleure adaptation de l'algorithme pour la base de données $STRUCT_A$.

La Figure 5.7 représente la distribution des valeurs médianes calculées sur les morceaux de la base de données $STRUCT_A$ par chacune des méthodes. L'axe des abscisses indique les plages de valeurs médianes possibles (entre 1 et 10 secondes, puis plus de dix secondes pour le dernier coefficient), tandis que l'axe des ordonnées indique le pourcentage de morceaux concernés. Les barres noires représentent les valeurs $M_{A|E}$, tandis que les barres grises représentent les valeurs $M_{E|A}$. Les méthodes les plus performantes sont caractérisées par une distribution aux valeurs importantes dans les premiers coefficients de l'histogramme, puis des valeurs faibles ou nulles pour les coefficients suivants. Comme précédemment, le déséquilibre entre les scores, donc les deux types de barres, témoigne d'une tendance à sur ou sous-segmenter. La méthode (d) semble fournir la plus grande précision pour l'identification des

1. Détail obtenu grâce à l'outil NEMA de l'IMIRSEL : http://nema.lis.illinois.edu/

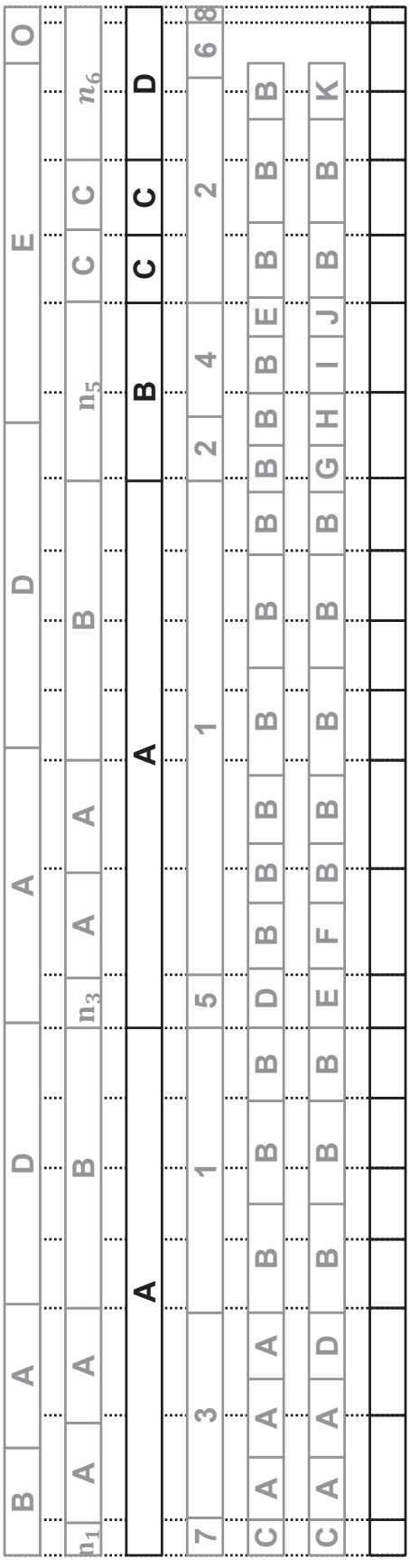


FIG. 5.8 – *Exemple de résultats obtenus sur le morceau* struct_mrx_10_1 *de la base de données* $STRUCT_B$. *Le résultat exact des méthodes (a) à (f) est représenté, la dernière ligne correspondant à la vérité terrain.*

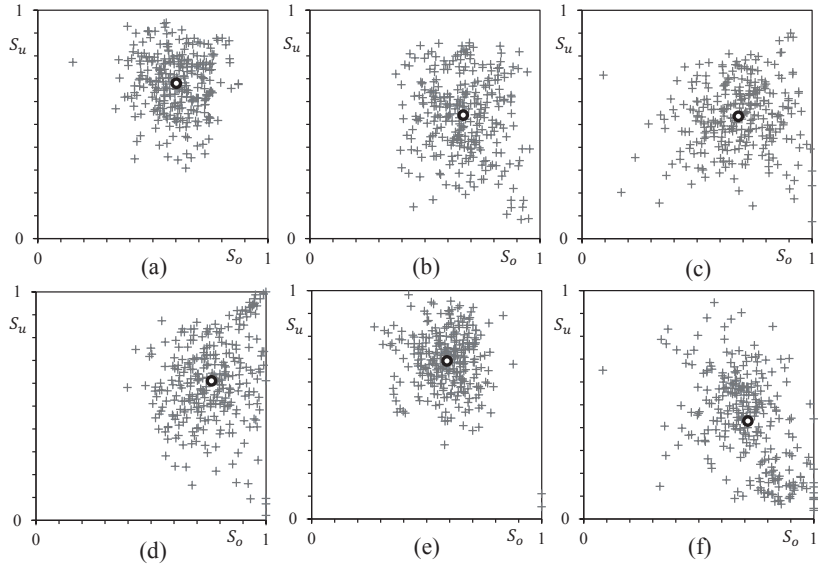


FIG. 5.9 – *Distribution des scores de sous-segmentation et de sur-segmentation pour les morceaux de la base de données $STRUCT_A$.*

frontières, puisqu'environ 40% des durées médianes sont inférieures à la seconde, pour les deux scores. Pour toutes les méthodes, les médianes entre frontières annotées et estimées se situent dans plus de 80% des cas entre 1 et 5 secondes. La distribution correspondant à notre méthode (c) souligne la relative faible précision de la détection des frontières.

#### 5.3.4.5 Évaluation des motifs

Les colonnes 3 à 5 du Tableau 5.4-haut présentent les scores d'évaluation de l'étiquetage trame-à-trame sur la base de tests $STRUCT_A$. Les colonnes 4 et 5 représentent respectivement la précision et le rappel par paires de trames, et la colonne 3 résume les deux scores en une F-mesure. L'étiquetage de notre méthode a un rappel moyen de 68% pour une précision de 51%, soit une F-mesure de 55%. Seule la méthode (d) semble fournir un résultat plus efficace, avec une F-mesure de 61%. On note cependant que les scores d'évaluation des méthodes semblent proches, aucune ne présentant *a priori* de fortes différences.

La Figure 5.10 représente un exemple de résultat pour chacune des méthodes pour un morceau de la base de données $STRUCT_A$. Dans cet exemple, notre méthode fournit un résultat proche de l'annotation. Le motif instrumental `S` de la vérité terrain est identifié comme similaire au motif `RC`. On note également quelques imprécisions sur les frontières, ainsi que la non détection de la conclusion `O`. Enfin, les motifs `R` et `C` sont réunis en un unique motif `B`, qui suggère que l'échelle de description de la structure estimée est différente de celle annotée. La section suivante évalue cet aspect sur toute la base de données.

#### 5.3.4.6 Évaluation de l'échelle de description

Les structures répétitives sont de nature hiérarchique. Il est alors pertinent d'évaluer dans quelle mesure la segmentation structurelle estimée correspond à la

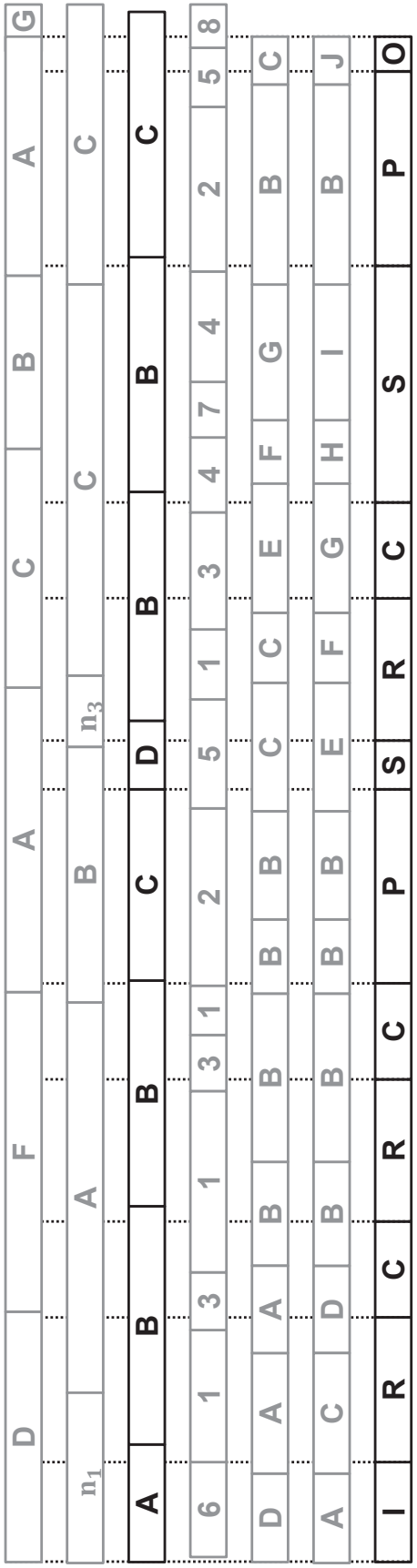


FIG. 5.10 – *Exemple de résultats obtenus sur le morceau* struct_mrx_09_33 *de la base de données* $STRUCT_A$. *Le résultat exact des méthodes (a) à (f) est représenté, la dernière ligne correspondant à la vérité terrain.*

| | Live | Classique | Jazz | Pop | World | Moyenne |
|---|---|---|---|---|---|---|
| (a) | 0.51 | 0.43 | 0.48 | 0.51 | 0.48 | 0.48 |
| (b) | 0.56 | 0.52 | 0.55 | 0.55 | 0.54 | 0.54 |
| **(c)** | **0.56** | **0.59** | **0.57** | **0.54** | **0.56** | **0.56** |
| (d) | 0.53 | 0.56 | 0.57 | 0.57 | 0.55 | 0.56 |
| (e) | 0.50 | 0.51 | 0.54 | 0.52 | 0.52 | 0.52 |
| (f) | 0.43 | 0.43 | 0.40 | 0.45 | 0.44 | 0.43 |
| Moyenne | 0.52 | 0.51 | 0.52 | 0.52 | 0.51 | 0.52 |

TAB. 5.5 – *Scores d'évaluation $F_p$ calculés pour chaque méthode et chaque style musical de $STRUCT_C$ (échelle large), d'après [EMD+11].*

même échelle temporelle de description que la segmentation annotée.

Les scores moyens de sur-segmentation et de sous-segmentation sur $STRUCT_A$ sont présentés en colonnes 1 et 2 du Tableau 5.4-haut. Notre méthode, dont les scores moyens sont de $S_o = 68\%$ et $S_u = 54\%$, semble sous-segmenter légèrement la base de données, tout comme les méthodes (b), (d) et (f).

Plus précisément, la Figure 5.9 représente le nuage de points des scores $S_o$ en fonction des scores $S_u$ pour tous les morceaux de la base de données $STRUCT_A$ et pour les différentes méthodes. Chaque mesure correspond à un morceau, le point blanc matérialisant la moyenne de tous les scores pour une méthode. Une mesure dans l'angle supérieur droit représente une correspondance parfaite entre la structure annotée et la structure estimée. En outre, la présence d'une mesure dans la région supérieure gauche de ce graphe indique une tendance à la *sur-segmentation* de la méthode pour un morceau en particulier. Inversement, la présence d'une mesure dans la région inférieure droite indique une tendance à la *sous-segmentation*. Ainsi, les méthodes (a) et (e) semblent définir des descriptions sur-segmentées. Les résultats des méthodes (c) et (d), en revanche, semblent être équilibrés autour de la diagonale et témoignent d'un équilibre moyen entre les deux caractères de sous- et sur-segmentation. Pour notre méthode (c), les mesures proches des axes montrent que quelques morceaux ont une segmentation très éloignée de la vérité terrain. Ces valeurs correspondent en réalité à des morceaux dont l'information tonale ne caractérise pas de structure répétitive. Par exemple, dans *We will Rock You* du groupe *Queen*, l'absence d'information tonale dans une grande partie du morceau rend la structuration par notre méthode non pertinente.

#### 5.3.4.7 Évaluation multi-échelles

Afin de se rapprocher d'une évaluation hiérarchique des segmentations calculées, il est intéressant de considérer la base de tests $STRUCT_C$. En effet, cette base de données de 1300 morceaux fournit pour chaque œuvre deux annotations sur des échelles de temps différentes [EMD+11] : une échelle temporelle dite *large*, correspondant notamment à l'échelle de description couplets/refrains, et une échelle dite *fine*.

La base de données $STRUCT_C$ comprend des morceaux de différents styles musicaux, classés dans 5 catégories : "Live", "Classique", "Jazz", "Pop" et "World". Le Tableau 5.5 présente les taux de F-mesures de l'étiquetage trame-à-trame pour chacune des méthodes en fonction du style musical. Ehmann *et al.* [EMD+11] montrent

| | Échelle | |
|---|---|---|
| | Fine | Large |
| (a) | 0.43 | 0.49 |
| (b) | 0.45 | 0.55 |
| **(c)** | **0.45** | **0.57** |
| (d) | 0.44 | 0.56 |
| (e) | 0.39 | 0.53 |
| (f) | 0.37 | 0.43 |
| Humain | 0.63 | 0.72 |

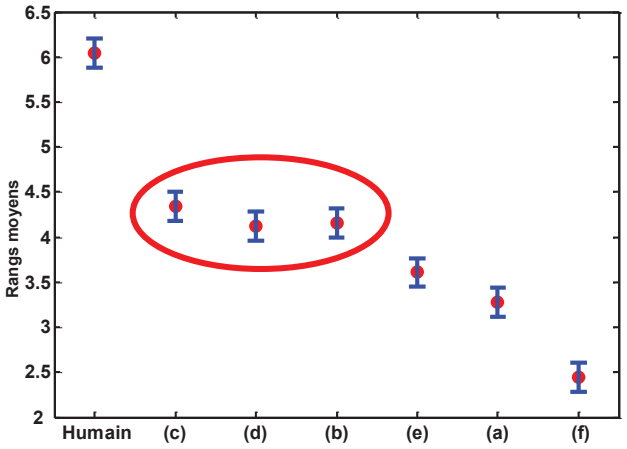


FIG. 5.11 – *Gauche : Résultats pour chaque méthode sur les deux échelles de description, d'après [EMD+11]. Droite : Représentation de la performance des résultats sous la forme d'un test de significativité (TKHSD, voir [Dow08]), d'après [EMD+11]. Le cercle rouge représente le groupe de méthodes ne présentant pas de différence significative.*

qu'il n'existe pas de différence significative sur l'efficacité des méthodes en fonction du style musical. En d'autres termes, le style musical ne semble pas influencer les algorithmes de segmentation dans leur ensemble. Sur la base de données $STRUCT_C$, notre méthode propose en moyenne la meilleure segmentation structurelle. En particulier, les structures répétitives de morceaux de style classique sont identifiées avec une F-mesure moyenne de 59%. On peut expliquer ce score élevé par l'importance de l'information tonale pour les structures répétitives en musique classique.

Le Tableau 5.11-Gauche présente le résultat de la comparaison des segmentations structurelles estimées avec les deux niveaux d'annotation. Toutes les méthodes affichent un score significativement plus élevé dans le cas de l'échelle large, et semblent donc plus adaptées à cette échelle. La dernière ligne du tableau est obtenue en confrontant pour chaque échelle les deux annotations d'experts. Pour l'échelle large, par exemple, aux annotations de deux experts correspond une F-mesure de 72%. Ce résultat met en avant la subjectivité des annotations manuelles de la structure répétitive.

Malgré l'aspect subjectif des annotations, la Figure 5.11-Droite souligne que le résultat des systèmes de segmentation structurelle n'atteint pas les performances humaines. Cette figure est obtenue en réalisant un test de signification statistique des résultats obtenus (voir [Dow08, EMD+11]). L'axe des ordonnées est une mesure de la performance moyenne des algorithmes. Le cercle rouge matérialise un groupe de trois méthodes, dont celle que nous proposons, qui ne présentent pas de différence significative entre elles. En revanche, le test indique une différence entre ce groupe et chacune des autres méthodes.

#### 5.3.4.8 Temps d'exécution

Aucune contrainte n'est imposée par l'équipe MIREX sur les temps d'exécution des algorithmes, ceux-ci étant renseignés à titre indicatif uniquement [1]. Notre méthode se situe dans un temps d'exécution moyen compris entre 1 et 3 minutes par morceau sur les bases de données $STRUCT_A$ et $STRUCT_B$. La base de données $STRUCT_C$ semble requérir plus de calculs, avec un temps moyen d'environ 6 minutes par morceau pour notre méthode [EMD+11]. Si les raisons d'une telle augmentation pour cette dernière base de données ne sont pas explicites, on peut supposer que la présence de morceaux (notamment de musique classique) de durée moyenne plus élevée que les autres ensembles de données implique le calcul d'un nombre de coefficients de programmation dynamique plus important pour notre méthode.

Il convient de préciser, cependant, que le temps de calcul n'est pas considéré dans ces travaux comme un facteur critique pour l'estimation d'un ensemble de structures répétitives, et que de nombreuses techniques d'optimisation peuvent être envisagées en perspective afin de réduire significativement cette durée d'exécution.

#### 5.3.4.9 Bilan expérimental

Nous avons exposé dans cette section les résultats d'évaluation de notre méthode sur trois bases de données annotées en les comparant aux autres algorithmes de l'état de l'art. Les nombreux tests présentés dans cette section soulignent la difficulté d'évaluation de segmentations structurelles sous la forme de séquences.

Les frontières des segmentations structurelles identifiées par notre méthode semblent relativement peu précises par rapport à l'état de l'art. Nous identifions deux raisons majeures pour ce manque de précision : un niveau de description de l'estimation souvent différent de l'annotation dans la hiérarchie structurelle, et une limitation de la pertinence de la structuration tonale pour certains morceaux. Malgré cette limitation, les résultats d'évaluation des motifs identifiés indiquent que notre méthode produit des résultats similaires aux meilleurs des algorithmes testés, notamment par rapport à la base d'annotations $STRUCT_C$.

1. Voir http://www.music-ir.org/mirex/wiki/2010:Runtime (accédé en décembre 2012).

## 5.4 Conclusion du chapitre

Dans ce chapitre, nous avons étudié le problème de l'inférence de structures répétitives depuis l'audio. Nous l'avons d'abord défini tel qu'il est traité dans la littérature, sous la forme de la recherche d'une segmentation structurelle. Nous avons exposé les travaux existants et justifié la nécessité de considérer un modèle différent, permettant de prendre en compte des relations hiérarchiques.

Nous avons proposé notre propre formalisation du problème sous la forme de l'inférence d'une hiérarchie de structures répétitives. Nous avons détaillé un algorithme permettant d'identifier une telle hiérarchie en se basant sur une théorie musicale bien-fondée. Nous avons enfin présenté une série d'évaluation des résultats de l'inférence structurelle selon des principes d'évaluation standardisés.

Bien que notre algorithme s'appuie sur un modèle hiérarchique, l'application de notre méthode pour l'inférence de segmentations structurelles fournit des résultats comparables à l'état de l'art. Malgré une certaine imprécision sur les frontières entre motifs identifiés, les tests suggèrent une position crédible de notre méthode par rapport aux algorithmes de segmentation structurelle de la littérature.

Ce résultat doit cependant être modéré par le caractère subjectif et mal défini des annotations structurelles. De plus, l'évaluation d'inférences hiérarchiques n'étant pas définie, les résultats de notre algorithme n'ont été évalués que partiellement. Ces deux points forment une perspective majeure de nos travaux : construire une méthodologie d'évaluation et d'annotation de hiérarchies structurelles.

# Conclusion et perspectives

Cette thèse décrit un ensemble de travaux autour du problème de l'inférence de structures répétitives à partir du contenu musical. Notre approche a suivi une démarche par étapes successives depuis l'étude de la similarité musicale jusqu'à l'inférence de structures répétitives.

Dans un premier temps, nous avons spécifié dans le Chapitre 1 la notion de structure répétitive dans la musique et justifié notre choix de représentation de l'information musicale sous une forme séquentielle, tout en exposant les enjeux majeurs d'efficacité et de mise en pratique des méthodes que nous proposons.

Nous avons présenté dans le Chapitre 2 un ensemble d'outils mathématiques permettant de représenter le signal audio sous la forme d'une séquence de symboles caractérisant l'information tonale, en justifiant ce choix par sa prépondérance dans la structuration répétitive de la musique.

Nous avons alors défini en Chapitre 3 un formalisme pour l'alignement entre séquences symboliques, que nous avons appliqué dans le cadre d'un système d'estimation de la similarité musicale. Nous avons étudié une solution permettant de réduire significativement le coût calculatoire de ce système afin de rendre possible son utilisation en pratique sur des bases de données de l'ordre de celles accessibles au grand public.

Nous avons introduit dans le Chapitre 4 plusieurs types de répétitions dans les œuvres musicales, et nous avons proposé des algorithmes d'identification de telles répétitions à partir des outils d'estimation de similarité. Ces méthodes ont été évaluées en comparant les résultats obtenus à des éléments de la perception musicale.

Nous avons étudié dans le Chapitre 5 le problème d'inférence de structures répétitives de morceaux de musique. Après le constat d'une définition insuffisante du problème dans la littérature, nous avons proposé une formalisation de celui-ci et avons détaillé un algorithme d'inférence permettant d'identifier une hiérarchie de structures répétitives. Ce dernier a été confronté à l'état de l'art en comparant les résultats avec des annotations perceptives de la structuration musicale.

## 6.1 Résultats majeurs et contributions

Cette thèse est d'abord une contribution à la recherche en information musicale dédiée aux notions de similarité musicale, de répétition et d'analyse de la structuration des morceaux de musique. Ces travaux s'appuient sur une approche algorithmique, généralisant des techniques développées dans le cadre de comparaisons de séquences biologiques.

Nos expériences mettent en évidence que ces techniques permettent d'estimer de manière précise la similarité musicale, notamment dans le cadre du problème de l'identification des reprises. Nous proposons [MBHF12] alors une méthode permettant de résoudre ce problème de manière efficace en adaptant une technique de recherche par l'application de filtres heuristiques successifs. Nous évaluons notre solution sur plusieurs bases de données audio musicales et mettons en évidence une perte d'environ un tiers de la précision du système tout en proposant un gain sub-

stantiel sur le temps de calcul, avec une identification évaluée comme environ 400 fois plus rapide en pratique sur notre jeu de données. Ce résultat suggère qu'estimer la similarité musicale approchée à partir d'heuristiques sur de courtes similitudes locales est une approche prometteuse pour l'analyse efficace de bases de données musicales conséquentes.

Nous présentons ensuite l'utilisation de ces techniques de similarité musicale afin d'identifier des répétitions particulières dans les morceaux de musique.

En premier lieu, nous étudions le cas de la meilleure répétition d'un extrait musical donné dans un morceau. Nous proposons [MHT+11] alors une méthode de reconstruction de données manquantes qui décrit une solution au problème dans un cadre applicatif particulier. À l'issue d'une évaluation par un ensemble de tests perceptifs, nous mettons en valeur la qualité de la méthode au sens de la perception. Ce résultat peut être vu comme la première approche d'un problème plus général de modification d'un morceau de musique en tenant compte de ses répétitions.

En second lieu, nous définissons une répétition particulière, appelée répétition majeure, et nous décrivons un algorithme permettant de l'identifier. Nous montrons [MHRF11a] alors que cette structure répétitive particulière correspond à une structuration perçue dans la musique.

Nous proposons [MHRF11b] en outre une utilisation de cette répétition majeure comme alternative à l'amélioration calculatoire de l'estimation de la similarité musicale. Dans le cadre de la recherche de reprises, nous montrons ainsi que la répétition majeure permet de conserver une précision élevée pour l'identification des reprises, avec une perte d'environ 10% de la précision du système, tout en proposant un gain d'un facteur 8 environ sur le temps de calcul pratique. Cette évaluation montre donc que l'identification automatique de reprises peut être effectuée à partir d'une simple partie des signaux.

Nous introduisons notre propre formalisation du problème de structuration de la musique sous une forme hiérarchique, et nous proposons [MHR+10] un algorithme d'inférence de structuration des répétitions. Nous montrons sur quelques exemples que notre algorithme est capable d'identifier des arbres de descriptions structurelles correspondant à des règles de compositions musicales bien-fondées. De plus, nous présentons une évaluation de notre algorithme par rapport à l'état de l'art qui témoigne de résultats prometteurs de notre technique pour la détection de segmentations structurelles. Ces résultats soulignent ainsi la pertinence de notre méthode à la fois pour répondre au problème existant d'analyse structurelle, mais aussi pour l'identification d'une structuration hiérarchique de la musique.

## 6.2 Perspectives

Les travaux décrits dans ce document couvrent plusieurs études successives qui suggèrent chacune de nombreuses perspectives. Dans la suite, nous décrivons les principaux axes de recherches futures en indiquant, le cas échéant, les résultats préliminaires associés à chacun d'entre eux.

### Amélioration du calcul de similarité musicale

Le Chapitre 2 décrit une méthode de représentation de la musique sous la forme de séquences tonales. Nous justifions ce choix dans le cadre de cette thèse par une volonté de circonscrire le problème et de l'étudier sur un unique critère musical. La littérature en analyse structurelle [PMK10] suggère néanmoins que l'utilisation combinée de différents descripteurs audio peut être bénéfique, notamment à la détection des frontières structurelles. L'utilisation d'un autre critère tel que l'information timbrale ou rythmique, conjointement à l'information tonale, constitue donc une perspective de ces travaux. Pour la combinaison des critères, une technique peut consister à définir pour la similarité locale un schéma de scores de pondération prenant en compte les symboles de deux séquences, chacune représentant un critère musical distinct.

Plusieurs améliorations des outils de comparaison présentés en Chapitre 3 sont envisageables. D'abord, il est nécessaire d'étudier l'impact de la prise en compte d'opérations d'édition supplémentaires, qui permettent de traduire des correspondances musicales particulières. Par exemple, on peut considérer l'introduction des opérations de *fragmentation* et de *consolidation* telles que définies par Mongeau et Sankoff [MS90] dans le cadre de représentations musicales symboliques, ou encore les opérations de *compression* et *expansion* dans le cadre de représentations audio [KL99]. De telles opérations permettent de donner une signification musicale plus forte à la comparaison séquentielle calculée, en améliorant par exemple la robustesse de l'estimation de similarité face à des variations locales de tempo. Le gain en précision des comparaisons à effectuer reste donc à démontrer. La définition des scores de pondération peut également être améliorée afin de correspondre au contenu musical comparé. Par exemple, notre formalisation introduit des coûts d'insertion, de substitution ou encore de transposition constants. Les travaux futurs peuvent se concentrer sur l'étude de fonctions de coûts affines voire plus complexes, ou encore de coûts de transposition dépendant de l'ampleur des décalages locaux effectués.

L'étude d'indexation BLAST présentée en Chapitre 3 est à notre connaissance la première approche de cette méthode heuristique pour indexer l'audio musical. En conséquence, elle débouche sur un ensemble de perspectives faisant écho à des travaux de comparaison de séquences biologiques qui définissent des raffinements des heuristiques effectuées par BLAST. En particulier, nos premiers tests indiquent qu'il semble judicieux de considérer des graines espacées [MTL02] afin d'augmenter la sensibilité de la détection ; toutefois, la disposition optimale et le nombre d'espaces idéal dans les graines restent des facteurs à déterminer. Une autre perspective consiste à utiliser des graines de taille variable [Csű04] afin d'indexer l'espace de

recherche en ne considérant que les transitions musicales les plus significatives d'un morceau de musique.

### Reconstruction audio avancée

La méthode de reconstruction de données manquantes présentée en Chapitre 4 est un autre axe majeur de travaux futurs et en cours. En particulier, il semble intéressant d'étendre la reconstruction en permettant qu'un extrait manquant soit reconstruit par un ensemble de courts extraits placés en série. Nous avons publié dans [BJM12] une première modélisation de cette alternative en utilisant une algèbre spécifique pour la combinaison de plusieurs extraits musicaux. Un algorithme d'optimisation du choix des extraits utilisés pour la reconstruction reste alors à définir.

Le système de reconstruction peut également être adapté pour une application en temps réel. Dans ce cas, on suppose que le contenu audio est reçu de manière progressive (on parle alors de *flux* entrant) et que l'on n'a connaissance à chaque instant que d'une partie du signal. Le problème consiste à poursuivre la lecture de son en cas d'incohérence musicale sur le flux entrant (rupture de flux), en générant un signal *espéré* [Hur06] et garantissant une certaine continuité musicale [Con08, Pac02]. Notre étude préliminaire de ce problème suggère qu'il est possible d'utiliser le résultat d'un alignement local optimal pour identifier le meilleur segment, permettant ainsi de générer une suite perceptivement cohérente, à l'image de [Jeh05a]. Une première implémentation montre que la méthode semble être robuste au passage au temps réel dans une certaine limite sur la durée de signal reçu. La formalisation précise de cette solution et son évaluation perceptive restent à établir.

### Analyse de structures répétitives spécifiques

Outre les répétitions présentées dans le Chapitre 4, plusieurs structurations spécifiques proches de celles étudiées n'ont pas été formalisées dans cette thèse, et leur étude approfondie est laissée en perspective. En particulier, on peut s'intéresser au problème de détection de la "meilleure répétition" d'une taille fixée, ou encore définir les problèmes d'identification de structures répétitives en relâchant la contrainte de disjonction, afin de prendre en compte certaines formes musicales particulières (telles que les fugues).

Le Chapitre 5 décrit l'inférence de structures répétitives sans *a priori*. Néanmoins, le problème de l'inférence d'une forme musicale peut être défini avec un ensemble d'hypothèses sur les morceaux comparés. En particulier, une étude a été commencée sur un ensemble de morceaux à la *forme strophique*, ou *stanzas*, qui désignent des pièces constituées de répétitions multiples d'un seul et unique motif. Les séquences représentatives possèdent alors une propriété dite de *cyclicité*, étudiée en algorithmique du texte (voir par exemple [GT96, CLS+06]). Les premières évaluations de l'inférence de formes strophiques ont été menées sur un corpus de chansons traditionnelles néerlandaises[1], et semblent produire des résultats prometteurs au vu des études existantes [MGW09, BM12, MG12].

1. http://www.liederenbank.nl

### Amélioration du modèle hiérarchique

Le modèle hiérarchique introduit en Chapitre 5 fait l'objet de nombreux travaux en cours. Notre modèle semble présenter des limitations sur certains morceaux. Il est donc nécessaire de caractériser plus précisément l'aspect hiérarchique des structures musicales, et de déterminer l'ensemble des œuvres concernées par une telle représentation.

En outre, il semble important dans le cadre de travaux futurs de pouvoir évaluer la validité de l'ensemble de la hiérarchie identifiée, et donc de disposer d'une vérité terrain arborescente. Une solution envisagée consiste à considérer des arborescences d'harmonies, tels que définies par exemple par [Roc11] ou [MdH11].

Afin de comparer deux hiérarchies, il convient de définir une métrique spécifique permettant d'estimer la similarité entre les motifs estimés et les motifs annotés malgré des différences de niveaux hiérarchiques. Un point de départ d'une telle métrique est suggéré par Chai dans [Cha05, 58–62], même si la comparaison est effectuée à partir de séquences et non d'une hiérarchie complète.

Enfin, si l'évaluation du système d'inférence de structures répétitives met en avant sa bonne précision par rapport à l'état de l'art, la considération de la seule structure tonale répétitive impose une limitation aux résultats. Dans le cas où les similarités et dissimilarités entre motifs structurels sont liées à des critères autres que la répétition ou l'information tonale, il est nécessaire de combiner notre approche avec une vue complémentaire de la structure musicale prenant en compte ces critères. Une telle combinaison devra faire l'objet d'une étude approfondie afin de tenter de réunir les qualités de plusieurs approches parmi les nombreuses techniques existantes pour l'analyse des répétitions musicales.

# Publications

### Conférences internationales avec comité de lecture et publication des actes

Benjamin Martin, Daniel G. Brown, Pierre Hanna, Pascal Ferraro, BLAST for audio sequences alignment: a fast scalable cover identification tool, *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 529–534, 2012

Florent Berthaut, David Janin, Benjamin Martin, Advanced synchronization of audio or symbolic musical patterns: an algebraic approach, *Proc. of the 6th IEEE International Conference on Semantic Computing (ICSC)*, pages 302–309, 2012

Benjamin Martin, Pierre Hanna, Matthias Robine, Pascal Ferraro, Towards an indexing method to speed-up music retrieval, *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1167–1168, 2011

Benjamin Martin, Pierre Hanna, Matthias Robine, Pascal Ferraro, Indexing musical pieces using their major repetition, *Proc. of the 11th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 153–156, 2011

Benjamin Martin, Pierre Hanna, Vinh-Thong Ta, Myriam Desainte-Catherine, Exemplar-based assignment of large missing audio parts using string matching on tonal features. *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 507–5012, 2011

Benjamin Martin, Matthias Robine, Pierre Hanna, Musical structure retrieval by aligning self-similarity matrices, *Proc. of the 10th International Society for Music Information Retrieval (ISMIR)*, pages 483–488, 2009

### Revues internationales

Benjamin Martin, Pierre Hanna, Matthias Robine, Pascal Ferraro, Major repetition in musical audio : indexing for content-based retrieval systems, Article soumis en cours de révision

Benjamin Martin, Pierre Hanna, Matthias Robine, Pascal Ferraro, Structural analysis of audio music using string matching techniques, Article soumis en cours de révision

### Brevet international

Julien Allali, Myriam Desainte-Catherine, Pascal Ferraro, Pierre Hanna, Benjamin Martin, Matthias Robine, Vinh-Thong Ta, A process for assigning audio data in missing audio parts of a music piece and device for performing the same, PCT IB 2012 055673, 2012

### Campagnes d'évaluation internationales

Benjamin Martin, Pierre Hanna, Matthias Robine, Julien Allali, Pascal Ferraro, Structural analysis of harmonic features using string matching techniques, Music Information Retrieval Evaluation eXchange (MIREX) - Structural Segmentation task, 2010, 2011, 2012

Benjamin Martin, Pierre Hanna, Matthias Robine, Julien Allali, Pascal Ferraro, String matching cover song detection algorithm, Music Information Retrieval Evaluation eXchange (MIREX) - Cover Song Identification task, 2010

# Bibliographie

[AAF+09] J. Allali, P. Antoniou, P. Ferraro, C.S. Iliopoulos, and S. Michalakopoulos. Overlay problems for music and combinatorics. In *Proc. of the 15th International Conference on Auditory Display (ICAD)*, 2009. *Cité p. 112 et 116*

[ABSW04] N.H. Adams, M.A. Bartsch, J.B. Shifrin, and G.H. Wakefield. Time series alignment for music information retrieval. In *Proc. of the 5th International Society for Music Information Retrieval Conference (ISMIR)*, pages 303–311, 2004. *Cité p. 35*

[AEJ+11] A. Adler, V. Emiya, M.G. Jafari, M. Elad, R. Gribonval, and M.D. Plumbley. Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):922–932, 2011. *Cité p. 74 et 75*

[AFHI07] J. Allali, P. Ferraro, P. Hanna, and C. Iliopoulos. Local transpositions in alignment of polyphonic musical sequences. In *String Processing and Information Retrieval*, volume 4726, pages 26–38. 2007. *Cité p. 47 et 50*

[AGM+90] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. *Cité p. 56 et 58*

[AMS+97] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997. *Cité p. 60 et 61*

[ANS+05] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes. Theory and evaluation of a bayesian music structure extractor. In *Proc. of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, pages 420–425, 2005. *Cité p. 110, 111 et 133*

[AP02] J.J. Aucouturier and F. Pachet. Finding songs that sound the same. In *Proc. of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, pages 1–8, 2002. *Cité p. 10*

[APS05] J.J. Aucouturier, F. Pachet, and M. Sandler. The way it sounds: Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Transactions on Multimedia*, 7:1028–1035, 2005. *Cité p. 110 et 111*

[AS02] J.J. Aucouturier and M. Sandler. Finding repeating patterns in acoustic musical signals : Applications for audio thumbnailing. In *Audio Engineering Society 22nd Virtual, Synthetic, and Entertainment Audio International Conference*, 2002. *Cité p. 12, 109, 110 et 114*

[Ash01] M. Ashikhmin. Synthesizing natural textures. In *Proc. of the ACM Symposium on Interactive 3D graphics*, pages 217–226, 2001. *Cité p. 12 et 76*

[Auc06] J.J. Aucouturier. *Dix expériences sur la modélisation du timbre polyphonique*. PhD thesis, University Paris VI, 2006. *Cité p. 18 et 19*

[BCL10] L. Barrington, A.B. Chan, and G. Lanckriet. Modeling music as a dynamic texture. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):602–612, 2010. *Cité p. 110 et 111*

[BDSV11] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent. Methodology and resources for the structural segmentation of music pieces into autonomous and comparable blocks. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 287–292, 2011. *Cité p. 113 et 114*

[BE47] H.S. Black and J.O. Edson. Pulse code modulation. *Transactions of the American Institute of Electrical Engineers*, 66(1):895–899, 1947. *Cité p. 22*

[Bel07] J.P. Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proc. of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, pages 239–244, 2007. *Cité p. 52*

[Bel11] J.P. Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013–2025, 2011. *Cité p. 12*

[Ben95] G. Benson. A space efficient algorithm for finding the best nonoverlapping alignment score. *Theoretical Computer Science*, 145(1):357–369, 1995. *Cité p. 91*

[Bit87] M. Bitsch. *Précis d'harmonie tonale.* A. Leduc, 1987. *Cité p. 27, 62 et 66*

[BJM12] F. Berthaut, D. Janin, and B. Martin. Advanced synchronization of audio or symbolic musical patterns: An algebraic approach. In *Proc. of the 6th IEEE International Conference on Semantic Computing*, pages 302–309, 2012. *Cité p. 105 et 146*

[BLBSV10] F. Bimbot, O. Le Blouch, G. Sargent, and E. Vincent. Decomposition into autonomous and comparable blocks : a structural description of music pieces. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 189–194, 2010. *Cité p. 113, 114 et 130*

[BM12] C. Bohak and M. Marolt. Finding repeating stanzas in folk songs. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 451–456, 2012. *Cité p. 146*

[BME11] T. Bertin-Mahieux and D.P.W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 117–120, 2011. *Cité p. 68*

[BMEWL11] T. Bertin-Mahieux, D.P.W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–596, 2011. *Cité p. 63*

[BMK06] M.J. Bruderer, M. McKinney, and A.G. Kohlrausch. Structural boundary perception in popular music. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 198–201, 2006. *Cité p. 113*

[BP05] J.P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proc. of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, pages 304–311, 2005. *Cité p. 28*

[BPMW12] J.P. Bello, Grosche P., M. Müller, and R.J. Weiss. Analyzing and visualizing repetitive structures in music recordings. *ACM Computers in Entertainment*, 2012. A paraître. *Cité p. 12*

[Bre94] A.S. Bregman. *Auditory scene analysis: The perceptual organization of sound*. The MIT Press, 1994. *Cité p. 14 et 17*

[BS09] B. Benward and M. Saker. *Music in Theory and Practice*, volume 2. McGraw-Hill, 2009. *Cité p. 119*

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of the 27th ACM annual conference on Computer Graphics and Interactive Techniques*, pages 417–424, 2000. *Cité p. 76*

[BW01] M.A. Bartsch and G.H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 15–18, 2001. *Cité p. 27*

[BW05] M.A. Bartsch and G.H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, 2005. *Cité p. 12, 90, 108, 109, 110 et 111*

[CBKH05] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, 41(3):271–284, 2005. *Cité p. 9*

[CCI+02] E. Cambouropoulos, M. Crochemore, C.S. Iliopoulos, L. Mouchard, and Y. Pinzon. Algorithms for computing approximate repetitions in musical sequences. *International Journal of Computer Mathematics*, 79:1135–1148, 2002. *Cité p. 35*

[CE10] C.V. Cotton and D.P.W. Ellis. Audio fingerprinting to identify multiple videos of an event. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 2386–2389, 2010. *Cité p. 9*

[CF04] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 127–130, 2004. *Cité p. 90, 109, 110 et 111*

[Cha03] W. Chai. Structural analysis of musical signals via pattern matching. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 549–552, 2003. *Cité p. 110 et 112*

[Cha05] W. Chai. *Automated analysis of musical structure*. PhD thesis, Massachusetts Institute of Technology, 2005. *Cité p. 97, 109, 113, 114, 119, 132 et 147*

[CHL07] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on strings*. Cambridge University Press, 2007. *Cité p. 36*

[CI04] R. Clifford and C.S. Iliopoulos. String algorithms in music analysis. *Soft Computing*, 8(9):597–603, 2004. *Cité p. 35*

[CIR98] T. Crawford, C. S. Iliopoulos, and R. Raman. String matching techniques for musical similarity and melodic recognition. In *Proc. of the 7th ACM International Conference on Multimedia*, volume 11, pages 71–100, 1998. *Cité p. 14*

[CL11] R. Chen and M. Li. Music structural segmentation by combining harmonic and timbral information. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 477–481, 2011. *Cité p. 110 et 111*

[CLS+06] H.L. Chan, T.W. Lam, W.K. Sung, S.L. Tam, and S.S. Wong. A linear size index for approximate pattern matching. *Journal of Discrete Algorithms*, 9(4):358–364, 2006. *Cité p. 146*

[Con08] A. Cont. *Modeling Musical Anticipation: From the time of music to the music of time*. PhD thesis, University of Paris 6 and University of California in San Diego, 2008. *Cité p. 15 et 146*

[CPT04] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004. *Cité p. 76 et 78*

[Cro80] R. Crochiere. A weighted overlap-add method of short-time fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(1):99–102, 1980. *Cité p. 80*

[CS06] M. Casey and M. Slaney. The importance of sequences in musical similarity. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, page V, 2006. *Cité p. 14*

[Csű04] M. Csűrös. Performing local similarity searches with variable length seeds. In *Combinatorial Pattern Matching*, Lecture Notes in Computer Science, pages 373–387. 2004. *Cité p. 145*

[CT65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics and Computation*, 19(90):297–301, 1965. *Cité p. 25*

[CVG+08] M.A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. In *Proc. of the IEEE*, volume 96, pages 668–696, 2008. *Cité p. 5, 6, 7 et 17*

[DBEJ08] J.S. Downie, M. Bay, A.F. Ehmann, and M.C. Jones. Audio cover song identification: Mirex 2006-2007 results and analyses. In *Proc. of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, pages 468–473, 2008. *Cité p. 51, 52 et 54*

[DBP+07] R.B. Dannenberg, W.P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis. A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701, 2007. *Cité p. 9 et 15*

[DCWS03] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. *Cité p. 111*

[DEBJ10] J.S. Downie, Andreas Ehmann, Mert Bay, and M. Jones. The music information retrieval evaluation exchange: Some observations

and insights. In *Advances in Music Information Retrieval*, volume 274 of *Studies in Computational Intelligence*, pages 93–115. 2010. *Cité p. 130*

[Deu82] D. Deutsch. *The processing of pitch combinations.* Academic Press, 1982. *Cité p. 27*

[DG09] R.B. Dannenberg and M. Goto. Music structure analysis from acoustic signals. In *Handbook of Signal Processing in Acoustics*, pages 305–331. 2009. *Cité p. 13 et 113*

[DH02] R.B. Dannenberg and N. Hu. Pattern discovery techniques for music audio. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, pages 63–70, 2002. *Cité p. 35, 109, 110 et 112*

[Dig12] DigitalMusicNews. *Drowning? The iTunes Store Now Has 28 Million Songs...*, Avril 2012. http://www.digitalmusicnews.com/permalink/2012/120425itunes. *Cité p. 5*

[Don09] O. Donnat. Les pratiques culturelles des français à l'ère numérique. *Culture études*, 1(5):1–12, 2009. *Cité p. 5*

[Dow03] J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37:295–340, 2003. *Cité p. 6, 8, 10 et 19*

[Dow08] J.S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008. *Cité p. 54, 130 et 139*

[DW05] S. Dixon and G. Widmer. Match: A music alignment tool chest. In *Proc. of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, pages 492–497, 2005. *Cité p. 35*

[EKR87] J.P. Eckmann, S.O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 4:973, 1987. *Cité p. 108*

[EL99] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of the 7th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038, 1999. *Cité p. 12 et 76*

[EMD+11] A.F. Ehmann, M.Bay, J.S. Downie, I. Fujinaga, and D. De Roure. Music structure segmentation algorithm evaluation: Expanding on mirex 2010 analyses and datasets. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 561–566, 2011. *Cité p. 130, 131, 138, 139 et 140*

[Eme98] E. Emery. *Temps et musique.* L'age D'homme, 1998. *Cité p. 12 et 17*

[EP07] D.P.W. Ellis and G.E. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 1429–1432. IEEE, 2007. *Cité p. 28, 30, 32, 52 et 68*

[Eri75] R. Erickson. *Sound structure in music.* University of California Press, 1975. *Cité p. 17 et 115*

[Esc88] F. Escal. Le thème en musique classique. *Communications*, 47(1):93–117, 1988. *Cité p. 11, 115 et 116*

[ET07] A. Eronen and F. Tampere. Chorus detection with combined use of mfcc and chroma features and image processing filters. In *Proc. of 10th International Conference on Digital Audio Effects (DAFx)*, pages 229–236, 2007. *Cité p. 109, 110, 111 et 112*

[Ett96] W. Etter. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. *IEEE Transactions on Signal Processing*, 44(5):1124–1135, 1996. *Cité p. 74*

[EVRK03] P.A.A. Esquef, V. Välimäki, K. Roth, and I. Kauppinen. Interpolation of long gaps in audio signals using the warped burg's method. In *Proc. of the 6th International Conference on Digital Audio Effects (DAFx)*, pages 08–11, 2003. *Cité p. 75*

[EWBL02] D.P.W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, volume 2, pages 170–177, 2002. *Cité p. 7*

[Fer99] A.J.S. Ferreira. An odd-dft based approach to time-scale expansion of audio signals. *IEEE Transactions on Speech and Audio Processing*, 7(4):441–453, 1999. *Cité p. 80*

[Foo99] J. Foote. Visualizing music and audio using self-similarity. In *Proc. of the 7th ACM International Conference on Multimedia*, pages 77–80, 1999. *Cité p. 90, 107 et 108*

[Foo00] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 452–455, 2000. *Cité p. 109, 110 et 111*

[Fre06] A. Freed. Music metadata quality: A multiyear case study using the music of skip james. In *Proc. of the Audio Engineering Society Convention*, 2006. *Cité p. 7*

[Fuj99] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proc. of the International Computer Music Conference (ICMC)*, pages 464–467, 1999. *Cité p. 27 et 28*

[GÓ6] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, pages 63–100, 2006. *Cité p. 19, 27, 28, 30, 31, 50, 52, 62 et 63*

[GH06] E. Gómez and P. Herrera. The song remains the same: Identifying versions of the same piece using tonal descriptors. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 180–185, 2006. *Cité p. 52*

[GHNO02] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, pages 287–288, 2002. *Cité p. 130*

[GML03] S. Gao, N.C. Maddage, and C.H. Lee. A hidden markov model based approach to music segmentation and identification. In *Proc. of the 4th*

*International Conference on Information, Communications and Signal Processing*, volume 3, pages 1576–1580. IEEE, 2003. *Cité p. 111*

[GMS12] P. Grosche, M. Müller, and J. Serrà. Audio content-based music retrieval. In *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 151–174. 2012. *Cité p. 9*

[GOH06] E. Gómez, BS Ong, and P. Herrera. Automatic tonal analysis from music summaries for version identification. *Audio Engineering Society Convention (AES)*, 2006. *Cité p. 13, 100 et 101*

[Góm06] E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006. *Cité p. 28*

[Got03] M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 437–440, 2003. *Cité p. 11, 12, 90, 100 et 108*

[Got06] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1783–1794, 2006. *Cité p. 28, 90, 108, 109, 110, 111 et 112*

[GR98] S.J. Godsill and P.J.W. Rayner. *Digital Audio Restoration-a statistical model based approach*. Springer, 1998. *Cité p. 25, 74 et 75*

[GSMA12] P. Grosche, J. Serrà, M. Müller, and J.L. Arcos. Structure-based audio fingerprinting for music retrieval. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 55–60, 2012. *Cité p. 12*

[GT96] J. Gregor and M.G. Thomason. Efficient dynamic programming alignment of cyclic strings by shift elimination. *Pattern Recognition*, 29(7):1179–1185, 1996. *Cité p. 146*

[Gus97] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, pages 215–253, 1997. *Cité p. 14, 36, 37, 40, 42, 43, 44 et 50*

[GYF+11] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano. Songle: A web service for active music listening improved by user contributions. *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 311–316, 2011. *Cité p. 12*

[HAE03] J. Herre, E. Allamanche, and C. Ertel. How similar do songs sound? towards modeling human perception of musical similarity. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 83–86, 2003. *Cité p. 10*

[Ham50] R.W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. *Cité p. 115*

[Hay95] S. Haykin. *Advances in spectrum analysis and array processing*, volume 3. Prentice-Hall, 1995. *Cité p. 26*

[HBPD03] P. Herrera-Boyer, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003. *Cité p. 9*

[HCC04] J.L. Hsu, A.L.P. Chen, and H.P. Chen. Finding approximate repeating patterns from sequence data. In *Proc. of the 5th International Society for Music Information Retrieval Conference (ISMIR)*, 2004. *Cité p. 35*

[HFR07] P. Hanna, P. Ferraro, and M. Robine. On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences. *Journal of New Music Research*, 36(4):267–279, 2007. *Cité p. 35*

[Hir75] D.S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975. *Cité p. 47*

[HS05] C.A. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. In *Proc. of the Audio Engineering Society Convention (AES)*, pages 291–301, 2005. *Cité p. 28*

[HSG06] C. Harte, M. Sandler, and M. Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM workshop on audio and music computing multimedia*, pages 21–26, 2006. *Cité p. 27*

[Hur06] D.B. Huron. *Sweet anticipation: Music and the psychology of expectation.* The MIT Press, 2006. *Cité p. 12, 17, 20 et 146*

[IT94] ITU-T. A method for subjective performance assessment of the quality of speech voice output devices. *International Telecommunication Union Recommandation*, page 85, 1994. *Cité p. 84 et 85*

[JCEJ08] J.H. Jensen, M.G. Christensen, D.P.W. Ellis, and S.H. Jensen. A tempo-insensitive distance measure for cover song identification based on chroma features. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2209–2212, 2008. *Cité p. 52*

[Jeh05a] T. Jehan. *Creating Music by Listening.* PhD thesis, Massachusetts Institute of Technology, pages 57–59, 2005. *Cité p. 10, 12, 18, 32, 64, 68, 76, 77, 110, 112, 114 et 146*

[Jeh05b] T. Jehan. Hierarchical multi-class self similarities. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 311–314, 2005. *Cité p. 114*

[Jen07] K. Jensen. Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Advances in Signal Processing*, 2007(1):159–159, 2007. *Cité p. 109, 110 et 111*

[KAC05] B. Kater, EHL Aarts, and I.R.A.W. Clout. Music based light effects. *Master's Thesis. Technische Universiteit Eindhoven*, 2005. *Cité p. 12*

[KAS11] F. Kaiser, M.G. Arvanitidou, and T. Sikora. Audio similarity matrices enhancement in an image processing framework. In *9th International Workshop on content-mased multimedia indexing (CBMI)*, pages 67–72, 2011. *Cité p. 109*

[KD06] A. Klapuri and M. Davy. *Signal processing methods for music transcription.* Springer, 2006. *Cité p. 19 et 31*

[KH04] J. Kilian and H.H. Hoos. Musicblast - gapped sequence alignment for mir. In *Proc. of the 5th International Conference on Music Information Retrieval*, pages 38–41, 2004. *Cité p. 57*

[Kil04] J.F. Kilian. *Inferring Score Level Musical Information From Low-Level Musical Data*. PhD thesis, Darmstadt University of Technology, pages 54–60, 2004. *Cité p. 57*

[KL99] J.B. Kruskal and M. Liberman. The symmetric time-warping problem: from continuous to discrete. In *Time Warps, String Edits, and Macromolecules - The Theory and Practice of Sequence Comparison*, chapter 4. CSLI Publications, 1999. *Cité p. 145*

[KM93] S. Kannan and E.W. Myers. An algorithm for locating non-overlapping regions of maximum alignment score. In *Proc. of the 4th Annual Symposium on Combinatorial Pattern Matching*, pages 74–86, 1993. *Cité p. 91 et 95*

[KM08] F. Kurth and M. Müller. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, 2008. *Cité p. 52 et 63*

[KN08] S. Kim and S. Narayanan. Dynamic chroma feature vectors with applications to cover song identification. In *10th IEEE Workshop on Multimedia Signal Processing*, pages 984–987, 2008. *Cité p. 52*

[Kru01] C.L. Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, USA, 2001. *Cité p. 17, 23 et 32*

[Kru04] C.L. Krumhansl. The cognition of tonality–as we know it today. *Journal of new music research*, 33(3):253–268, 2004. *Cité p. 19 et 27*

[KS10] F. Kaiser and T. Sikora. Music structure discovery in popular music using non-negative matrix factorization. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 429–434, 2010. *Cité p. 110, 111 et 114*

[KSM+10] Y.E. Kim, E.M. Schmidt, R. Migneco, B.G. Morton, P. Richardson, J. Scott, J.A. Speck, and D. Turnbull. Music emotion recognition: A state of the art review. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 255–266, 2010. *Cité p. 7 et 10*

[KYB03] I. Korf, M. Yandell, and J. Bedell. *Blast*. O'Reilly & Associates, 2003. *Cité p. 56*

[Lab] FX Palo Alto Laboratory. Self-similarity analysis: a selected bibilography: http://www.fxpal.com/?p=similaritybib. *Cité p. 108*

[Lar98] C. Larkin. *The encyclopedia of popular music*. Macmillan, 1998. *Cité p. 50*

[LC00] B. Logan and S. Chu. Music summarization using key phrases. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 749–752, 2000. *Cité p. 12, 90, 110 et 111*

[Lem00] K. Lemström. *String matching techniques for music retrieval*. PhD thesis, University of Helsinki, 2000. *Cité p. 35*

[Lev66] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. *Cité p. 37 et 40*

[Lev08] D.J. Levitin. *This is your brain on music: Understanding a human obsession*. Penguin Publishing, 2008. *Cité p. 10 et 51*

[LJ96] F. Lerdahl and R. Jackendoff. *A generative theory of tonal music.* MIT Press, 1996. *Cité p. 14, 17, 18, 20, 115 et 116*

[LLZ06] L. Lu, D. Liu, and H.J. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):5–18, 2006. *Cité p. 10*

[LMR05] M. Lagrange, S. Marchand, and J.B. Rault. Long interpolation of audio signals using linear prediction in sinusoidal modeling. *Journal of the Audio Engineering Society*, 53(10):891–905, 2005. *Cité p. 74 et 75*

[LRKO+08] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama. Computational auditory induction by missing-data non-negative matrix factorization. *ISCA Tutorial and Research Workshop on Statistical And Perceptual Audition (SAPA)*, 2008. *Cité p. 74 et 75*

[LS+99] D.D. Lee, H.S. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. *Cité p. 111*

[LS08] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):318–326, 2008. *Cité p. 110 et 132*

[Luk08] H.M. Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proc. of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, pages 375–380, 2008. *Cité p. 131, 132 et 133*

[LWZ04a] L. Lu, M. Wang, and H.J. Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 275–282, 2004. *Cité p. 108, 109, 110 et 111*

[LWZ04b] L. Lu, L. Wenyin, and H.J. Zhang. Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, 12(2):156–167, 2004. *Cité p. 12, 76 et 77*

[Mah94] R.C. Maher. A method for extrapolation of missing digital audio data. *Journal of the Audio Engineering Society*, 42(5):350–357, 1994. *Cité p. 75*

[Man83] B.B. Mandelbrot. The fractal geometry of nature. *W. H. Freeman and Co.*, 1983. *Cité p. 107*

[Mar06] M. Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 280–285, 2006. *Cité p. 52, 109, 110 et 112*

[MBHF12] B. Martin, D.G. Brown, P. Hanna, and P. Ferraro. Blast for audio sequences alignment: a fast scalable cover identification tool. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 529–534, 2012. *Cité p. 57, 59, 62 et 143*

[MC90] E. Moulines and F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5):453–467, 1990. *Cité p. 80*

[MCD+09] M. Mauch, C. Cannam, M. Davies, C.Harte, S. Kolozali, D. Tidhar, and M. Sandler. Omras2 metadata project 2009. In *10th International*

*Conference on Music Information Retrieval (ISMIR), Late-Breaking Session*, 2009. *Cité p. 82, 95 et 130*

[MdH11] J.P. Magalhães and W.B. de Haas. Functional modelling of musical harmony: an experience report. In *Proc. of the 16th ACM SIGPLAN International Conference on Functional Programming*, volume 46, pages 156–162, 2011. *Cité p. 147*

[ME10] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010. *Cité p. 31*

[MEKR11] M. Müller, D. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing*, (99), 2011. *Cité p. 17, 18, 19 et 30*

[MG12] M. Müller and P. Grosche. Automated segmentation of folk song field recordings. In *Proc. of the 10th IEEE ITG Symposium on Speech Communication*, pages 1–4. VDE VERLAG GmbH, 2012. *Cité p. 115 et 146*

[MGW09] M. Müller, P. Grosche, and F. Wiering. Robust segmentation and annotation of folk song recordings. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740, 2009. *Cité p. 146*

[MHR+10] B. Martin, P. Hanna, M. Robine, J. Allali, and P. Ferraro. Structural analysis of harmonic features using string matching techniques. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2010. *Cité p. 110, 131 et 144*

[MHRF11a] B. Martin, P. Hanna, M. Robine, and P. Ferraro. Indexing musical pieces using their major repetition. In *Proc. of the 11th ACM/IEEE Joint Conference on Digital Libraries*, pages 153–156, Ottawa, Canada, June 2011. *Cité p. 52, 73, 95 et 144*

[MHRF11b] B. Martin, P. Hanna, M. Robine, and P. Ferraro. Towards an indexing method to speed-up music retrieval. In *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1167–1168, 2011. *Cité p. 52, 73, 100 et 144*

[MHT+11] B. Martin, P. Hanna, V.T. Ta, M. Desainte-Catherine, and P. Ferraro. Exemplar-based assignment of large missing audio parts using string matching on tonal features. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 507–512, 2011. *Cité p. 73, 77 et 144*

[Mid99] R. Middleton. *"Form", Key Terms in Popular Music and Culture*. Wiley-Blackwell, 1999. *Cité p. 10*

[Mil92] W. Miller. An algorithm for locating a repeated region, 1992. Unpublished Manuscript. *Cité p. 91, 93, 94 et 95*

[MK07] M. Müller and F. Kurth. Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Applied Signal Processing*, 1(1):163–184, 2007. *Cité p. 12, 109, 110, 111, 112 et 114*

[MKC05] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proc. of the 6th International Society*

*for Music Information Retrieval Conference (ISMIR)*, pages 288–295, 2005. *Cité p. 52*

[ML81] J.V. Maizel and R.P. Lenk. Enhanced graphic matrix analysis of nucleic acid and protein sequences. In *Proc. of the National Academy of Sciences*, volume 78, pages 7665–7669, 1981. *Cité p. 108*

[MM98] S. Masnou and J.M. Morel. Level lines based disocclusion. In *Proc. of International Conference on Image Processing (ICIP)*, pages 259–263, 1998. *Cité p. 76*

[MND09] M. Mauch, K.C. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 231–236, 2009. *Cité p. 13, 109, 110, 112 et 131*

[MRH09] B. Martin, M. Robine, and P. Hanna. Musical structure retrieval by aligning self-similarity matrices. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 483–488, 2009. *Cité p. 12*

[MRS08] C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008. *Cité p. 55, 102, 131 et 132*

[MS90] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990. *Cité p. 37 et 145*

[MTL02] B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002. *Cité p. 71 et 145*

[Mül07] M. Müller. *Information retrieval for music and motion*. Springer-Verlag, 2007. *Cité p. 23, 35, 112, 113 et 114*

[Mül11] M. Müller. New developments in music information retrieval. In *Proc. of the 42nd Audio and Engineering Society Conference*, pages 11–20, 2011. *Cité p. 107*

[MXK06] N.C. Maddage, C. Xu, and M.S. Kankanhalli. Automatic structure detection for popular music. *IEEE MultiMedia*, 13:65–77, 2006. *Cité p. 110*

[Nic12] A. Nicolas. Etat des lieux de l'offre de musique numérique. Observatoire de la musique, 2012. *Cité p. 5*

[NKM02] H. Nagano, K. Kashino, and H. Murase. Fast music retrieval using polyphonic binary feature vectors. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 101–104, 2002. *Cité p. 52*

[NW70] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. *Cité p. 40*

[Ong07] B. Ong. *Structural Analysis and Segmentation of Music Signals*. PhD thesis, Universitat Pompeu Fabra, 2007. *Cité p. 10, 108, 109, 110, 111 et 112*

[Ori06] N. Orio. *Music retrieval: A tutorial and review*, volume 1 of *Foundations and Trends in Information Retrieval*. Now Publishers, 2006. *Cité p. 8*

[Oud10] L. Oudre. *Template-based chord recognition from audio signals*. PhD thesis, Télécom ParisTech, 2010. *Cité p. 28*

[Pac02] F. Pachet. The continuator: Musical interaction with style. In *Proc. of International Computer Music Conference (ICMC)*, pages 211–218, 2002. *Cité p. 146*

[Par94] R. Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11:409–464, 1994. *Cité p. 17*

[Par06a] B. Pardo. Finding structure in audio for music information retrieval. *IEEE Signal Processing Magazine*, 23(3):126–132, 2006. *Cité p. 8*

[Par06b] B. Pardo. Music information retrieval. *Communications of the ACM*, 49(8):29–31, 2006. *Cité p. 6*

[Pau10] J. Paulus. *Signal Processing Methods for Drum Transcription and Music Structure Analysis*. PhD thesis, Tampere University of Technology, 2010. *Cité p. 107, 108 et 114*

[PBO00] H. Purwins, B. Blankertz, and K. Obermayer. A new method for tracking modulations in tonal music in audio data format. In *Proc. of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 6, pages 270–275, 2000. *Cité p. 28*

[PBR02] G. Peeters, A. La Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, pages 94–100, 2002. *Cité p. 111 et 131*

[PD09] G. Peeters and E. Deruty. Is music structure annotation multi-dimensional? a proposal for robust local music annotation. In *Proc. of the 3rd Workshop on Learning the Semantics of Audio Signals (LSAS)*, pages 75–90, Graz, Austria, 2009. *Cité p. 84, 90, 91, 107, 113, 114, 130 et 131*

[Pee04] G. Peeters. Deriving musical structures from signal analysis for music audio summary generation: Sequence and state approach. In *Computer Music Modeling and Retrieval*, volume 2771 of *Lecture Notes in Computer Science*, pages 169–185. Springer, 2004. *Cité p. 13, 109, 110 et 131*

[Pee06] G. Peeters. Musical key estimation of audio signal based on hidden markov modeling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 127–131, 2006. *Cité p. 28*

[Pee07] G. Peeters. Sequence representation of music structure using higer-order similarity matrix and maximum likelihood approach. In *Proc. of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007. *Cité p. 109, 110, 111, 112 et 131*

[PK06] J. Paulus and A. Klapuri. Music structure analysis by finding repeated parts. In *Proc. of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM)*, pages 59–68, 2006. *Cité p. 110 et 132*

[PK09] J. Paulus and A. Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions*

*on Audio, Speech and Language Processing*, 17(6):1159–1170, 2009. *Cité p. 110, 111, 112 et 113*

[PK12] G. Peeters and F. Karën. Towards a (better) definition of the description of annotated mir corpora. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 25–30, 2012. *Cité p. 113*

[PLR08] E. Peiszer, T. Lidy, and A. Rauber. Automatic audio segmentation: Segment boundary and structure detection in popular music. In *Proc. of International Workshop on Learning the Semantics of Audio Signals (LSAS)*, 2008. *Cité p. 112*

[PMK10] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–636, Utrecht, Netherlands, 2010. *Cité p. 107, 109, 111, 113, 114, 132 et 145*

[PQ11] A. Pires and M. Queiroz. Real-time unsupervised music structural segmentation using dynamic descriptors. In *Proc. of the Sound and Music Conference (SMC)*, 2011. *Cité p. 111*

[PS01] E. Pollastri and G. Simoncelli. Classification of melodies by composer with hidden markov models. In *Proc. of the 1st International Conference on Web Delivering of Music*, pages 88–95, 2001. *Cité p. 9*

[Pur05] H. Purwins. *Profiles of Pitch Classes - Circularity of Relative Pitch and Key - Experiments, Models, Music Analysis, and Perspectives.* PhD thesis, Technischen Universität Berlin, 2005. *Cité p. 27*

[PWL01] F. Pachet, G. Westermann, and D. Laigre. Musical data mining for electronic music distribution. In *Proc. of the 1st International Conference on Web Delivering of Music*, pages 101–106, 2001. *Cité p. 7*

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–286, 1989. *Cité p. 110 et 111*

[Rao04] V.M. Rao. Audio compression using repetitive structures in music. Master's thesis, University of Miami, 2004. *Cité p. 13 et 111*

[RC07] C. Rhodes and M. Casey. Algorithms for determining and labelling approximate hierarchical self-similarity. In *Proc. of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, pages 41–46, 2007. *Cité p. 110, 112 et 115*

[Ric06] J.A. Rice. *Mathematical statistics and data analysis.* Thomson Learning, 2006. *Cité p. 103*

[RN88] J.L. Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. *American Statistician*, pages 59–66, 1988. *Cité p. 32*

[RN01] D. Reefman and P. Nuijten. Why direct stream digital is the best choice as a digital audio format. *Proc. of Audio Engineering Society Convention (AES)*, 2001. *Cité p. 22*

[Roc11] T. Rocher. *Analyse et modélisation des informations tonales dans la musique occidentale.* PhD thesis, Université Bordeaux 1, 2011. *Cité p. 13, 62, 66 et 147*

[SBF+11] J.B.L. Smith, J.A. Burgoyne, I. Fujinaga, D. De Roure, and J.S. Downie. Design and creation of a large-scale database of structural annotations. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 555–560, 2011. *Cité p. 114 et 130*

[SBV10] G. Sargent, F. Bimbot, and E. Vincent. A structural segmentation of songs using generalized likelihood ratio under regularity assumptions. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2010. *Cité p. 110, 111 et 131*

[SBV+11] G. Sargent, F. Bimbot, E. Vincent, et al. A regularity-constrained viterbi algorithm and its application to the structural segmentation of songs. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 483–488, 2011. *Cité p. 110 et 111*

[Sch95] J.P. Schmidt. All shortest paths in weighted grid graphs and its application to finding all approximate repeats in strings. In *Proc. of the 3rd Israel Symposium on theTheory of Computing and Systems*, pages 67–77, 1995. *Cité p. 95*

[Sch12] D. Schnitzer. *Indexing Content-Based Music Similarity Models for Fast Retrieval in Massive Databases*. PhD thesis, Johannes Kepler Universität, 2012. *Cité p. 56*

[SD06] C. Sailer and K. Dressler. Finding cover songs by melodic similarity. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2006. *Cité p. 52*

[SGH08] J. Serrà, E. Gómez, and P. Herrera. Transposing chroma representations to a common key. In *Proc. of IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, pages 45–48, 2008. *Cité p. 33*

[SGH10] J. Serrà, E. Gómez, and P. Herrera. *Audio cover song identification and similarity: background, approaches, evaluation, and beyond*, volume 274 of *Studies in Computational Intelligence*, chapter 14, pages 307–332. 2010. *Cité p. 50, 51, 52, 54 et 55*

[SGHS08] J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1138–1151, 2008. *Cité p. 30, 31, 32, 33, 51, 52, 53 et 68*

[She82] R.N. Shepard. Structural representations of musical pitch. *The psychology of music*, pages 343–390, 1982. *Cité p. 27*

[SHOB11] A. Sprotte-Hansen, J. Orpinel, and J.P. Bello. The music intervisualizer–integrating navigation and visualization. *13th International Society for Music Information Retrieval Conference (ISMIR) - Late-breaking session*, 2011. *Cité p. 115*

[SJK05] Y. Shiu, H. Jeong, and C.C.J. Kuo. Musical structure analysis using similarity matrix and dynamic programming. In *Proc. of SPIE*, volume 6015, pages 398–409, 2005. *Cité p. 110 et 112*

[SK09] M. Schedl and P. Knees. Context-based music similarity estimation. In *Proc. of the 3rd International Workshop on Learning Semantics of Audio Signals*, page 59, 2009. *Cité p. 7*

[SL01] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001. *Cité p. 110*

[Smi10] J.B.L. Smith. *A comparison and evaluation of approaches to the automatic formal analysis of musical audio*. PhD thesis, McGill University, 2010. *Cité p. 112*

[SPA07] A. Spanias, T. Painter, and V. Atto. *Audio signal processing and coding*. Wiley & Sons, 2007. *Cité p. 24*

[Spo12] SpotifyLtd. *Spotify Metadata API*, Accédé en Août 2012. http://ws.spotify.com/search/1/track?q=year:0-9999. *Cité p. 5*

[SR99] D. Schwarz and X. Rodet. Spectral envelope estimation and representation for sound analysis-synthesis. In *Proc. of the International Computer Music Conference (ICMC)*, 1999. *Cité p. 31*

[SRS10] P. Smaragdis, B. Raj, and M. Shashanka. Missing data imputation for time-frequency representations of audio signals. *Journal of Signal Processing Systems*, 65(3):1–10, 2010. *Cité p. 75*

[SSA09] J. Serra, X. Serra, and R.G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009. *Cité p. 52*

[SSSE00] A. Schödl, R. Szeliski, D.H. Salesin, and I. Essa. Video textures. In *Proc. of the 27th annual conference on Computer Graphics and Interactive Techniques*, pages 489–498, 2000. *Cité p. 12*

[Ste79] L. Stein. *Structure & style: the study and analysis of musical forms*. Suzuki, 1979. *Cité p. 11 et 20*

[SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. *Cité p. 41, 42 et 92*

[TB02] B. Tillmann and E. Bigand. A comparative review of priming effects in language and music. *Advances In Consciousness Research*, 35:231–240, 2002. *Cité p. 19*

[TC99] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organised sound*, 4(3):169–175, 1999. *Cité p. 12*

[TC02] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002. *Cité p. 9*

[TD05] D. Tschumperle and R. Deriche. Vector-valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 506–517, 2005. *Cité p. 76*

[Tem04] D. Temperley. *The Cognition of Basic Musical Structures*. The MIT Press, 2004. *Cité p. 17 et 19*

[Ter60] USA Standard Acoustical Terminology. Timbre, 1960. *Cité p. 18*

[TL07] D. Turnbull and G. Lanckriet. A supervised approach for detecting boundaries in music using difference features and boosting. In *Proc. of*

*the 5th International Society for Music Information Retrieval Conference (ISMIR)*, pages 42–49, 2007. *Cité p. 131 et 132*

[TLX+09] A. Tian, W. Li, L. Xiao, D. Wang, J. Zhou, and T. Zhang. Histogram matching for music repetition detection. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 662–665, 2009. *Cité p. 109 et 110*

[TP80] J. Tenney and L. Polansky. Temporal gestalt perception in music. *Journal of Music Theory*, 24(2):205–241, 1980. *Cité p. 17*

[TYW08] W.H. Tsai, H.M. Yu, and H.M. Wang. Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *Journal of Information Science and Engineering*, 24(6):1669–1687, 2008. *Cité p. 50*

[VHP02] H. Vinet, P. Herrera, and F. Pachet. The cuidado project. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, 2002. *Cité p. 9*

[Vin05] H. Vinet. The semantic hifi project. *Proc. of International Conference on Music Computing (ICMC)*, 2005. *Cité p. 11*

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. *Cité p. 111*

[VV05] M. Viswanathan and M. Viswanathan. Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (mos) scale. *Computer Speech & Language*, 19(1):55–83, 2005. *Cité p. 85*

[Wan84] C.C. Wang. Effects of some aspects of rhythm on tempo perception. *Journal of Research in Music Education*, 32(3):169–176, 1984. *Cité p. 87*

[Wan03] A. Wang. An industrial strength audio search algorithm. In *Proc. of the 4th International Society for Music Information Retrieval Conference (ISMIR)*, volume 2, 2003. *Cité p. 9*

[WB10] R. J. Weiss and J. P. Bello. Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 123–128, 2010. *Cité p. 110, 111, 114 et 131*

[WB11] R. Weiss and J. Bello. Unsupervised discovery of temporal structure in music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1240–1251, 2011. *Cité p. 111*

[WCB10] A. Wankhammer, I.V.L. Clarkson, and A.P. Bradley. Music structure discovery based on normalized cross correlation. In *Proc. of the 13th International Conference on Digital Audio Effects (DAFx-10)*, pages 488–493, 2010. *Cité p. 109, 110 et 111*

[WCL09] L. Wang, E.S. Chng, and H. Li. Efficient sparse self-similarity matrix construction for repeating sequence detection. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 458–461, 2009. *Cité p. 109 et 110*

[Wei99] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 975–982, 1999. *Cité p. 111*

[WH03] J. Wellhausen and M. Hoeynck. Audio thumbnailing using mpeg-7 low-level audio descriptors. In *Internet Multimedia Management Systems IV*, volume 5242, pages 65–73, 2003. *Cité p. 109, 110, 111 et 112*

[Whi60] B.W. White. Recognition of distorted melodies. *The American journal of psychology*, 73(1):100–107, 1960. *Cité p. 51*

[WL02] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proc. of International Computer Music Conference (ICMC)*, pages 591–598, 2002. *Cité p. 7*

[Wri07] C.M. Wright. *Listening to music.* Schirmer Books, 2007. *Cité p. 128 et 129*

[Yan01] C. Yang. Music database retrieval based on spectral similarity. *Proc. of the 2nd International Society for Music Information Retrieval Conference (ISMIR)*, 2001. *Cité p. 52*

[ZKG05] Y. Zhu, M.S. Kankanhalli, and S. Gao. Music key detection for musical audio. In *Proc. of the 11th International Multimedia Modelling Conference (MMM)*, pages 30–37, 2005. *Cité p. 28*