Master 2 Acoustique, Traitement du signal, Informatique, Appliqués à la Musique

Mémoire de stage

21 août 2025

Détection automatique de motifs répétitifs dans des données musicales en utilisant la morphologie mathématique

Mathys Daniel

Période de stage : du 03/03/2025 au 31/08/2025

Sujet du stage : non confidentiel

Paul Lascabettes
Post-doctorant (IRMA)
paul.lascabettes@math.unistra.fr

Moreno Andreatta
Directeur de recherches (IRMA, CNRS)
moreno.andreatta@math.unistra.fr

Isabelle Bloch
Professeure des universités (LIP6, Sorbonne Université)
isabelle.bloch@sorbonne-universite.fr

Institut de recherche mathématique avancée (IRMA), Université de Strasbourg

Laboratoire d'Informatique de Sorbonne Université (LIP6 UMR 7606), Sorbonne Université, Paris

Résumé

L'identification automatique de motifs répétitifs dans des données musicales symboliques représente un défi fondamental dans le domaine de la musicologie computationnelle. Le travail présenté ici explore une approche originale basée sur la morphologie mathématique, un cadre initialement développé pour le traitement d'images, afin d'analyser des structures musicales représentées sous forme d'ensembles de points dans un espace en deux dimensions

temps/hauteur. L'objectif de cette étude est double : premièrement, concevoir de nouveaux algorithmes capables de détecter des motifs répétitifs dans des fichiers MIDI à l'aide d'opérations morphologiques telles que l'érosion et l'ouverture ; et deuxièmement, utiliser ces motifs pour déduire des informations musicales telles que la durée des motifs, les signatures rythmiques et les occurrences de structures polymétriques. Ce rapport détaille point par point les étapes de développement des travaux : du prétraitement des données à l'extraction des motifs, en passant par l'interprétation structurelle. De plus, la méthode fournit des représentations compactes et interprétables musicalement, facilitant la lisibilité de ces résultats. Le cadre proposé offre des avancées tant théoriques que pratiques dans ce domaine, permettant ainsi une analyse robuste et efficace des stuctures rythmiques dans la musique polyphonique.

Mots clés : **Détection de motifs musicaux**, **Morphologie mathématique**, **Analyse de musique symbolique**, **Musicologie computationnelle**

Abstract

The automatic identification of repetitive patterns in symbolic music data represents a fundamental challenge in the field of computational musicology. The present work explores an original approach based on mathematical morphology, a framework initially developed for image processing, to analyse musical structures in a two dimensions space time/pitch. The objective of this study is dual : first, to design new algorithms capable of detecting repeated patterns in MIDI files using morphological operations such as erosion and opening ; and secondly, to use these patterns to deduce higher-level musical information, with a particular focus on rhythmic signatures and instances of polymetric structure. This report details each stage of the development process : from data pre-processing to pattern extraction and structural interpretation. In addition, the method provides compact and musically interpretable representations, facilitating the readability of these results. The proposed framework offers both theoretical and practical advances in the field, thereby enabling robust and efficient analysis of rhythmic structures in polyphonic music.

Keywords : **Musical pattern discovery**, **Mathematical morphology**, **Symbolic music analysis**, **Computational musicology**

Table des matières

1 Introduction **3**

2 État de l'art **4**
2.1 Contexte scientifique . . . 4
2.1.1 Représentation symbolique de la musique . . . 4
2.1.2 Détection de motifs musicaux . . . 5
2.2 Algorithmes existants . . . 5
2.2.1 SIA : Structure Induction Algorithm . . . 6
2.2.2 SIATEC : Structure Induction Algorithm with Translation Equivalence Classes . . . 6
2.2.3 COSIATEC : Compression Optimised Structure Induction Algorithm with Translation Equivalence Classes . . . 6
2.2.4 Résumé : . . . 7
2.3 Morphologie mathématique et musique . . . 8
2.3.1 Introduction à la morphologie mathématique . . . 8
2.3.2 Lien entre détection de motifs musicaux et morphologie mathématique . . . 9
2.3.3 Signature rythmique et polymétrie . . . 11
2.4 Objectifs du stage . . . 12

3 Prétraitement des données musicales **15**
3.1 De la musique à un nuage de points 2D . . . 15
3.2 Choix de la base de données . . . 16
3.3 Traitements des données . . . 16
3.3.1 Séparation et nettoyage . . . 16
3.3.2 Conversion temporelle et quantification . . . 17

4 Détection de motifs musicaux via la morphologie mathématique **19**
4.1 Premier algorithme : répétitions avec nombre d'occurrences fixe . . . 19
4.1.1 Étapes de l'algorithme . . . 19
4.1.2 Résultats . . . 20
4.1.3 Limites . . . 21
4.2 Second algorithme : extraction des sous-motifs pertinents . . . 21
4.2.1 Étapes de l'algorithme . . . 23
4.2.2 Définitions et propriétés . . . 23
4.2.3 Résultats . . . 25
4.3 Comparaison avec les algorithmes existants . . . 25

5 Extraction d'informations musicales **28**
5.1 Détection automatique de signatures rythmiques . . . 28
5.1.1 Critères . . . 29
5.1.2 Résultats . . . 30
5.2 Répartition métrique . . . 32
5.2.1 Méthode de calcul . . . 32
5.2.2 Résultats . . . 33

6 Conclusion et perspectives **34**

Bibliographie **35**

Annexes **37**

Chapitre 1

Introduction

L'analyse des structures répétitives occupe une place importante dans la musicologie computationnelle. L'identification automatique des motifs qui composent une œuvre musicale permet non seulement de mieux comprendre son organisation interne, mais aussi d'alimenter un ensemble d'applications, allant de la transcription automatique à la génération de contenu musical, en passant par des outils d'analyse et de pédagogie musicale.

Ce mémoire s'inscrit à la croisée de deux domaines : les mathématiques appliquées et la musique. Il explore une idée originale : transposer les outils de la morphologie mathématique, historiquement développés pour le traitement d'images, dans le cadre de l'analyse musicale symbolique. L'intuition sous-jacente est que, de la même manière qu'une image discrète, une pièce musicale peut être modélisée comme un ensemble de points dans un espace discrétisé (temps/hauteur) où les motifs répétitifs deviennent des sous-ensembles de points.

L'objectif principal de ce travail est double. D'une part, il s'agit de développer des algorithmes capables de détecter automatiquement des motifs dans des données musicales symboliques (fichiers MIDI), en s'appuyant sur des opérateurs morphologiques simples, tels que l'érosion et l'ouverture. D'autre part, ces motifs sont analysés afin d'en déduire des informations structurelles, en particulier des signatures rythmiques locales. Ce type d'analyse permet notamment d'identifier des situations de polymétrie, où plusieurs signatures rythmiques co-existent simultanément.

Pour ce faire, une chaîne de traitement complète a été mise en œuvre, allant du nettoyage des données à la visualisation des résultats : constitution d'un corpus adapté, transformation des fichiers MIDI en nuages de points exploitables, définition d'éléments structurants permettant de détecter des répétitions, développement d'algorithmes de détection de motifs, sélection de motifs pertinents selon des critères musicaux, puis extraction de signatures rythmiques.

Au-delà des aspects techniques, ce mémoire a pour but de présenter les travaux menés afin de vérifier la validité de résultats théoriques récents et d'utiliser ces idées pour introduire de nouveaux outils d'analyse musicale. L'ensemble des méthodes présentées vise à obtenir des résultats plus interprétables et plus efficaces sur le plan computationnel, ouvrant ainsi la voie à une analyse automatique plus fine des structures rythmiques de la musique.

Chapitre 2

État de l'art

2.1 Contexte scientifique

Ce stage s'inscrit dans un domaine de recherche à l'intersection de la **morphologie mathématique** et de l'**analyse musicale automatique**. L'objectif général est d'explorer comment des outils issus du traitement d'images peuvent être adaptés pour l'étude de structures musicales symboliques.

Le travail est mené sous la direction de **Moreno Andreatta** (IRMA, Université de Strasbourg, CNRS) qui travaille sur les liens entre mathématiques et musique ; **Isabelle Bloch** (LIP6, Sorbonne Université) spécialisée en traitement d'images et en particulier sur les outils de morphologie mathématique ; et **Paul Lascabettes** (IRMA, Université de Strasbourg), dont les recherches portent en partie sur l'utilisation d'opérateurs morphologiques pour détecter des structures répétitives dans des données symboliques musicales. En particulier, ce stage vise à appliquer, tester et affiner les méthodes théoriques développées dans sa thèse de doctorat [1], en les confrontant à un **corpus réel** soigneusement préparé.

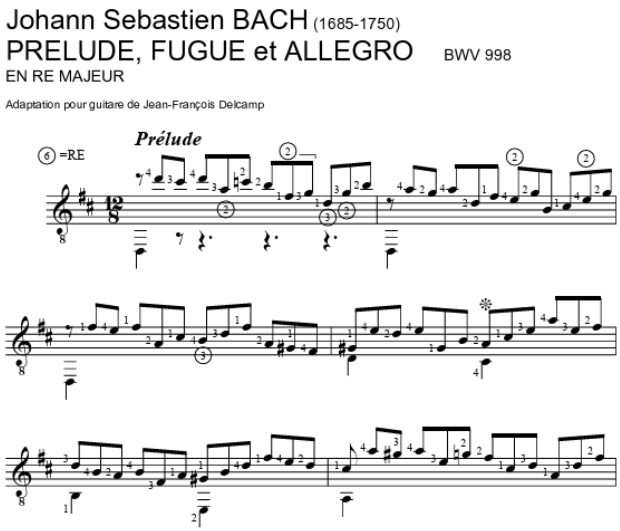
2.1.1 Représentation symbolique de la musique

La musique peut être représentée sous différentes formes, la plus intuitive étant le format audio (mp3, wav...), mais elle n'est pas toujours la plus adaptée à tout type de tâche. Une autre manière de représenter la musique est la **représentation symbolique** (typiquement utilisée dans les fichiers MIDI), où chaque événement musical (note ou silence) est décrit par son *instant d'activation* (onset), sa *hauteur* (pitch), sa *durée*, ou des métadonnées comme l'instrument auquel l'évènement appartient.

Cette représentation considère qu'un morceau de musique peut être représenté comme un sous-ensemble fini X de $\mathbb{R}^n$ (ou $\mathbb{Z}^n$). Chaque point de X représente alors une note, et ses coordonnées en donnent les éléments caractéristiques, chaque axe représentant une information sur la note. On peut typiquement avoir un ensemble de points dans $\mathbb{R}^2$ avec un axe représentant les activations temporelles, et l'autre représentant les hauteurs.

Dans cette représentation, nous n'avons pas l'information de timbre, contrairement à un fichier audio par exemple. Ainsi, cette représentation pourrait s'apparenter à une partition de musique. La figure 2.1 nous permet de visualiser une telle représentation.

Cette représentation discrète est particulièrement adaptée pour des traitements mathématiques car elle permet de modéliser les événements musicaux sous forme d'un ensemble fini de points dans un espace vectoriel.



$$\Big\{(0,38),(1,74),(2,73),(3,74),(4,69),$$
$$(5,72),(6,71),(7,66),(8,67),(9,62),$$
$$\ldots \qquad \ldots \qquad \ldots$$
$$(68,61),(69,62),(70,57),(71,62),(72,66)\Big\}$$

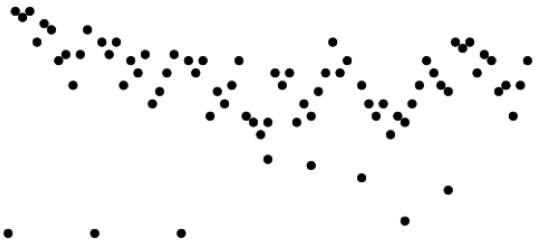


FIGURE 2.1 – Trois représentations d'un exrait du Prélude BWV998 de Bach, arrangé pour guitare classique. En haut : partition usuelle. À gauche : représentation comme sous-ensemble de $\mathbb{R}^2$ (instant d'apparition $\times$ hauteur MIDI). À droite : nuage de points dans $\mathbb{R}^2$ (figure issue de [2]).

2.1.2 Détection de motifs musicaux

De nombreux musicologues montrent que l'identification perceptuelle de répétitions est une composante essentielle pour qu'un auditeur puisse appréhender une pièce musicale [3, 4, 5]. Heinrich Schenker affirme, par exemple, que la répétition est «*la base de la musique en tant qu'art*» [6]. La détection de motifs répétitifs dans un contexte musical semble donc être particulièrement adaptée pour mieux appréhender une pièce musicale, un artiste ou un genre de musique [7]. En effet, dès lors que nous avons des répétitions, on peut chercher ce qui se répète. Ainsi, nous appelons un ensemble de points qui se répète un motif. Plus formellement, on décrit un motif comme «un ensemble de paires "onset/pitch" qui apparaissent au moins deux fois (i.e. est répété au moins une fois) dans la pièce musicale» [8]. On peut alors s'interroger sur la pertinence relative de certains motifs pour représenter une œuvre musicale donnée : quels motifs sont les plus adaptés, à quels moments apparaissent-ils, et avec quelle fréquence ? L'analyse peut également porter sur les informations musicologiques que ces motifs permettent de mettre en évidence.

2.2 Algorithmes existants

C'est avec tant de questions ouvertes que des études sont menées depuis plus de 20 ans afin de développer des algorithmes de détection de motifs spécifiquement dans un contexte musical. Nous allons ici détailler trois parmi les principaux.

2.2.1 SIA : Structure Induction Algorithm

L'algorithme **SIA** (« Structure Induction Algorithm ») [3] est l'un des premiers outils proposés pour la découverte automatique de motifs dans des données musicales symboliques en représentation multidimensionnelle (i.e. ensembles de points). Il repose sur l'idée centrale qu'une pièce musicale est définie par les **motifs** qui la composent ainsi que par les **translations** qui permettent de retrouver ces motifs dans la pièce musicale.

Principe de fonctionnement : SIA renvoie tous les sous-ensembles de points **MTP** [1] (« Maximal Translatable Patterns ») $P \subset X$ (où X est le nuage de points représentant la pièce) qui sont reliés par une **translation commune**. Autrement dit, SIA détermine l'ensemble de tous les plus grands sous-ensembles admettant une occurrence translatée exacte, ainsi que le vecteur de translation associé à chaque motif.

Les étapes de SIA sont les suivantes :

1. Pour chaque paire de points (x_i, x_j), on calcule le vecteur de translation $t = x_j - x_i$.
2. Pour ce vecteur t, on détermine le plus grand sous-ensemble de points de X qui reste entièrement inclus dans X lorsqu'il est translaté par t.
3. Les ensembles obtenus sont appelés des **Maximal Translatable Patterns** (MTPs).

Interprétation musicale : L'algorithme SIA permet d'identifier l'ensemble des **répétitions maximales** présentes dans une pièce musicale, sous la forme de **copies exactes translatées**. Autrement dit, il détecte les segments musicaux répétés à un autre emplacement, conservant strictement les mêmes intervalles temporels et intervalles de hauteurs. Cela inclut, par exemple, un motif reproduit à l'identique plus loin dans l'œuvre. En revanche, les répétitions avec variations (comme une transposition mélodique ou une modification rythmique) ne sont pas détectées.

2.2.2 SIATEC : Structure Induction Algorithm with Translation Equivalence Classes

L'algorithme **SIATEC** [3] est une extension de SIA qui introduit la notion de **classe d'équivalence par translation** (TEC : *Translation Equivalence Class*), ce qui permet de trouver les occurrences des motifs obtenus avec SIA.

Principe de fonctionnement :

1. Comme dans SIA, les motifs translatés (**MTPs**) sont d'abord détectés.
2. Ces motifs sont ensuites regroupés en classes d'équivalence : une **TEC** est constituée d'un motif de base P et de l'ensemble des vecteurs de translation $T = \{t_1, t_2, ..., t_k\}$ qui permettent d'obtenir toutes ses occurrences dans le morceau.

Chaque classe TEC est ainsi une description compacte de **ce qui se répète** et **où cela se répète**.

Interprétation musicale : L'algorithme SIATEC permet d'identifier toutes les occurrences d'un même motif répétitif et fournit ainsi une description structurée de la répétition, précisant quels motifs sont répétés, combien de fois, et où se situent ces occurrences dans la pièce. Une classe TEC extraite par SIATEC correspond à une **famille d'occurrences d'un même motif**, ce qui permet, d'un point de vue musicologique, d'obtenir une vision globale des répétitions au sein de l'œuvre.

2.2.3 COSIATEC : Compression Optimised Structure Induction Algorithm with Translation Equivalence Classes

COSIATEC [9] est une extension de SIATEC, proposée par Meredith en 2013, qui introduit une nouvelle approche : la recherche de motifs doit être orientée par un critère de **compacité**. Cela repose sur l'idée que l'analyse musicale peut être vue comme une **compression des données symboliques** :

> *« The best ways of understanding a piece of music are those that are represented by the shortest descriptions of the piece. » [9]*

1. Un motif MTP d'une translation v dans X est un sous-ensemble de points de X qui reste inclus dans X lorsqu'il est translaté par v.

Cette extension est justifiée par le fait que lorsqu'on applique SIATEC sur un morceau, on peut obtenir un grand nombre de motifs redondants ou partiellement superposés. En effet, plusieurs milliers de motifs sont découverts avec SIATEC alors qu'un musicologue en aurait retenu quelques dizaines tout au plus. Cela peut rendre l'interprétation difficile : quels motifs sont réellement significatifs ? Lesquels sont les plus représentatifs de la pièce musicale ?

L'objectif de COSIATEC n'est alors plus seulement d'extraire toutes les classes d'équivalence par translation (TEC), mais de trouver une représentation **minimale et efficace** de la structure répétitive d'une pièce musicale. Pour cela est introduit un score permettant de classer les motifs détéctés. À la fin de l'algorithme, chaque point du morceau X appartient à exactement une occurrence d'un motif sélectionné. Nous n'avons donc pas de chevauchement entre les TECs qui ont été gardés. Si $\mathcal{M} = \{(P_1, T_1), \ldots, (P_k, T_k)\}$ est l'ensemble des motifs retenus, on a alors :

$$X = \bigcup_{i=1}^{k} \bigcup_{t \in T_i} (P_i + t), \quad \text{et} \quad (P_i + t) \cap (P_j + t') = \emptyset \text{ si } i \neq j \text{ ou } t \neq t'$$

Principe de fonctionnement :

1. L'ensemble des TECs possibles est d'abord généré (comme dans SIATEC).
2. Une fois un TEC choisi (celui avec le meilleur score de compacité), les points qu'il couvre sont retirés, puis l'algorithme recommence la sélection sur les points restants.
3. Le processus s'arrête lorsque tous les points du morceau sont couverts.

Meredith propose de sélectionner les motifs maximisant la fonction suivante :

$$\text{Compacité}(P, T) = \frac{|P| \times |T|}{|P| + |T| - 1}$$

où :
- P est le motif de base (un ensemble de points),
- $|P|$ est le nombre de points dans le motif,
- T est l'ensemble des vecteurs de translation associés aux occurrences,
- $|T|$ est le nombre d'occurrences (copies translatées du motif).

Plus cette valeur est grande, plus le motif est considéré comme **compact** : il couvre beaucoup de points avec une description courte.

Interprétation musicale : COSIATEC permet d'obtenir une **carte des motifs musicaux** qui résume la structure répétitive d'une pièce avec un compromis optimal entre le nombre de motifs, leur taille et leur nombre d'occurrences.

Cela correspond à une **compression des données musicales** : l'algorithme extrait les motifs qui permettent de décrire le plus efficacement possible les répétitions significatives d'un morceau.

2.2.4 Résumé :

- **SIA** : recherche des motifs maximaux translatés (MTPs),
- **SIATEC** : regroupe ces motifs en classes d'équivalence par translation (TECs),
- **COSIATEC** : sélectionne un sous-ensemble de TECs pour compresser les données musicales de manière optimale en supprimant la redondance.

Ces approches constituent le socle historique des méthodes de découverte de motifs symboliques. Ces trois algorithmes nous servent donc de références, tant en matière de méthodes proposées qu'en matière de résultats auxquels nous pouvons nous comparer.

2.3 Morphologie mathématique et musique

2.3.1 Introduction à la morphologie mathématique

Nous définirons nos outils dans le cadre spécifique de notre étude : un ensemble fini de points $X \subset E$, où E est un espace comme $\mathbb{R}^n$ ou $\mathbb{Z}^n$ [10] [11], représentant une pièce musicale sous forme symbolique. Nous nous plaçons dans le cas de transformations invariantes par translations, pouvant être définies à partir d'un élément structurant.

L'une des deux opérations fondamentales de la morphologie mathématique est la dilatation.

Définition 2.1 : Dilatation

Soient $X \subset E$ un ensemble de points, et $S \subset E$ un autre ensemble de points, appelé élément structurant. La dilatation d'un ensemble X par S est définie par :

$$\delta_S(X) = X \oplus S = \{x + s \mid x \in X,\ s \in S\}$$

Cela revient à translater l'élément structurant S sur chaque point de X, puis à prendre la réunion de toutes ces copies translatées. Cela produit un «agrandissement» de X selon la forme de S. On dit donc que la dilatation est un opérateur **génératif**.

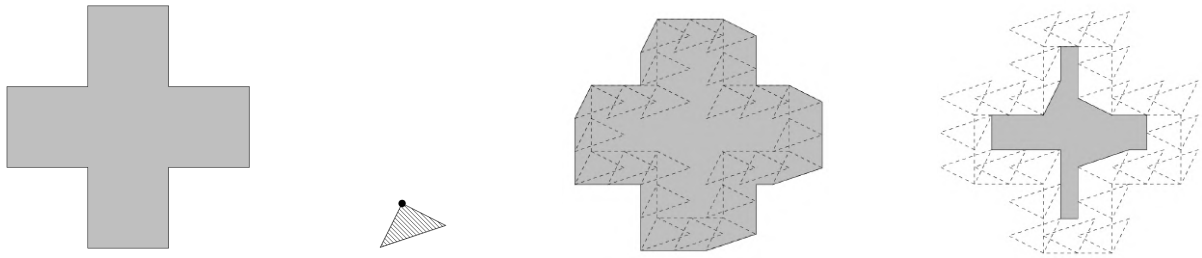
L'autre opération fondamentale de la morphologie mathématique est l'érosion, que nous définissons de la façon suivante :

Définition 2.2 : Érosion

Soient $X \subset E$ un ensemble de points, et $S \subset E$ un autre ensemble de points. L'érosion d'un ensemble X par l'élément structurant S est définie par :

$$\varepsilon_S(X) = X \ominus S = \{x \in E \mid S_x \subset X\} \quad \text{, où } S_x = \{s + x \mid s \in S\}.$$

L'érosion sélectionne les points de X où la translation de S en ce point est entièrement incluse dans X. Cela permet d'extraire les positions dans X où une occurrence exacte du motif S est présente. Cela produit un «affinement» de X selon la forme de S. On dit donc que l'érosion est un opérateur **analytique**. La figure 2.2 montre comment visualiser ces deux opérations :



(a) Ensemble X (b) Élément structurant S (c) Dilatation $\delta_S(X)$ (d) Érosion $\varepsilon_S(X)$

FIGURE 2.2 – Illustration de l'érosion et de la dilatation morphologique de X par S (figure issue de [10]).

On peut donc maintenant définir d'autres opérations morphologiques pouvant être obtenues par composition de ces opérations de base. Nous nous intéresserons ici aux deux les plus directes.

Définition 2.3 : Ouverture

Soient $X \subset E$ un ensemble de points, et $S \subset E$ un autre ensemble de points. L'ouverture d'un ensemble X par l'élément structurant S est définie par :

$$\gamma_S(X) = (X \ominus S) \oplus S = \delta_S(\varepsilon_S(X))$$

L'ouverture élimine les structures plus petites que S (au sens de l'inclusion) et régularise la forme de X.

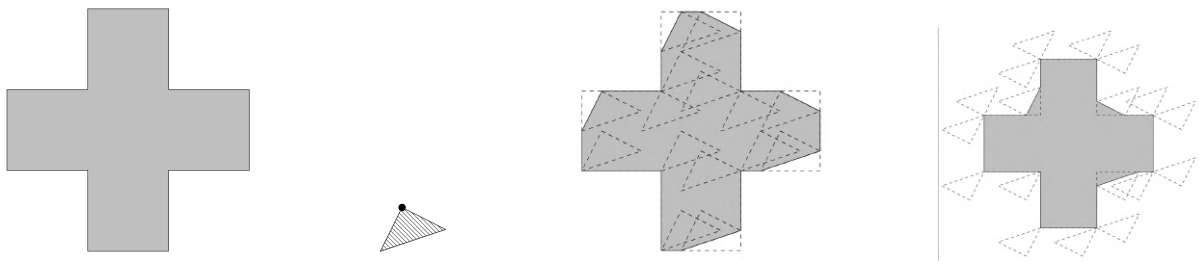
Définition 2.4 : Fermerture

Soient $X \subset E$ un ensemble de points, et $S \subset E$ un autre ensemble de points. L'ouverture d'un ensemble X par l'élément structurant S est définie par :

$$\varphi_S(X) = (X \oplus S) \ominus S = \varepsilon_S(\delta_S(X))$$

L'ouverture comble les trous plus petits que S (au sens de l'inclusion) et «colle» les composantes connexes proches.

La figure 2.3 montre comment de telles opérations peuvent être visualisées :



(a) Ensemble X (b) Élément structurant S (c) Ouverture $\gamma_S(X)$ (d) Fermeture $\varphi_S(X)$

FIGURE 2.3 – Illustration de l'ouverture et de la fermeture morphologiques de X par S (figure issue de [10]).

La découverte automatique de motifs musicaux pouvant être formalisée comme une recherche d'occurrences de l'élément structurant S dans X, l'érosion et l'ouverture permettent d'extraire toutes les occurrences exactes d'un motif donné, tandis que la dilatation et la fermeture peuvent être utilisées pour générer ou enrichir des motifs. C'est pourquoi il peut être intéressant de se réapproprier les algorithmes existants sous ce formalisme et d'en développer de nouveaux spécifiquement en partant de ces outils.

2.3.2 Lien entre détection de motifs musicaux et morphologie mathématique

C'est dans ce contexte que Lascabettes propose dans sa thèse [1] une **réinterprétation des algorithmes SIA et SIATEC à l'aide des outils de la morphologie mathématique**. Son idée principale consiste à remplacer la recherche exhaustive de translations par des opérations morphologiques simples, mieux adaptées à la structure des données musicales symboliques.

Ce formalisme permet de manipuler les données musicales de manière géométrique. Pour cela, on introduit un **élément structurant** $S \subset E$, qui représente une structure musicale locale que l'on souhaite détecter ou manipuler. Ce motif S peut par exemple être un rythme, un intervalle mélodique, un accord ou bien un motif. Les opérations de morphologie mathématique s'interprètent alors dans ce contexte musical de la façon suivante :

Opération	Interprétation musicale
Érosion $\varepsilon_S(X)$	Permet de détecter toutes les positions où le motif S apparaît exactement dans l'ensemble X.
Dilatation $\delta_S(X)$	Permet de générer des occurrences potentielles du motif S en l'appliquant sur l'ensemble X.
Ouverture $\gamma_S(X) = \delta_S(\varepsilon_S(X))$	Permet d'extraire toutes les occurrences exactes du motif S présentes dans X.

Cette lecture morphologique permet ainsi de donner un nouvel éclairage aux algorithmes développés par Meredith pour la découverte de motifs musicaux, en les exprimant au moyen d'opérateurs géométriques simples et robustes. En particulier, voici comment ont été ré-interprétés SIA et SIATEC. Le détail des preuves se trouve dans [1].

SIA La recherche du MTP revient simplement à calculer une érosion. Soient $x_1, x_2 \in X$ et $\{x_1, x_2\}$ l'élément structurant avec son origine en x_1. On a :

$$MTP(x_2 - x_1, X) = \varepsilon_{\{x_1,x_2\}}(X)$$

Autrement dit, les occurrences du motif $P = \{x_1, x_2\}$ translaté par $x_2 - x_1$ peuvent être directement obtenues par une érosion de X par cet élément structurant. L'algorithme SIA peut donc être interprété morphologiquement comme l'ensemble des érosions de X par des motifs à deux points. Cela permet de reformuler l'algorithme SIA sous forme morphologique comme suit :

$$\text{SIA} : \left\{ \left\langle x_2 - x_1,\ \varepsilon_{\{x_1,x_2\}}(X) \right\rangle \,\middle|\, x_1, x_2 \in X \wedge x_1 \leq x_2 \right\}$$

Pour chaque paire x_1, x_2 de points de X, on a donc un motif à deux points $\{x_1, x_2\}$ dont les occurrences sont données par érosion. L'algorithme consiste donc à :

1. Considérer tous les couples $(x_1, x_2) \in X^2$ avec $x_1 \leq x_2$;
2. Appliquer l'érosion $\varepsilon_{\{x_1,x_2\}}(X)$;
3. Enregistrer le couple $\left\langle x_2 - x_1,\ \varepsilon_{\{x_1,x_2\}}(X) \right\rangle$.

SIATEC La recherche de classes d'équivalence de translation (TEC) revient au calcul d'une érosion. Soient $x_1, x_2 \in X$ et $\{x_1, x_2\}$ l'élément structurant. On a :

$$TEC_X(\{x_1, x_2\}) = \varepsilon_{\{x_1,x_2\}}(X)$$

Dans ce cadre, la classe d'équivalence de translation associée au motif $\{x_1, x_2\}$ est obtenue par une première érosion de X. Pour chaque classe, on peut ensuite extraire les occurrences correspondantes dans X à l'aide d'une seconde érosion, ce qui conduit à la formulation suivante de SIATEC en termes morphologiques :

$$\text{SIATEC} : \left\{ \left\langle \varepsilon_{\{x_1,x_2\}}(X),\ \varepsilon_{\varepsilon_{\{x_1,x_2\}}(X)}(X) \right\rangle \,\middle|\, x_1, x_2 \in X \wedge x_1 \leq x_2 \right\}$$

Autrement dit, l'algorithme SIATEC s'interprète comme :

1. Calcul de la première érosion $\varepsilon_{\{x_1,x_2\}}(X)$, qui donne l'ensemble des positions de translation possibles du motif $\{x_1, x_2\}$;
2. Application d'une seconde érosion par cet ensemble pour obtenir les points de X effectivement couverts par les occurrences du motif : $\varepsilon_{\varepsilon_{\{x_1,x_2\}}(X)}(X)$;
3. Enregistrement du couple $\left\langle \varepsilon_{\{x_1,x_2\}}(X),\ \varepsilon_{\varepsilon_{\{x_1,x_2\}}(X)}(X) \right\rangle$.

On a encore une fois un lien très fort entre les outils de morphologie mathématique et les algorithmes existants de détection de motifs. Et ce, avec seulement des outils de base, ce qui en fait une piste particulièrement intéressante pour avoir une nouvelle interprétation des algorithmes déjà existants, mais surtout pour pouvoir créer de nouveaux algorithmes, basés spécifiquement sur ces outils.

2.3.3 Signature rythmique et polymétrie

La **signature rythmique** est un symbole de notation musicale qui définit la structure métrique d'une composition. Elle se présente sous la forme de deux chiffres placés l'un au-dessus de l'autre au début d'une partition, après la clé et l'armure (bien que la signature rythmique puisse être modifiée au cours de la pièce musicale). Le chiffre du haut indique le nombre de temps par mesure, tandis que celui du bas précise la valeur de note qui représente chaque temps.

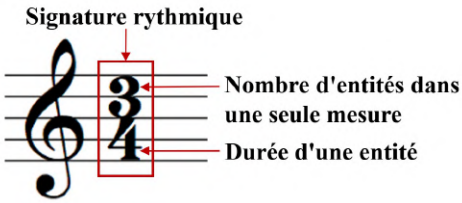


FIGURE 2.4 – Méthode de lecture d'une signature rythmique.

Par exemple, dans une signature 4/4, il y a quatre temps par mesure et chaque temps correspond à une noire. Une signature 3/4 (comme en figure 2.4) contient trois temps par mesure, toujours basés sur la noire. Cette notation permet d'organiser la musique en segments réguliers appelés mesures, ce qui facilite la lecture et l'interprétation.

Selon les recherches en neurosciences cognitives, la perception des signatures rythmiques implique des processus complexes. L'étude de Ross et al. (2022) sur la perception temporelle des rythmes musicaux montre que notre système sensori-moteur joue un rôle crucial dans la façon dont nous percevons et anticipons les structures métriques [12]. Cette capacité d'anticipation rythmique est fondamentale pour l'expérience musicale et permet aux auditeurs de "sentir" le rythme et de synchroniser leurs mouvements avec la musique. Bien qu'il semble y avoir une capacité naturelle à sentir la signature rythmique, il parait donc pertinent de chercher à détecter cette valeur de manière automatique.

La **polymétrie** consiste à utiliser simultanément plusieurs signatures rythmiques différentes dans une même pièce musicale. Contrairement à la polyrythmie qui superpose des rythmes différents dans la même mesure, la polymétrie fait coexister des mesures de longueurs différentes qui se décalent progressivement les unes par rapport aux autres. Un exemple classique serait une partie de piano jouée en 3/4 pendant qu'une partie de batterie reste en 4/4. Les deux parties commencent ensemble mais se décalent progressivement, créant un effet de "déphasage" avant de se retrouver synchronisées à nouveau après un certain nombre de mesures ($12 = 3 \times 4$). La figure 2.5 nous montre une polymétrie, et comment un rythme durant 2 temps et un rythme durant 3 temps peuvent se superposer.

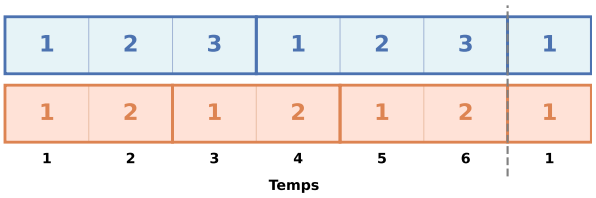


FIGURE 2.5 – Polymétrie entre 2 et 3 - qui se réaligne au bout de 6 temps.

La recherche de Møller et al. (2021) sur la perception des battements dans les polyrythmes démontre que notre système perceptuel traite le temps musical de manière hiérarchique et binaire [13]. L'étude révèle que lorsque plusieurs structures métriques entrent en compétition (comme dans la polymétrie), notre cerveau tend à privilégier l'une d'entre elles comme référence temporelle principale, ce qui explique pourquoi certaines combinaisons polymétriques sont plus faciles à percevoir que d'autres. Il est donc pertinent de chercher à extraire de manière automatique les différentes signatures rythmiques d'une même pièce, ce qui peut être difficile à appréhender avec des polymétries complexes, difficiles à percevoir facilement à l'oreille.

2.4 Objectifs du stage

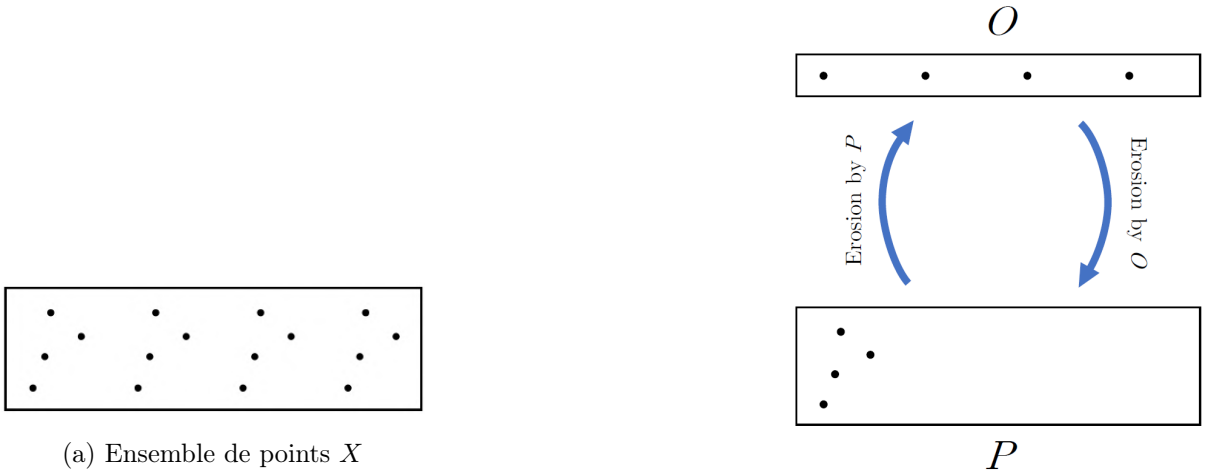
L'idée qui a fait émerger le sujet des travaux effectués lors de ce stage est donc d'extraire des informations musicales (ici la signature rythmique) à partir d'algorithmes de détection de motifs informés par la morphologie mathématique. La première partie consiste donc à développer un algorithme de détection de motifs informé par la morphologie mathématique, et la seconde partie consiste à en extraire des informations musicales, grâce aux propriétés des outils morphologiques.

Comment détecter un motif avec la morphologie mathématique ? Commençons par un exemple simple : un ensemble de points X constitué uniquement d'un même motif qui est répété 4 fois à intervalles de temps réguliers. En définissant P comme étant ce motif, si on calcule l'érosion de X par P, on obtient les activations temporelles O (ou *onset*) du motif, comme illustré à la figure 2.6.

(a) Ensemble X (b) Motif P (c) $O = \varepsilon_P(X)$

FIGURE 2.6 – Erosion d'un ensemble de points par un motif bien choisi (figure issue de [1]).

En choisissant correctement le motif, on peut donc trouver tous les moments où celui-ci apparait dans l'ensemble X de points, ou dans notre cas, dans la pièce musicale. Mais ce processus nécessite de connaître à l'avance le motif que l'on cherche à détecter, ce qui évite le problème que l'on cherche à résoudre qui est la détection de motifs musicaux. Cependant, les propriétés des outils morphologique permettent une solution (figure 2.7).



(a) Ensemble de points X

(b) Relation entre P et O

FIGURE 2.7 – Propriété de l'érosion (figure issue de [1]).

En effet, on a, pour (P, O) une paire MTEC (i.e. un sous ensemble de points tel qu'ajouter un point diminue son nombre d'occurrences) :

$$O = \varepsilon_P(X) \quad \text{mais aussi} \quad P = \varepsilon_O(X)$$

Ce qui veut dire que de manière analogue à la découverte des occurrences d'un motif, on peut découvrir le motif en connaissant là où il apparait. Ainsi, en choisissant de manière judicieuse l'élément structurant comme une occurrence classique de motifs, on peut potentiellement découvrir des motifs.

De plus, le choix des onsets en tant qu'élément doit être d'autant plus judicieux qu'on n'en extrait pas les mêmes informations selon lequel on choisit. En effet, comme on peut l'observer en figure 2.8, en fonction de la taille de l'élément structurant, on n'obtient pas forcément le même motif. Certains éléments structurants peuvent donc être plus judicieux à choisir en fonction de ce que l'on cherche. Ici l'élément structurant qui nous permet de trouver le nombre maximal de répétition est celui de taille 4, car notre motif se répète 4 fois dans X. On peut donc extraire une première information depuis l'élément structurant : le nombre d'occurences du motif. Le principal défi réside dans le fait d'avoir le bon élément structurant pour extraire les motifs qui nous intéressent.

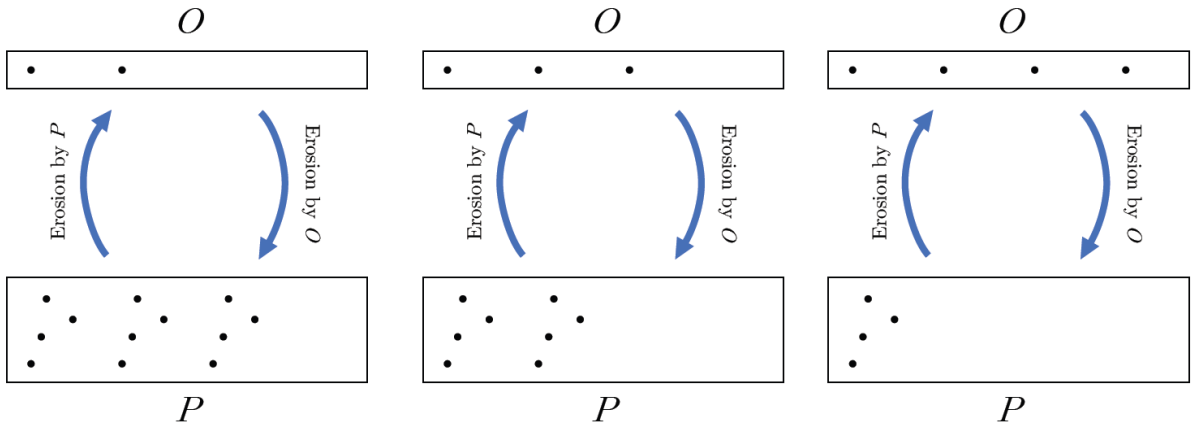


FIGURE 2.8 – Influence du choix de l'élément structurant (figure issue de [1])

Idée Choisissons un élément strucurant, notons O, défini comme suit :

$$O = \{(0,0),(L,0),(2L,0),...,(N \times L,0)\}$$

Il représente N points distants à intervalle de temps régulier L (figure 2.9).

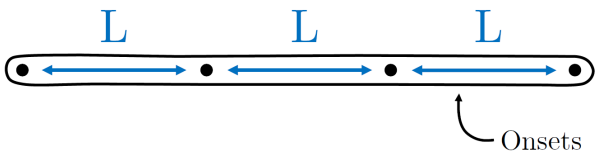


FIGURE 2.9 – Exemple d'onsets définissant l'élément structurant (ici avec $N = 4$) (figure issue de [1])

Ainsi, on sait pour tout P définit comme $P = \varepsilon_O(X)$ que c'est **un motif qui se répète exactement N fois à intervalle de temps régulier L** [11].

De plus, Lascabettes émet l'idée que cette distance L peut elle aussi avoir une interprétation musicale : la signature rythmique. Ou plus précisément, la valeur du haut de la signature rythmique (figure 2.10).

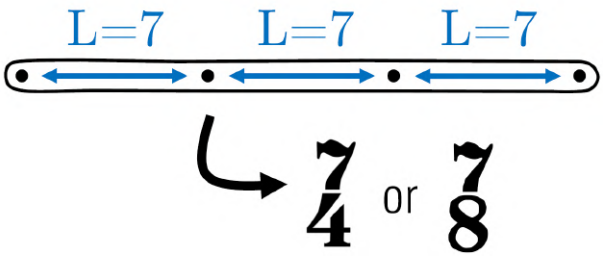


FIGURE 2.10 – Interprétation musicale de L (figure issue de [1]).

Remarque : Il est bien important de voir que l'on peut déduire la signature rythmique, mais que ce n'est pas trivial. Une valeur de 7 peut en effet désigner une signature rythmique de 7/4 ou de 7/8. Ici nous nous concentrerons uniquement sur la veleur du haut de la signature rythmique, que nous appellerons tout de même signature rythmique tout au long de ce rapport pour des questions de lisibilité.

L'objectif de ce stage est donc de vérifier cette idée sur un corpus de pièces musicales, le tout en développant des algorithmes de détection de motifs musicaux innovants, basés sur la morphologie mathématique, et en extrayant des informations musicales (la signature rythmique et le nombre de répétitions). Ces travaux sont d'autant plus innovants que les algorithmes actuels n'ont pas d'interprétabilité musicale plus forte que la simple répétition du motif trouvé. Ici on a pour but de détecter les motifs, mais surtout d'en sortir des informations directement interprétables et utiles à des fins d'analyses musicologiques.

Chapitre 3

Prétraitement des données musicales

3.1 De la musique à un nuage de points 2D

Comme décrit dans la section 2.1.1, la musique peut être représentée comme un ensemble finis de points dans $\mathbb{R}^n$. Dans notre cas, chaque note est encodée comme un point dans l'espace $\mathbb{R}^2$, selon ses deux caractéristiques principales :

— **Onset** : temps d'apparition de la note (en battements),
— **Pitch** : hauteur de la note (standard MIDI).

On construit ainsi un *nuage de points* $X \subset \mathbb{R}^2$, où chaque élément représente une note de la pièce musicale.

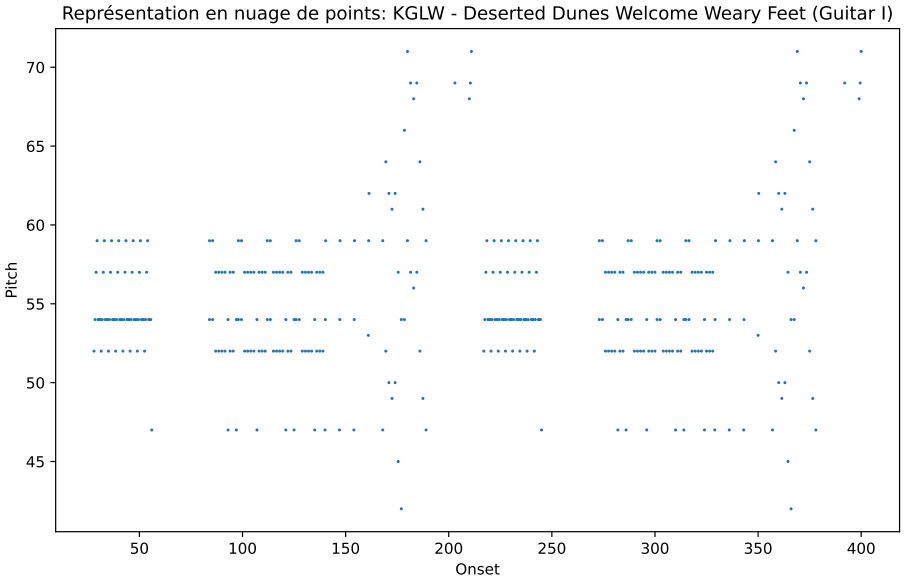


FIGURE 3.1 – Représentation d'une pièce musicale en nuage de points 2D.

Cette modélisation permet de transposer des outils issus de la **morphologie mathématique** – initialement développés pour le traitement d'images – dans un cadre musical : l'idée est de traiter l'ensemble X comme une image binaire pour en extraire des structures répétitives et significatives.

Remarque : Bien que nous ayons fait le choix de travailler en dimension $n = 2$, les outils et algorithmes qui ont été développés sont tout autant valides pour tout n dans $\mathbb{N}$.

3.2 Choix de la base de données

L'objectif de ce travail est de détecter des motifs rythmiques répétitifs afin d'en extraire la signature métrique. Idéalement, il serait souhaitable de disposer d'une base de données comportant une grande variété de signatures rythmiques. Cependant, la majorité des œuvres musicales enregistrées repose sur un nombre restreint de signatures métriques, en particulier les mesures binaires ou ternaires simples [14].

Un groupe australien, *King Gizzard & the Lizard Wizard*, constitue toutefois une exception notable. Formé en 2010, il est reconnu pour sa créativité et son éclectisme, explorant des styles allant du rock psychédélique au microtonal en passant par le thrash metal et le jazz fusion. Leur production prolifique (vingt-sept albums en quinze ans) s'accompagne d'une volonté constante d'expérimentation formelle, notamment dans l'usage de signatures rythmiques atypiques et de structures polymétriques complexes. Parmi leurs nombreuses réalisations, l'album *Polygondwanaland* [15], sorti en 2017, occupe une place singulière. Offert librement au public sous licence Creative Commons BY-ND, il a été mis à disposition avec ses *masters*, autorisant ainsi sa redistribution et son exploitation, y compris à des fins commerciales. Cette initiative a suscité un engouement considérable : un an après sa sortie, on recensait plus de 215 éditions différentes, publiées par plus d'une centaine de labels à travers le monde. Cet album a par ailleurs bénéficié d'un accueil critique très favorable, la presse spécialisée saluant à la fois sa richesse compositionnelle et sa capacité à rendre accessibles des formes rythmiques complexes à un large public. Ce succès commercial et artistique atteste de son statut d'œuvre de musique populaire contemporaine, tout en intégrant des procédés rythmiques relevant d'une recherche musicale avancée.

Parmi les nombreuses adaptations issues de cette mise à disposition, on trouve notamment une version « 8-bit » produite par *8-bit Escapades* [16], accompagnée de ses fichiers MIDI librement téléchargeables. Cette version constitue un matériau de choix pour nos expérimentations, car elle combine la complexité polymétrique caractéristique de l'album original avec un format MIDI permettant un traitement algorithmique direct. Ainsi, *Polygondwanaland* représente une base de données idéale pour évaluer la performance de nos méthodes : il s'agit d'une œuvre actuelle, populaire, et riche en signatures rythmiques peu communes, offrant un terrain d'expérimentation pertinent pour la détection automatique de motifs.

3.3 Traitements des données

Dans ce contexte, la préparation du dataset est une étape cruciale, afin de garantir que la représentation temporelle et la précision des données soient compatibles avec les outils morphologiques utilisés par la suite. La section suivante détaille les étapes de séparation, nettoyage et quantification appliquées aux fichiers MIDI du corpus.

3.3.1 Séparation et nettoyage

Chaque fichier MIDI contient différentes pistes, qui représentent chacune un instrument de la pièce musicale. Il est donc naturel de vouloir traiter indépendamment ces pistes lorsque l'on veut faire un travail analytique sur de tels fichiers. La première étape consiste donc à séparer les fichiers MIDI piste par piste, afin d'étudier les ensembles de points de chaque instrument de manière indépendante. Parmi ces pistes, certaines ne sont pas pertinentes pour notre étude. En effet, notre objectif étant de détecter des motifs dans un contexte musical, il faut qu'il y ait la possibilité d'avoir des motifs dans notre ensemble de points. Ainsi, il parait peu pertinent d'étudier une piste qui a seulement quelques notes isolées. Toutes les pistes «non pertinentes» ont donc été supprimées de nos fichiers (soit, 8 pistes supprimées et 89 pistes conservées). En annexe A, la liste de toutes les pistes concernées est détaillée. Le tableau 3.1 montre toutes les pistes que nous avons gardées pour notre base de données.

Instruments	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**
Guitar I	X	X	X	X	X	X	X	X	X	X
Guitar II	X	X	X	X	X	X	X	X		
Guitar III	X	X	X	X	X	X	X			
Bass	X	X	X	X	X	X	X	X	X	X
Keys I	X	X	X	X	X	X	X			
Drums Right	X	X	X	X	X	X	X	X	X	
Drums Left	X	X	X	X	X	X	X	X	X	X
Harmonica	X	X	X	X						
Flute	X	X	X	X	X	X	X			
Keys II	X									
Flute/Keys II	X									
Intro Keys	X									
Flute III	X	X	X	X	X					
Keys III	X									
Keys IV										
Low bass Key										
Flute II	X	X	X							
Percussion 1	X									
Percussion 2	X									
Percussion 3	X									
Percussion 4	X									
Percussion 5	X									
Nombre de pistes	12	7	9	9	7	9	9	11	9	7

TABLE 3.1 – Bilan de toutes les pistes MIDI utilisables par pièce musicale dans l'album *Polygondwanaland*. Les numéros (1) à (10) correspondent aux dix pièces de l'album, dans l'ordre de la tracklist.

3.3.2 Conversion temporelle et quantification

Conversion temporelle Dans un fichier MIDI, chaque note est encodée avec une valeur d'activation temporelle (ou *onset*). Cette dernière est donnée en secondes. Cette unité est peu informative dans un contexte musical. Une idée est donc de représenter l'axe temporel en prenant en compte le contexte musical de la pièce en question, c'est-à-dire d'avoir des valeurs d'activations temporelles en battements (ou *beats*). Ainsi, on a une vraie information facilement interprétable pour une pièce musicale : 1 battement = 1 noire, 2 battements = 1 blanche et 4 battements = 1 ronde, 1/2 battement = 1 croche, etc ...

Les activations temporelles en secondes n'ont pas cette interprétabilité. Nous les avons convertis en battements selon la formule suivante :

$$\text{onset}_{\text{beats}} = \text{onset}_{\text{seconds}} \cdot \frac{\text{bpm}}{60}$$

Quantification Nous travaillons avec des outils de morphologie mathématique, sensibles aux erreurs. Ainsi, nous avons nécessairement besoin de valeurs **exactes**. Autrement dit, si nous cherchons deux motifs identiques séparés d'une distance L, il n'existe aucun ε non nul tel que deux motifs séparés de $L + \varepsilon$ soient détectés. Or dans un contexte informatique, il est assez courant d'avoir des approximations, pour des raisons d'encodage et de mémoire. Par exemple, si l'on veut représenter 1/3 en valeur flottante (**float**), on aura forcément une approximation (ex : $0,33334$ ou $0,33330$). Or, cette approximation, aussi fine soit-elle, met en péril l'approche morphologique. Il nous faut donc un encodage des valeurs nous permettant d'avoir toujours des valeurs exactes. Pour cela, il est nécessaire de les **quantifier**, c'est à dire, de ne plus permettre n'importe quelle valeur mais d'imposer que toutes les valeurs soient un multiple d'un pas de quantification.

Dans un contexte musical, le pas de quantification peut être vu de manière analogue au **tatum**, qui est défini par Bilmes comme « le plus petit intervalle de temps entre deux notes successives dans une phrase rythmique » [17]. Pour des raisons d'uniformité au sein de la base de données, nous avons retenu un tatum fixe pour l'ensemble de la quantification. La valeur choisie est $\frac{1}{8\times3}$. Ce choix permet de prendre en compte les rythmes ternaires (facteur $\frac{1}{3}$) tout en conservant une résolution temporelle fine à la triple croche (facteur $\frac{1}{8}$). Bien que cette valeur ait été fixée de manière arbitraire, elle demeure cohérente avec les observations d'Iyer et al., qui indiquent que le tatum en musique occidentale se situe généralement autour de $\frac{1}{16}$ ou de $\frac{1}{24}$ [18].

Il faut donc ensuite quantifier les valeurs d'activations temporelles afin qu'elles soient toutes des multiples du tatum. On notera évidemment que pour des raisons de précision, nous avons laissé toutes les valeurs en fraction et non en valeur flottante. Ainsi, aucune approximation n'est faite par l'ordinateur lors des différents calcul, ce qui permet d'éviter que l'érosion ne reconnaisse pas des motifs à cause d'une approximation interne lors de la gestion des flottants par l'ordinateur.

Format de sortie Une fois la base de données traitée, elle a été enregistrée à des fins de réutilisabilité. La base de données finale est constituée de fichiers `.csv` pour chaque piste, contenant pour chaque note les informations suivantes :

- **Onset** en battements,
- **Pitch** (hauteur de la note, standard MIDI),
- **Durée** en battements.

Remarque : Il est important de noter que la valeur de durée n'est pas utilisée dans le cadre de notre étude, mais il a été jugé pertinent de la laisser dans la base de données pour d'autres études.

L'ensemble des pistes du corpus a ainsi été restructuré de manière homogène, fournissant une base solide pour nos traitements ultérieurs de détection de motifs rythmiques répétitifs. Ce jeu de données constitue une ressource précieuse en soi pour l'étude de la musique polyphonique sous forme symbolique, et peut potentiellement servir à d'autres projets en analyse musicale computationnelle.

Chapitre 4

Détection de motifs musicaux via la morphologie mathématique

4.1 Premier algorithme : répétitions avec nombre d'occurrences fixe

4.1.1 Étapes de l'algorithme

Comme décrit dans la section 2.4, on peut extraire des informations musicales sur des motifs se répétant de manière exacte à intervalles de temps réguliers en appliquant une érosion à l'ensemble X de points par l'élément structurant suivant : $O = \{(0,0),(L,0),...,(N \times L,0)\}$ [11].

Une première idée serait alors d'appliquer des érosions successives pour tout un ensemble de valeurs de L et de N, ce qui nous permettrait de trouver tous les motifs qui se répètent exactement N fois à intervalle de temps régulier L. C'est un résultat qui serait intéressant, mais qui peut vite être coûteux en calcul si l'on teste un grand nombre de valeurs de N et de L. Nous avons alors choisi de fixer la valeur de N à 4 pour ne garder que les motifs qui se répètent exactement 4 fois, et de tester tout un ensemble de valeurs de L différentes. Il est bon de préciser que le choix de $N = 4$ est arbitraire, mais justifié d'un point de vue musical. En effet, si l'on prend tous les «motifs» ayant seulement deux occurrences, beaucoup d'entre eux risquent d'être non pertinents. Et choisir les motifs qui se répètent 4 fois a du sens car le nombre de répétitions semble être suffisant pour avoir un motif qui a un intérêt à étudier. D'autant plus que 4 est une valeure courante dans le contexte de l'analyse musicale.

Principe de fonctionnement :

— Pour chaque longueur L dans une plage prédéfinie (par exemple $L \in \{2,\ 2{,}25,\ \ldots,\ 20\}$, un élément structurant régulier est défini :

$$O_L = \{(0,0),(L,0),(2L,0),(3L,0)\}$$

— L'ensemble X des points est ensuite soumis à une érosion morphologique par O_L, ce qui donne :

$$P = \varepsilon_{O_L}(X)$$

— Chaque motif P détecté est enregistré avec la longueur L correspondante.

Chaque paire (P, L) représente ainsi un motif rythmique extrait et la signature rythmique associée.

Algorithme 1 Détection par érosion simple

1: **for** $L \in [2, 2.25, \ldots, L_{max}]$ **do**
2: $\quad O_L \leftarrow \{(0,0), (L,0), (2L,0), (3L,0)\}$
3: $\quad P \leftarrow \varepsilon_{O_L}(X)$
4: $\quad$ Enregistrer (P, L)
5: **end for**
6: **return** Ensemble des couples (P, L) pour chaque longueur L

Définition 4.1 : Algorithme 1

L'algorithme extrait tous les sous-ensembles **MTEC** de X qui se répètent **exactement 4 fois** à intervalles de temps réguliers.

$$\left\{ \langle P, L \rangle \,\middle|\, P = \varepsilon_{O_L}(X),\ O_L = \{(n \cdot L, 0)\}_{n=0}^{3},\ L \in \{2,\ 2{,}25,\ \ldots,\ L_{max}\} \right\}$$

Rappel : Un motif MTEC est un sous ensemble de points tel qu'ajouter un point diminue son nombre d'occurence.

Cet algorithme constitue une base simple pour la détection de motifs élémentaires (dans le sens où ils se répètent de manière exacte et régulière).

4.1.2 Résultats

Lors de l'application de ce premier algorithme, nous prenons comme ensemble de points X une piste (un instrument) d'une pièce musicale donnée. En effet, il semble pertinent de considérer les notes d'un instrument comme indépendantes des autres dans notre cas. Autrement dit, on va chercher les motifs répétitifs d'un instrument donné, et non pas des motifs qui comprennent plusieurs instruments en même temps. Ainsi, notre algorithme nous renvoie pour différents L, un motif détecté. Le nombre de motifs trouvés est inférieur ou égal au nombre de valeurs de L testées. En effet, on va trouver au maximum un motif par L, et il se peut (très probablement) que pour un L donné, on ne trouve pas de motif. Cette observation constitue un élément rassurant quant à la validité de notre approche : dans la mesure où la valeur de L possède une interprétation musicale directe, il apparaît cohérent que, pour une œuvre donnée, seules certaines valeurs soient effectivement porteuses de motifs répétitifs. La figure 4.1 nous montre un exemple de motif trouvé pour un L donné dans une pièce musicale.

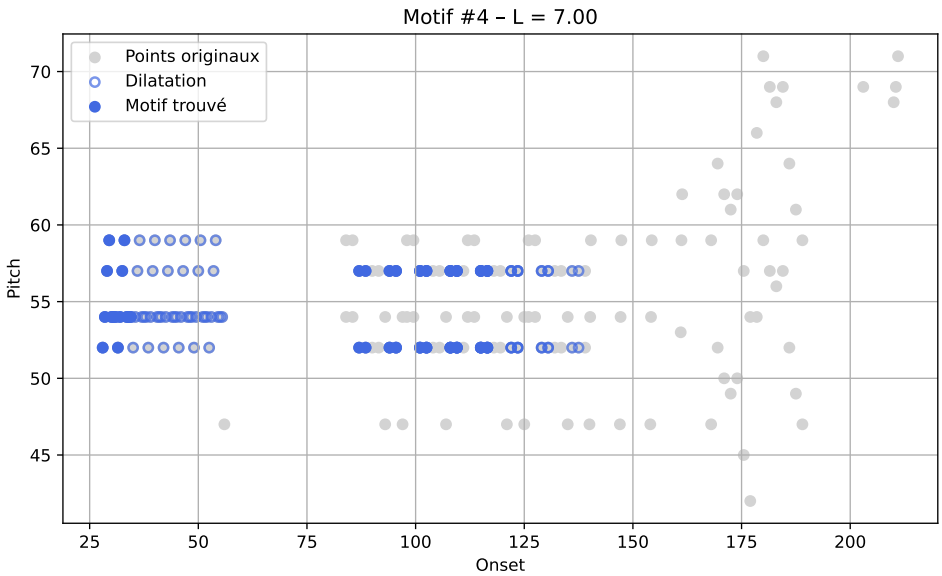


FIGURE 4.1 – Motif détecté pour $L = 7$ dans la Guitar I de *Deserted Dunes Welcome Weary Feet*.

On voit les points du motif détecté, ainsi que tous les points appartenant à une des quatre occurences de ce même motif. Il est important de rappeler que ce n'est pas le seul motif détecté par l'algorithme. Ici pour près de 75 valeurs de L testées, on obtient un motif pour 7 valeurs différentes de L.

4.1.3 Limites

Cet algorithme montre des limitations claires, qui sont clairement observables sur la figure 4.1. Tout d'abord, on voit qu'un motif détecté pour un L donné n'est pas forcément un seul motif au sens que l'on a défini depuis le début. En effet, si l'on reprend la figure 4.1, on peut voir que les points détectés à gauche et les points détectés à droite font partie de deux motifs différents d'un point de vue musical. Outre le fait que tous les deux se répètent à un même intervalle de temps régulier L, les deux ensembles de points ne semblent pas être liés. On aimerait donc permettre de **séparer ces deux ensembles de points** et avoir un algorithme qui trouve automatiquement **les plus pertinents** d'un point de vue musical. Ensuite, une autre limite est que le nombre de répétitions est fixe ($N = 4$). Or, rien ne nous garantit qu'un motif se répète seulement 4 fois. Il est tout à fait envisageable qu'un motif ait un nombre de répétitions supérieur à 4. Ainsi, il faudrait **trouver le nombre de répétitions maximal** des motifs.

4.2 Second algorithme : extraction des sous-motifs pertinents

Afin de dépasser les limitations de ce premier algorithme, des critères ont été ajoutés en développant un nouvel algorithme permettant d'extraire des motifs de manière plus pertinente et de manière plus informative d'un point de vue musical.

Séparation des motifs après détection pour un L donné Comme indiqué précédemment, pour une valeur donnée de L, le premier algorithme extrait l'ensemble des occurrences répétées exactement quatre fois à intervalles réguliers de durée L. Toutefois, rien ne garantit que cet ensemble corresponde à un unique motif du point de vue musical : il peut en réalité s'agir de la réunion de plusieurs motifs distincts. L'objectif est donc de mettre en place un procédé permettant de les séparer automatiquement. Pour ce faire, nous considérons que deux motifs sont distincts lorsque la distance temporelle qui les sépare dépasse un certain seuil, traduisant l'existence d'une « discontinuité » entre les points. Bien que ce terme ne soit pas strictement approprié dans le cadre d'un ensemble discret, il reflète l'idée d'une rupture significative dans la distribution des points. Ce principe a conduit à la mise en place d'un processus de fenêtrage. Plus précisément, les points P issus de l'érosion sont divisés en sous-ensembles de durée temporelle exacte L au moyen d'un glissement de fenêtre. Chaque sous-ensemble obtenu est considéré comme un motif candidat, formant ainsi la liste P_{list}. Le fenêtrage est initialisé au premier point des motifs candidats, puis des fenêtres de taille L sont générées successivement. Dès qu'aucun point n'est contenu dans une fenêtre (phénomène assimilé à une « discontinuité »), une nouvelle fenêtre est ouverte en commençant au point suivant.

Algorithme 2 Fenêtrage des points (`windowing`)

Require: Ensemble de points P, longueur L
1: Trier P selon l'onset
2: Initialiser windows $\leftarrow \emptyset$
3: **while** il reste des points dans P **do**
4: Prendre le plus petit onset t
5: Extraire la fenêtre $W \leftarrow \{p \in P \mid t \leq p_0 < t + L\}$
6: **if** $W \neq \emptyset$ **then**
7: Ajouter W à windows
8: **end if**
9: Supprimer les points p de P tels que $p_0 < t + L$
10: **end while**
11: **return** windows

Nombre maximal de répétitions L'algorithme précédemment décrit permettait de détecter les motifs se répétant **exactement** quatre fois. Toutefois, rien ne garantit qu'un motif ne puisse pas se répéter plus de quatre fois. Une approche envisagée consistait à faire varier la valeur de N dans la définition

de l'élément structurant, de manière à tester plusieurs nombres de répétitions successifs, jusqu'à identifier la valeur maximale. Cependant, cette méthode s'avère particulièrement coûteuse en temps de calcul. Nous avons néanmoins identifié une méthode plus efficace, permettant de contourner cette limite tout en réduisant la complexité computationnelle.

Résultat 4.1 : Propriété de hiérarchie des motifs répétés

Soit P un motif se répétant N fois dans un ensemble X. Soit P' un motif tel que $P' \subset P$ et P' se répète N' fois dans X avec $N' \geq N$.
Alors le motif P' se répète **périodiquement** à l'intérieur de P.

Conséquence : Une fois qu'une érosion a été effectuée pour une valeur donnée N, l'ensemble de points extraits par cette érosion permet de retrouver tous les motifs ayant un nombre maximal de répétitions supérieur ou égal à N simplement en vérifiant s'il y a une périodicité interne dans l'ensemble de points érodé.

Explication du processus de filtrage Les motifs candidats après fenêtrage sont utilisés comme une liste de motifs à tester. Ainsi, il suffit de voir si des sous-ensembles de points fenêtrés sont égaux à un sous-ensemble de points d'une autre fenêtre. En morphologie mathématique, cela revient à voir si un motif candidat est inclus dans **l'ouverture** d'un autre motif. Nous pouvons ainsi savoir quel est le motif MTEC qui se répète un nombre maximal de fois.

Algorithme 3 Filtrage morphologique (`filtering`)

Require: P_{list}, ensemble initial X, longueur L
1: $P_{\text{to_test}} \leftarrow P_{\text{list}}$
2: $P_{\text{output}} \leftarrow \emptyset$
3: **while** $P_{\text{to_test}} \neq \emptyset$ **do**
4: Choisir $P_j \in P_{\text{to_test}}$
5: $\mathcal{O}_j \leftarrow \delta_{P_j}(\varepsilon_X(P_j))$
6: $N_j \leftarrow 4$
7: **for all** $P_k \in P_{\text{list}} \setminus \{P_j\}$ **do**
8: **if** $P_k \cap \mathcal{O}_j \neq \emptyset$ **then**
9: $N_j \leftarrow N_j + 1$
10: **if** $P_k \subseteq \mathcal{O}_j$ **then**
11: Supprimer P_k de $P_{\text{to_test}}$
12: **end if**
13: **end if**
14: **end for**
15: Ajouter (P_j, N_j, L) à P_{output}
16: Supprimer P_j de $P_{\text{to_test}}$
17: **end while**
18: **return** P_{output}

Autrement dit, le filtrage se fait de la manière suivante :

1. Tant que $P_{\text{to_test}}$ n'est pas vide, on sélectionne le motif P_j le plus long encore à tester.
2. On calcule l'ouverture $\mathcal{O}(P_{\text{to_test}}, P_j)$, définie comme la dilation de l'érosion de $P_{\text{to_test}}$ par P_j.
3. Pour chaque autre motif P_k (distinct de P_j) dans P_{list} :
   — Si au moins un point de P_k est inclus dans l'ouverture de P_j, on incrémente le compteur N_j.
   — Si **tous** les points de P_k sont inclus dans cette ouverture, alors P_k est considéré comme redondant et est supprimé de $P_{\text{to_test}}$.
4. On ajoute alors (P_j, N_j) à la liste finale P_{output}.
5. On retire P_j de la liste $P_{\text{to_test}}$.

4.2.1 Étapes de l'algorithme

Maintenant que tous les outils utilisés ainsi que le principe de l'algorithme ont été introduits, il est possible de s'intéresser en détail à l'architecture de notre algorithme. Soit X un ensemble de points représentant les événements musicaux dans un espace de type $(\text{onset}, \text{pitch})$.

1. **Érosion structurée.** Pour chaque valeur de longueur L dans une plage prédéfinie ($L \in \{2,\ 2{,}25,\ \ldots,\ 20\}$), un élément structurant $O_L = \{(0,0),(L,0),(2L,0),(3L,0)\}$ est défini. L'érosion de X par O_L est appliquée pour extraire les points P correspondant à des occurrences initiales de motifs traduits (exactement comme pour le premier algorithme).
2. **Fenêtrage.** Les points P issus de l'érosion sont divisés en sous-ensembles de longueur temporelle exacte L à l'aide d'un glissement de fenêtre. Chaque sous-ensemble obtenu constitue un candidat motif, formant ainsi la liste P_{list}.
3. **Filtrage morphologique.** Un processus de sélection basé sur une opération d'ouverture est appliqué. L'idée est de :
   — Maintenir une liste $P_{\text{to_test}}$ initialisée avec les éléments de P_{list}.
   — Garder une liste P_{output} vide au départ.

Algorithme 4 Détection par ouverture morphologique

```
for L ∈ [2, 2.25, ..., L_max] do
    O_L ← {(0,0), (L,0), (2L,0), (3L,0)}
    P ← ε_{O_L}(X)
    P_windowed ← windowing(P, L)
    (P_j, N_j, L) ← filtering(P_windowed, P, L)
end for
return Ensemble des triplets (P_j, N_j, L)
```

Sortie L'algorithme retourne, pour chaque valeur de L, une collection de motifs sous la forme (P'_j, N_j, L), où P'_j est un motif filtré, N_j le nombre d'occurrences détectées, et L la longueur temporelle considérée. Cette méthode garantit un encodage compact en représentant les données par un nombre optimal de motifs et leur nombre d'occurences.

4.2.2 Définitions et propriétés

Érosion et onsets Toutes ces idées de fenêtrage et de filtrage, bien que facilement explicables d'un point de vue informatique, sont difficiles à formaliser en mathématiques de manière rigoureuse. Ainsi, afin de pouvoir donner une définition formelle à notre algorithme, il est nécessaire d'introduire de nouvelles notations.

Définition 4.2 : Érosion translatée d'un ensemble

Soit X un ensemble fini de points - par exemple, des paires $(\text{onset}, \text{pitch})$ - soit $P = \{p_0, p_1, \ldots, p_k\}$ un motif, et v un vecteur de ce même ensemble. On définit l'érosion de X par P translatée par v, notée $\varepsilon_P(X)_v$, comme :

$$\varepsilon_P(X)_v = \{t + v \mid P_t \subset X\} = \varepsilon_P(X) + v$$

Avec une telle érosion, n'importe quelle vecteur n'est pas judicieux à choisir. En effet, pour qu'un vecteur ait un intérêt en analyse de données, il faudrait qu'il extraie des points qui sont inclus dans l'ensemble X. On appellera un tel objet un vecteur **valide**.

Résultat 4.2 : Validité d'un vecteur

Soient $X \subset \mathbb{R}^n$ un ensemble de points, $P = \{p_0, \ldots, p_k\} \subset \mathbb{R}^n$ un motif, et v un vecteur de l'ensemble de points X. Alors :

$$\varepsilon_P(X)_v \subseteq X \quad \Leftrightarrow \quad v \in P$$

Autrement dit, une occurrence de P dans X via l'érosion est uniquement possible si le vecteur de translation correspond aux coordonnées d'un point du motif P. En particulier, nous pouvons alors nous intéresser au **premier point** de ce motif.

Définition 4.3 : Onsets d'un motif

Les *onsets* d'un motif $P = \{p_0, \ldots, p_k\}$ dans X sont définis comme les positions d'apparition de P dans X, via l'érosion avec le vecteur $p_0 \in P$ (le premier point du motif). On note cet ensemble :

$$\Omega_P(X) = \varepsilon_P(X)_{p_0}$$

Ce sont les points de X à partir desquels P est détecté par érosion.

Depuis le début de cette étude, nous considérons des « motifs se répétant à intervalle de temps régulier L », ce qui a permis la mise en place d'un fenêtrage de taille L. Nous proposons à présent une définition formelle de cette notion, en utilisant les onsets du motif.

Résultat 4.3 : Occurrences régulières

Soit $P \subset \mathbb{R}^2$ un motif tel que

$$\Omega_P(X) = \{(t_0, p_0), (t_1, p_1), \ldots, (t_N, p_N)\}.$$

Si les onsets sont équidistants, c'est-à-dire $\exists L > 0$ tel que :

$$t_{i+1} - t_i = L \quad \forall i \in [\![0, N]\!],$$

alors le motif P se répète à intervalles réguliers dans X, et chaque occurrence est contenue dans une fenêtre de longueur L, débutant à un onset de $\Omega_P(X)$.

Avec ces nouvelles notations, il est maintenant possible de définir formellement le second algorithme de la manière suivante :

Définition 4.4 : Second algorithme

L'algorithme extrait tous les sous-ensembles **MTEC** $P \subset X$, répétés un nombre **maximal** de fois $N \geq 4$ à intervalles de temps réguliers dans des fenêtres successives de même longueur L, chacune débutant à un onset (i.e. un point de $\Omega_P(X)$).

Rappel : Un motif MTEC est un sous-ensemble de points tel qu'ajouter un point diminue son nombre d'occurrences.

L'algorithme reconstruit un ensemble $X' \subseteq X$ tel que :

$$X' = \bigcup_{i=1}^{m} \gamma_{P_i}(X)$$

où chaque P_i est un motif détecté et sélectionné, $\gamma_{P_i}(X)$ son ouverture et m le nombre de motifs trouvés.

Résultat 4.4 : Différence entre les nombres de motifs détectés

La différence entre le nombre de motifs détectés par le second algorithme et le nombre de motifs détectés par le premier algorithme est exactement égale au nombre de *trous* [a] dans les motifs extraits par le premier algorithme.

a. Un **trou** est défini comme une *discontinuité* dans la suite des fenêtres successives de longueur L contenant des occurrences d'un motif P.

Justification du résultat

- Un **trou** est un ensemble maximal de fenêtres vides qui sépare deux occurrences consécutives de P.
- Le premier algorithme regroupe toutes les occurrences d'un motif P répété à intervalle régulier L, même si certaines fenêtres intermédiaires ne contiennent pas le motif (présence de trous).
- En revanche, le second algorithme impose que les occurrences apparaissent dans des fenêtres strictement *consécutives*, sans interruption. Ainsi, chaque discontinuité dans la suite des occurrences provoque une séparation du motif initial en plusieurs motifs distincts.
- Par conséquent, chaque **trou** dans la suite des occurrences de P détectées par le premier algorithme entraîne une *séparation* en deux motifs dans le second algorithme, ce qui explique que la différence de cardinalité entre les ensembles de motifs détectés par le second algorithme et le premier algorithme est égale au nombre total de trous.

Bilan des motifs détectés La figure 4.2 permet de visualiser les différences entre les motifs détectés par les deux algorithmes.

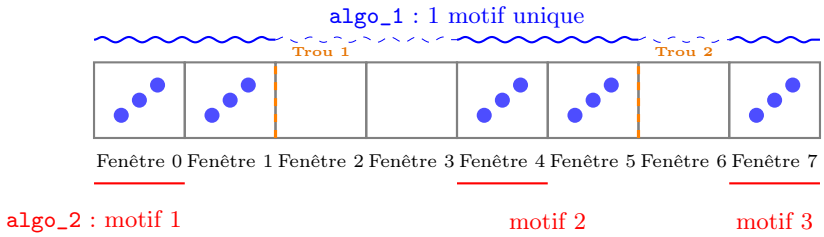


FIGURE 4.2 – Différence entre les motifs détéctés par les deux algorithmes et le nombre de trous.

4.2.3 Résultats

La figure 4.3 nous montre un exemple de motif trouvé pour un L donné dans une pièce musicale. De nouvelles informations apparaissent clairement par rapport au résultat précédent. En effet, il y a bien **deux motifs distincts**, ici en rouge et bleu, qui ont leurs propriétés propres et un contexte musical différent. Il semblait donc pertinent de les extraire distinctement. De plus, on obtient une information supplémentaire sur le motif rouge qui est son **nombre exact de répétitions** (huit fois), et non seulement quatre, ce qui permet d'avoir une représentation plus compacte des motifs, et aussi d'avoir une information musicale intéressante. Il est toujours bon de rappeler que cette figure est pour une valeur de L seulement, mais des motifs sont trouvés pour différentes valeurs de L (le même nombre que dans l'algorithme précédent puisque la même érosion est appliquée et que les étapes suivantes sont appliquées seulement sur le sous-ensemble érodé). Les figures générées pour chaque pièce musicale de l'album (pour L_{max}) sont en annexe C.

4.3 Comparaison avec les algorithmes existants

L'objectif de ces travaux étant de trouver des nouvelles méthodes de détection de motifs musicaux en utilisant des outils de morphologie mathématique, il est naturel de chercher à comparer les résultats obtenus par ces nouvelles méthodes avec les algorithmes de référence. Pour ce faire, il y a trois axes de comparaison sur lesquels nous allons nous pencher : le nombre de motifs détectés, la complexité computationnelle et l'interprétabilité musicale. Pour ce faire, nous allons comparer nos résultats avec ceux de SIA et SIATEC [3].

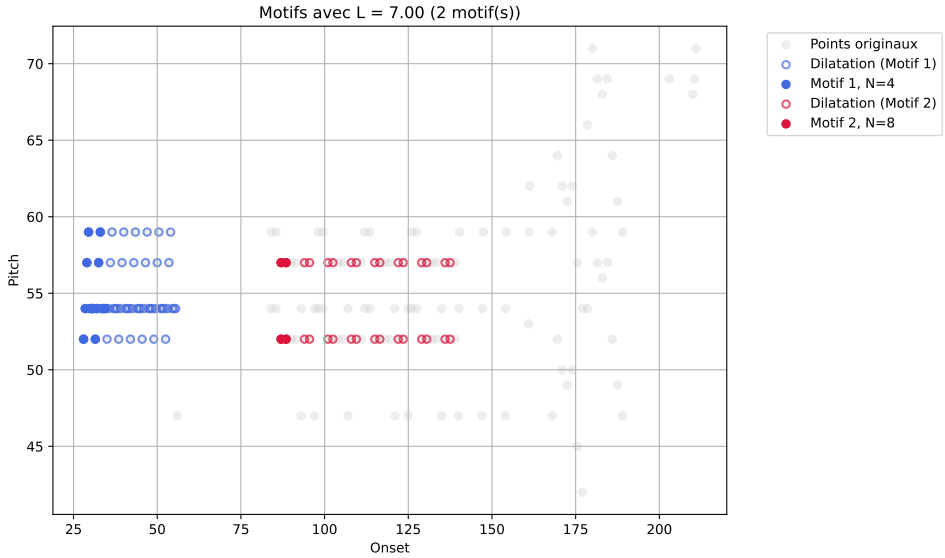


FIGURE 4.3 – Motif détecté pour un L donné dans la Guitar I de *Deserted Dunes Welcome Weary Feet*

Nombre de motifs détectés Un des principaux axes d'amélioration des algorithmes de découverte de motifs est le nombre de motifs détectés. En effet, ces algorithmes cherchent l'exhaustivité des motifs répétitifs, ce qui fait que de nombreux, voire une majorité, de ce qui est extrait n'est pas pertinent d'un point de vue musical. Cela est illustré au tableau 4.1

Track	**Nombre de motifs**	**Algo 1**	**Algo 2**	**SIA**	**SIATEC**
Tetrachromacy (Bass)		23	88	11133	8951
Tetrachromacy (Drums Left)		47	618	57065	33896
Tetrachromacy (Drums Right)		47	510	53218	32219
Tetrachromacy (Flute)		4	7	109	82
Tetrachromacy (Flute II)		1	1	65	51
Tetrachromacy (Flute III)		1	1	81	51
Tetrachromacy (Guitar I)		29	133	12760	10763
Tetrachromacy (Guitar II)		23	50	7463	4035
Tetrachromacy (Guitar III)		27	162	17455	14800
Tetrachromacy (Harmonica)		0	0	38	26
Tetrachromacy (Keys I)		0	0	7	6

TABLE 4.1 – Comparaison du nombre de motifs détectés par piste et par algorithme.

Nous pouvons voir qu'avec SIA et SIATEC, plusieurs dizaines de milliers de motifs peuvent être détectés pour une seule piste alors qu'avec nos algorithmes quelques centaines tout au plus sont identifiés. Nous pouvons expliquer ce résultat simplement car, contrairement aux méthodes proposées par Meredith, nous ne cherchons pas l'exhaustivité des motifs mais seulement les répétitions exactes à intervalles de temps réguliers. Cela naturellement, explique que nous ne détectons pas un nombre très conséquent de motifs potentiels. Mais ce choix est justifié car nous cherchons seulement les motifs les plus pertinents, et il nous permet d'enlever énormément de motifs peu informatifs musicalement. De plus, nous avons choisi de n'accepter qu'un nombre de répétitions minimal de quatre, ce qui est encore une fois justifié par le fait de chercher des motifs réellement interprétables. Le rapport moyen entre le nombre de motifs détectés par chaque algorithme est détaillé dans le tableau 4.2.

Comparaison	Rapport moyen
SIA / Algo 1	902.784
SIATEC / Algo 1	548.415
SIA / Algo 2	196.138
SIATEC / Algo 2	116.265

TABLE 4.2 – Rapport moyen de réduction du nombre de motifs détectés par nos algorithmes par rapport à SIA et SIATEC (dans *Polygondwanaland*).

Le nombre de motifs détectés pour chaque piste de l'album est détaillé en annexe B.

Complexité computationnelle Les algorithmes SIA et SIATEC reposent sur le calcul de tables de vecteurs de translation entre toutes les paires de points de l'ensemble musical X, entraînant une complexité algorithmique au minimum quadratique : $\mathcal{O}(|X|^2 \times log_2(|X|))$ pour SIA [3] et $\mathcal{O}(|X|^3)$ pour SIATEC [3]. Leur approche exhaustive permet une couverture complète de toutes les classes d'équivalence par translation, mais au prix d'une charge computationnelle importante. À l'inverse, dans les algorithmes morphologiques développés dans ce stage, l'érosion ne parcourt que l'ensemble initial X à chaque itération. Ainsi, ils ont une complexité linéaire par valeur de L. Pour le second algorithme, malgré le post-traitement des motifs, l'idée permettant de travailler uniquement sur les sous-ensemble érodés permet de garder cette complexité. Nos deux algorithmes ont donc un coût total de $\mathcal{O}(|X| \cdot |\mathcal{L}|)$ ($\mathcal{L}$ étant la liste des valeurs de L testées).

Interprétabilité musicale L'un des objectifs des algorithmes développés est d'extraire des informations interprétables musicalement. Les algorithmes de Meredith et de ses collaborateurs n'ont pas d'informations supplémentaires clairement accessibles, car ils n'ont pas été conçus dans cet objectif. Ici, pour chaque motif, nous avons donc deux informations intéressantes pour une analyse musicale :

— N : le nombre de répétitions, qui est une valeur pertinente d'un point de vue musicologique. En effet, lorsqu'un musicien joue ou appréhende une pièce musicale, c'est bien de ce type d'informations dont il a besoin.
— L : la durée du motif, qui est interprétable comme la valeur du haut de la signature rythmique, ce qui donne une information sur la construction de la pièce musicale.

Pour résumer, nous avons donc des algorithmes qui font une sélection de motifs **plus pertinents**, de manière **plus rapide** et **plus interprétable** dans notre contexte d'étude.

Chapitre 5

Extraction d'informations musicales

Après avoir vu qu'il était possible de détecter des motifs et d'en extraire des informations musicales, le tout informé par la morphologie mathématique, l'idée suivante est de se servir de ces résultats afin d'avoir des affichages clairs, lisibles, et pédagogiques nous donnant des informations structurelles sur une pièce musicale choisie. L'objectif est d'avoir un programme entièrement automatisé qui reçoit un fichier MIDI et qui en ressort des figures informatives et intelligibles par un musicologue sans forcément demander une connaissance sur les notions de morphologie mathématique utilisées. Cette partie a donc pour objectif de détailler les recherches qui ont été faites dans cette voie et de montrer les résultats obtenus.

5.1 Détection automatique de signatures rythmiques

La première approche est donc d'afficher une représentation de l'évolution de la signature rythmique lors de la pièce musicale. Pour cela, est utilisée, pour chaque motif, sur la plage de temps correspondant, la valeur de L associée. Ainsi, il devrait pouvoir être observée l'évolution au cours du temps de la signature rythmique utilisée dans la pièce musicale. La figure 5.1 nous montre le résultat obtenu par cette méthode.

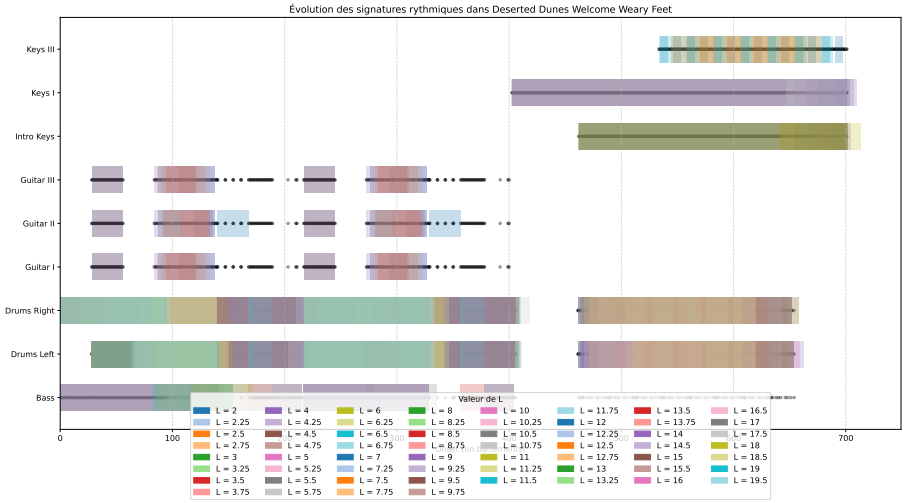


FIGURE 5.1 – Visualisation naïve de l'évolution des signatures rythmiques dans *Deserted Dunes Welcome Weary Feet*.

Cette visualisation met en évidence un problème clair qui rend cette figure illisible : il y a de la **superposition** entre les différents motifs. En effet, un même point (une même note) peut être détecté dans plusieurs motifs. Ainsi, apparaissent des recouvrements entre différents motifs, il est donc impossible d'interpréter correctement cette figure.

5.1.1 Critères

Il manque donc un critère qui permette de choisir, parmi plusieurs motifs candidats, lequel est le meilleur, celui qui représente le mieux la pièce musicale. Le critère choisi est, encore une fois, inspiré des algorithmes proposés par Meredith avec la notion de score de compacité, introduit dans COSIATEC [9]. Ce critère permet de classer des motifs en fonction de leur taux de compression (i.e. *leur capacité à représenter un maximum de points avec le minimum d'informations*). Lorsqu'il y a plusieurs motifs qui se superposent, le motif choisi est donc celui qui a la compacité la plus élevée (le meilleur taux de compression), selon la formule suivante :

$$\text{compacité}(P, N) = \frac{|P| \times N}{|P| + 2 + \varepsilon},$$

où ε est le nombre de points de X dans la plage de temps de $\mathcal{D} = P \oplus \{(n \cdot L, 0)\}_{n=0}^{N}$ mais qui ne sont pas couverts par $\mathcal{D}$. Le terme constant $+2$ représente le nombre d'informations dont on a besoin pour représenter notre élément structurant (ici N et L).

Cette définition de la compacité est construite de manière analogue à celle de COSIATEC [9]. Nous appliquons donc ce critère de manière itérative, comme détaillé dans le pseudocode de l'Algorithme 5.

Algorithme 5 Filtrage des segments par compacité

Require: Liste de triplets (P_j, N_j, L) ; Nombre d'itérations maximal n_{max}
1: Construire pour chaque triplet un segment $S_j = [a_j, b_j]$, où :
— $a_j = \min\{t \mid (t, \cdot) \in P_j\}$
— $b_j = a_j + N_j \cdot L$
2: Trier les segments S_j par ordre croissant de a_j
3: **for all** paires de segments S_1, S_2 qui se chevauchent **do**
4: **if** compacité$(S_1) >$ compacité(S_2) **then**
5: Retirer les points communs de S_2
6: Recalculer la compacité de S_2
7: **if** longueur de $S_2 < L$ **then**
8: Supprimer S_2
9: **end if**
10: **end if**
11: **end for**
12: Répéter jusqu'à stabilisation ou jusqu'à un nombre maximal d'itérations n_{max}
13: **return** Ensemble filtré des segments S_j associés aux motifs

Il est important de noter qu'on supprime tous les segments candidats dès lors qu'ils ont une longueur inférieure à L, car nous considérons qu'ils ne sont plus interprétables musicalement.

En sortie de cet algorithme, chaque note n'est donc associé qu'à un seul motif, ce qui permet donc de n'avoir qu'une seule signature rythmique par segment d'instrument de la pièce musicale. De plus, certains critères ont été ajoutés simplement pour l'affichage (n'affectant donc pas les motifs extraits) afin d'avoir une figure plus lisible :

- Une fois les meilleurs segments sélectionnés, on fusionne les intervalles consécutifs $[a, b]$ et $[c, d]$ si :
  - les deux segments ont le même L ;
  - il n'y a pas de segment avec un L différent entre les deux ;
- les longueurs L des segments sont regroupées par classes d'équivalence définies par des rapports entiers. Deux longueurs L_1 et L_2 sont considérées équivalentes s'il existe un entier k tel que $L_1 = k \cdot L_2$ ou $L_2 = k \cdot L_1$. La couleur associée dans les figures est déterminée par la plus petite valeur entière L_{base} du groupe, si elle existe. Si aucun entier n'est présent, on prend simplement la plus petite valeur du groupe.

5.1.2 Résultats

La figure 5.2 nous permet de visualiser ce qu'on obtient après que tous les critères ont été appliqués aux motifs détectés par le second algorithme.

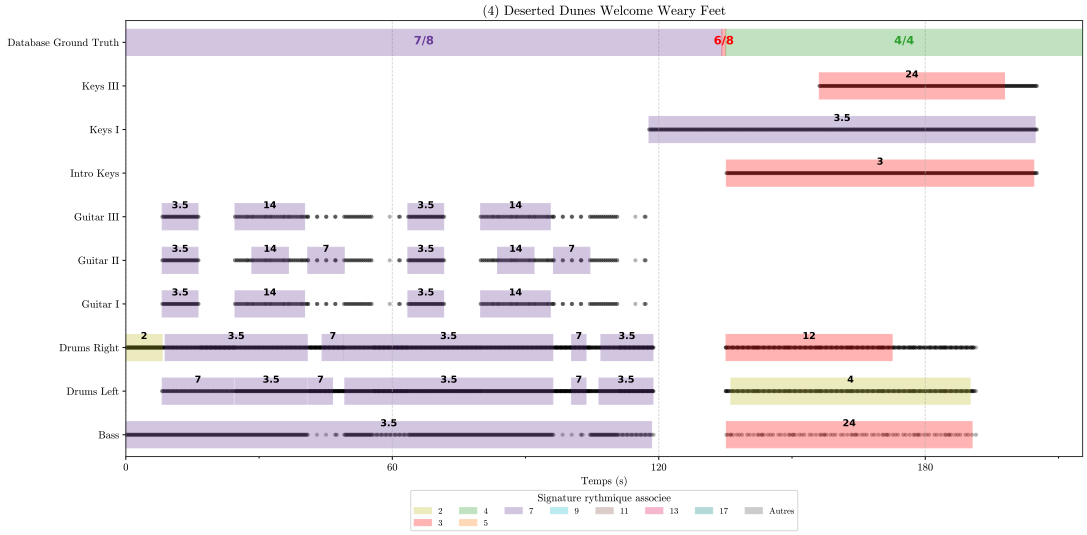


FIGURE 5.2 – Visualisation de l'évolution des signatures rythmiques dans *Deserted Dunes Welcome Weary Feet*.

Les informations y sont claires, lisibles et informatives. Nous avons des informations structurelles sur la pièce musicale, car nous pouvons voir l'évolution de la signature rythmique lors de la pièce, et nous pouvons y deviner les différentes parties de cette dernière. En particulier, dans cette figure nous pouvons voir que lors de la première partie de la pièce musicale, la valeur supérieure de la signature est 7, ce qui est confirmé par la «*Ground Truth*» (valeur annotée fournie dans la base de données) qui est en 7/4. Ces valeurs de signatures rythmiques étant indépendantes par instrument, nous pouvons alors facilement observer les polymétries, comme ici dans la deuxième partie de la pièce où il y a des polymétries entre du 3, du 4 et du 7.

Cette méthode informée par la morphologie mathématique nous permet donc automatiquement et rapidement d'analyser une pièce musicale et de proposer une visualisation claire d'informations qui peuvent être difficiles à entendre. En effet, dans une pièce musicale où plusieurs polymétries se superposent, il devient très compliqué d'entendre les différentes signatures rythmiques, même pour un expert, alors que pour cet algorithme, la complexité de la construction de la pièce ne change rien à l'analyse. Un bon exemple pour illustrer ce phénomène est celui illustré en figure 5.3.

Cette représentation permet clairement de voir de nombreuses polyrythmies, ici en superposant du 9, 10, 11, 12, et du 13. Une polymétrie entre 5 signatures rythmiques qu'il est quasi impossible à entendre sans écouter les instruments séparément. Or, notre représentation permet d'avoir cette information très facilement, alors même qu'aucune trace d'une telle polymétrie sur ce morceau n'a été mise en avant en ligne. Il semble donc que personne n'a réussi à repérer quelles étaient les polymétries en jeu dans cette pièce musicale avant que ces travaux permettent de le voir de manière élégante informée par la morphologie mathématique.

Importance des valeurs testées Nous avons remarqué en travaillant avec ces figures qu'elles étaient dépendantes des valeurs de L que nous testions. En effet, il est possible que la valeur la plus pertinente soit supérieure aux valeurs testées. Dans ce cas alors, en testant plus de valeurs nous pourrions découvrir soit de nouveau motifs, soit une nouvelle signature rythmique dominante un segment, auparavant "mal" évalué. Pour cette même pièce musicale, *Searching*, si le nombre de valeurs de L testées est augmenté, passant de $L_{max} = 20$ à $L_{max} = 50$, il est obtenu ce qui est illustré en figure 5.4.

On observe que dans deux pistes (Percussion 2 et Percussion 4), la valeur dominante de L est maintenant 28, ce qui n'était pas le cas précédemment puisque cette valeur n'était pas testée. Cette observation pour-

FIGURE 5.3 – Visualisation de l'évolution des signatures rythmiques dans *Searching*.

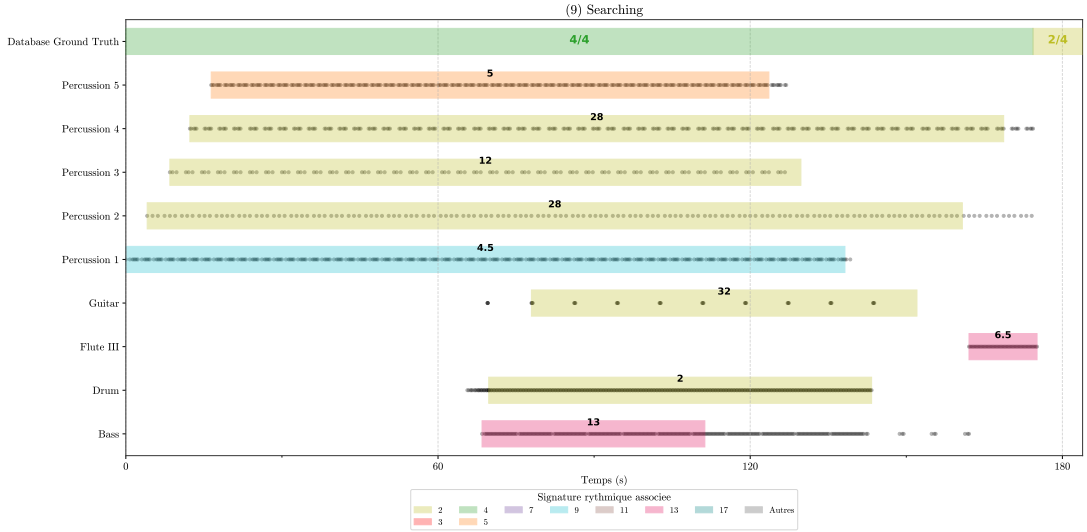


FIGURE 5.4 – Visualisation de l'évolution des signatures rythmiques dans *Searching* pour $L_{max} = 50$.

rait faire penser que nos résultats antérieurs étaient incohérents et remettre en question notre approche théorique. Cependant, elle nous apporte en réalité une information supplémentaire importante. En effet, quand on utilise la polymétrie dans une pièce musicale, il faut que les différentes signatures rythmiques se réalignent périodiquement pour que l'auditeur ne se perde pas dans le rythme. Ce réalignement se produit généralement au plus petit commun multiple (PPCM) de chaque signature rythmique, comme le montre la figure 2.5, mais dans notre cas, avec cinq signatures rythmiques différentes, le PPCM serait beaucoup trop grand pour être utilisable musicalement. Ce que nous obtenons en augmentant la valeur de L_{max} révèle donc le nombre de temps entre deux réalignements : les différents instruments se réalignent tous les 28 temps. Cela indique l'utilisation d'une technique appelée réalignement forcé, qui permet de réaligner artificiellement les différentes pistes en coupant la fin d'un motif plutôt qu'en attendant le PPCM naturel. Cette technique est notamment utilisée par le groupe Meshuggah [19], connu pour son travail avec la polymétrie complexe. Ainsi, le fait d'obtenir des résultats différents en changeant L_{max} n'est pas incohérent. Il est courant en analyse musicale d'avoir des interprétations différentes selon l'échelle d'observation. Ici, nous identifions à la fois la signature rythmique de base de chaque motif et la période de réalignement global, deux niveaux d'organisation complémentaires qui nous permettent de mieux comprendre la structure de l'œuvre.

5.2 Répartition métrique

De ces motifs détectés, chacun étant associé à une valeur de signature rythmique, il est donc possible de connaître la répartition des signatures rythmiques et de visualiser la complexité rythmique d'une pièce donnée. En particulier, nous pourrions essayer de voir quelles sont les polymétries en jeu et quelles proportions du morceau elles représentent. Pour cela, nous avons donc développé un programme qui récupère les motifs détectés et filtrés par compacité, et qui compare les pistes deux à deux afin de connaître les homométries ou polymétries dans la pièce musicale.

5.2.1 Méthode de calcul

L'algorithme développé procède en plusieurs étapes pour quantifier la complexité polymétrique d'une pièce musicale :

Préparation des données Les segments rythmiques détectés précédemment sont organisés par instrument, chacun étant caractérisé par :
- sa signature rythmique L et sa base multiplicative associée,
- ses bornes temporelles (début et fin du segment),
- sa durée.

Calcul du pourcentage «sans signature» Un point clé de notre approche consiste à calculer d'abord la proportion de la pièce où l'information de signature rythmique n'est pas détectable (pas de motif détecté). Cette proportion est calculée en comparant :
- la durée totale cumulée de tous les instruments,
- la durée cumulée couverte par des segments avec signature rythmique détectée.

Cette étape permet de distinguer les zones d'activité rythmique structurée des zones sans motif rythmique identifiable.

Comparaisons croisées des segments Pour chaque paire d'instruments, l'algorithme examine tous les segments temporels où les deux instruments sont actifs simultanément. Pour chaque chevauchement temporel, il détermine :
- si les deux instruments partagent la même base rythmique (homométrie),
- si ils utilisent des bases rythmiques différentes (polymétrie),
- la durée de ce chevauchement.

Construction de la matrice de complexité Les résultats sont agrégés dans une matrice triangulaire où :
- la diagonale représente les durées d'homométrie (même signature rythmique),
- les éléments hors diagonale représentent les durées de polymétrie,
- une case spéciale "Sans signature" quantifie les zones sans structure rythmique.

Normalisation et scores La matrice est normalisée pour que la somme de tous ses éléments soit égale à 1, permettant une interprétation en termes de proportions temporelles. De plus, un score de polymétrie est calculé :

$$\text{Score de complexité} = \frac{\text{Temps polymétrique total}}{\text{Temps avec signature total}}$$

Le score de complexité varie de 0 (purement homométrique) à 1 (purement polymétrique).

Cette représentation permet d'identifier rapidement les signatures rythmiques dominantes, les polymétries les plus fréquentes, et d'évaluer la complexité rythmique globale de la pièce.

5.2.2 Résultats

Des figures nous permettant de voir facilement la répartition de signatures rythmiques d'une pièce musicale donnée sont obtenues, comme illustré à la figure 5.5.

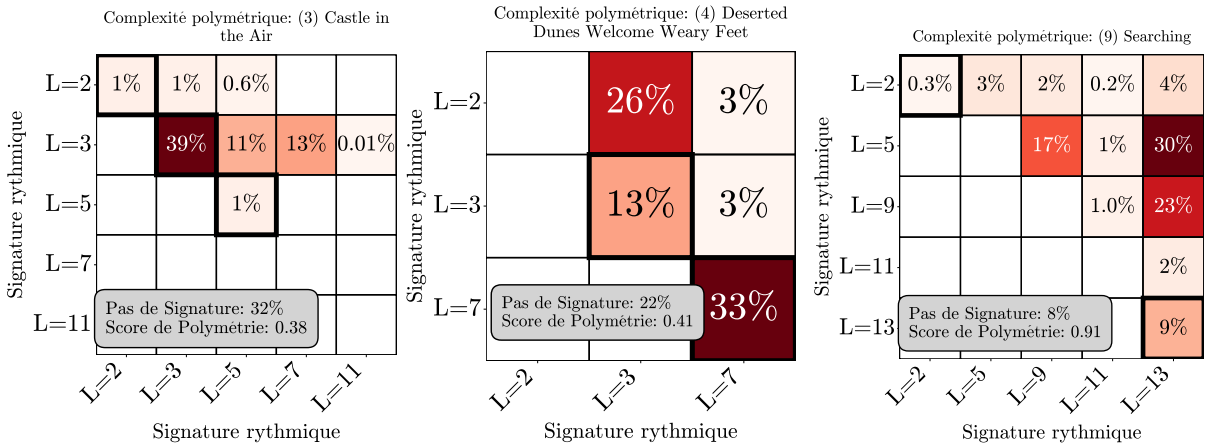


FIGURE 5.5 – Visualisation des matrices de complexités polymétriques pour 3 pièces musicales de *Polygondwanaland* (diagonale en **gras** = homométrie, le reste = polymétrie).

Remarque : Il est important de rappeler que l'absence de signature (« Pas de Signature ») correspond aux segments communs entre deux instruments où **au moins** un des deux instruments n'a pas de signature rythmique trouvée grâce à un motif. Toutes les grilles correspondant aux pièces musicales restantes de l'album sont en annexe D.

Chapitre 6

Conclusion et perspectives

L'ensemble de ces travaux montre des résultats intéressants car efficaces et informatifs d'un point de vue musical. Ils permettent de montrer que l'approche théorique développée par Lascabettes [1] peut s'appliquer sur des bases de données concrètes et de valider les résultats qu'ils proposent. De plus, nous avons pu montrer que l'approche morphologique permet d'avoir des algorithmes plus rapides que ce qui existe et que des informations musicales supplémentaires peuvent en être extraites.

Cependant, il est important de noter que la base de données sur laquelle nous avons travaillé est particulièrement adaptée à notre cadre d'étude. En effet, nos résultats ont été trouvés sur un album de King Gizzard & the Lizard Wizard qui utilise de nombreuses superpositions de polymétrie, avec des signatures rythmiques peu usuelles. Ainsi, pour que l'auditeur ne soit pas trop perdu, il faut contrebalancer la complexité rythmique par une simplicité mélodique. C'est pourquoi nous avons énormément de répétitions exactes de motifs dans cet album, ce qui nous permet de détecter avec précision des motifs dans la quasi intégralité des pièces musicales de l'album. Mais les répétitions exactes ne sont pas systématiques et elles sont souvent seulement **approchées**. C'est pourquoi, on pourrait essayer de ne plus seulement avoir des outils morphologiques simples détectant les répétitions exactes, mais alors de prendre en compte de potentielles **variations**. Des travaux ont déjà été menés dans ce sens avec les répétitions à un demi-ton près [2] ou sur des pièces du *Clavier bien tempéré* de J.-S. Bach [20], et vont être prolongés dans ma thèse à venir sur la détection de motifs dans le cadre de la morphologie **floue** [21].

Ainsi, nous allons chercher à se demander quelles variations sont acceptables d'un point de vue musical et de définir des outils morphologiques plus subtils afin de détecter des répétitions approchées. Nous allons aussi chercher à définir des métriques permettant de comparer des motifs dans un contexte musical. Et enfin, nous chercherons à créer une base de données de pièces musicales annotées par motifs, ce qui manque cruellement dans la recherche musicologique à l'heure actuelle, et de l'intégrer potentiellement dans des plateformes dédiées [22].

Bibliographie

[1] Paul Lascabettes. « Mathematical Models for the Discovery of Musical Patterns, Structures and for Performances Analysis ». Thèse de doctorat. Sorbonne Université, 2023.

[2] Enguérand Tamagna. « Découverte de motifs musicaux avec variations en utilisant la morphologie mathématique ». Mémoire de master. Paris, France : Sorbonne Université, 2024.

[3] David Meredith, Kjell Lemström et Geraint A. Wiggins. « Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music ». In : *Journal of New Music Research* 31.4 (2002), p. 321-345.

[4] Nicolas Ruwet. *Langage, musique, poésie*. Paris : Éditions du Seuil, 1972.

[5] Fred Lerdahl et Ray S. Jackendoff. *A Generative Theory of Tonal Music*. Réédition avec nouvelle préface. MIT Press, 1996.

[6] Heinrich Schenker. *Harmony*. T. 1. University of Chicago Press, 1954.

[7] Brent Auerbach. *Musical Motives : A Theory and Method for Analyzing Shape in Music*. Oxford University Press, 2021.

[8] Berit Janssen. *Discovery of Repeated Themes & Sections*. `https://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections`. 2017.

[9] David Meredith. « Music Analysis and Point-Set Compression ». In : *Journal of New Music Research* 44.3 (2015), p. 245-270.

[10] Isabelle Bloch, Henk Heijmans et Christian Ronse. « Mathematical Morphology ». In : *Handbook of Spatial Logics*. Springer, 2007, p. 857-944.

[11] Paul Lascabettes et Isabelle Bloch. « Discovering Repeated Patterns From the Onsets in a Multidimensional Representation of Music ». In : *International Conference on Discrete Geometry and Mathematical Morphology*. Florence, Italie : Springer, 2024.

[12] Jessica M. Ross et Ramesh Balasubramaniam. « Time Perception for Musical Rhythms : Sensorimotor Perspectives on Entrainment, Simulation, and Prediction ». In : *Frontiers in Integrative Neuroscience* 16 (2022).

[13] Cecilie Møller et al. « Beat Perception in Polyrhythms : Time is Structured in Binary Units ». In : *PLOS One* 16.8 (2021).

[14] Gabe Hardman et Patricia Talarczyk. *The Effects of Odd Time Signatures on Pop Song Enjoyment*. Journal of Student Research, 2021.

[15] King Gizzard & the Lizard Wizard. *Polygondwanaland*. `https://kinggizzardandthelizardwizard.com/polygondwanaland`. 2017.

[16] 8-bit Escapades. *Polygondwanaland*. `https://8-bitescapades.bandcamp.com/album/polygondwanaland`.

[17] Jeff A. Bilmes. *Techniques to Foster Drum Machine Expressivity*. Vision et Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1993.

[18] Vijay Iyer et al. « A Novel Representation for Rhythmic Structure ». In : *Proceedings of the 23rd International Computer Music Conference*. 1997, p. 97-100.

[19] Mathémusique. *MESHUGGAH et la Musique Algorithmique ?* `https://www.youtube.com/watch?v=euSki63c-SQ&t=39s`. 2023.

[20] Nils Quaetaert. « Discovery of Musical Patterns and Their Variations with Mathematical Morphology ». Mémoire de master. Strasbourg, France : School of Engineering Sciences, 2025.

[21] Isabelle Bloch et Henri Maitre. « Fuzzy mathematical morphology ». In : *Annals of Mathematics and Artificial Intelligence* 10 (1994), p. 55-84.

[22] Mathieu Giraud, Richard Groult et Emmanuel Leguy. « Dezrann, a web framework to share music analysis ». In : *International Conference on Technologies for Music Notation and Representation*. 2018, p. 104-110.

Annexes

Annexe A : Liste des pistes supprimées

Les pistes qui ont été supprimées car jugées non pertinentes sont les suivantes :

- Innercell – Keys II
- Polygondwanaland – Keys I
- Polygondwanaland – Intro Keys
- Deserted Dunes Welcome Weary Feet – Flute/Keys II
- Deserted Dunes Welcome Weary Feet – Bass Key
- Searching – Keys I
- Searching – Keys II
- The Fourth Colour – Low Bass Key

Annexe B : Nombre de motifs détectés pour chaque piste

Table 1 – Comparaison du nombre de motifs détectés par piste et par algorithme.

Track	**Nombre de motifs**	**algo_1**	**algo_2**	**SIA**	**SIATEC**
Tetrachromacy (Bass)		23	88	11133	8951
Tetrachromacy (Drums Left)		47	618	57065	33896
Tetrachromacy (Drums Right)		47	510	53218	32219
Tetrachromacy (Flute)		4	7	109	82
Tetrachromacy (Flute II)		1	1	65	51
Tetrachromacy (Flute III)		1	1	81	51
Tetrachromacy (Guitar I)		29	133	12760	10763
Tetrachromacy (Guitar II)		23	50	7463	4035
Tetrachromacy (Guitar III)		27	162	17455	14800
Tetrachromacy (Harmonica)		0	0	38	26
Tetrachromacy (Keys I)		0	0	7	6
Horology (Bass)		11	53	19127	12718
Horology (Drums Left)		40	405	28289	15774
Horology (Drums Right)		45	500	28021	15444
Horology (Flute)		11	44	1378	913
Horology (Flute II)		1	2	132	87
Horology (Flute III)		1	2	107	74
Horology (Guitar I)		19	198	26118	21504
Horology (Guitar III)		25	163	19070	16550
Horology (Keys I)		27	116	12647	11353
Castle in the Air (Bass)		6	37	2139	1133
Castle in the Air (Drums Left)		58	364	26378	14959
Castle in the Air (Drums Right)		23	87	12071	6482
Castle in the Air (Flute)		0	0	11	6
Castle in the Air (Guitar I)		8	29	4033	1250
Castle in the Air (Guitar II)		7	47	31378	14361
Castle in the Air (Guitar III)		16	79	13874	9826
Castle in the Air (Harmonica)		13	31	1466	379
Castle in the Air (Keys I)		18	69	12774	7164
Polygondwanaland (Bass)		13	105	18255	10923
Polygondwanaland (Drums Left)		33	455	24953	15617
Polygondwanaland (Drums Right)		39	633	26814	20569
Polygondwanaland (Flute)		2	3	355	213
Polygondwanaland (Flute/Keys II)		3	4	176	77
Polygondwanaland (Guitar II)		10	36	34986	17419
Polygondwanaland (Guitar III)		10	36	34986	17419
Deserted Dunes (Bass)		18	36	19128	14307
Deserted Dunes (Drums Left)		39	300	25385	18378
Deserted Dunes (Drums Right)		67	423	32014	24268

Suite page suivante

Track	Nombre de motifs	algo_1	algo_2	SIA	SIATEC
Deserted Dunes (Flute)		0	0	0	0
Deserted Dunes (Guitar I)		8	24	9737	6726
Deserted Dunes (Guitar II)		8	28	11746	7208
Deserted Dunes (Guitar III)		8	24	9737	6726
Deserted Dunes (Intro Keys)		6	12	2119	869
Deserted Dunes (Keys I)		5	10	1608	1608
Deserted Dunes (Keys III)		4	52	3278	2332
Deserted Dunes (Keys IV)		0	0	0	0
Innercell (Bass)		9	46	12735	7962
Innercell (Drums Left)		73	482	24869	13649
Innercell (Drums Right)		73	482	24869	13649
Innercell (Flute)		3	8	286	140
Innercell (Guitar II)		13	103	36995	24211
Innercell (Guitar III)		13	99	37193	24778
Innercell (Keys I)		13	32	12093	3036
Loyalty (Bass)		20	79	17050	9948
Loyalty (Drums Left)		45	510	26180	17841
Loyalty (Drums Right)		45	510	26180	17841
Loyalty (Flute)		3	5	665	290
Loyalty (Flute III)		2	6	3897	2264
Loyalty (Guitar I)		13	82	16507	10115
Loyalty (Guitar II)		0	0	51	33
Loyalty (Guitar III)		0	0	61	41
Loyalty (Keys I)		7	7	951	330
Searching (Bass)		18	107	2933	2059
Searching (Drum)		40	54	3065	868
Searching (Flute III)		1	1	267	144
Searching (Guitar)		1	1	122	53
Searching (Percussion 1)		13	104	1478	653
Searching (Percussion 2)		6	40	288	288
Searching (Percussion 3)		3	3	263	151
Searching (Percussion 4)		1	11	1460	1167
Searching (Percussion 5)		39	310	429	429

Annexe C : Signatures rythmiques des morceaux (pour $L_{max} = 50$)

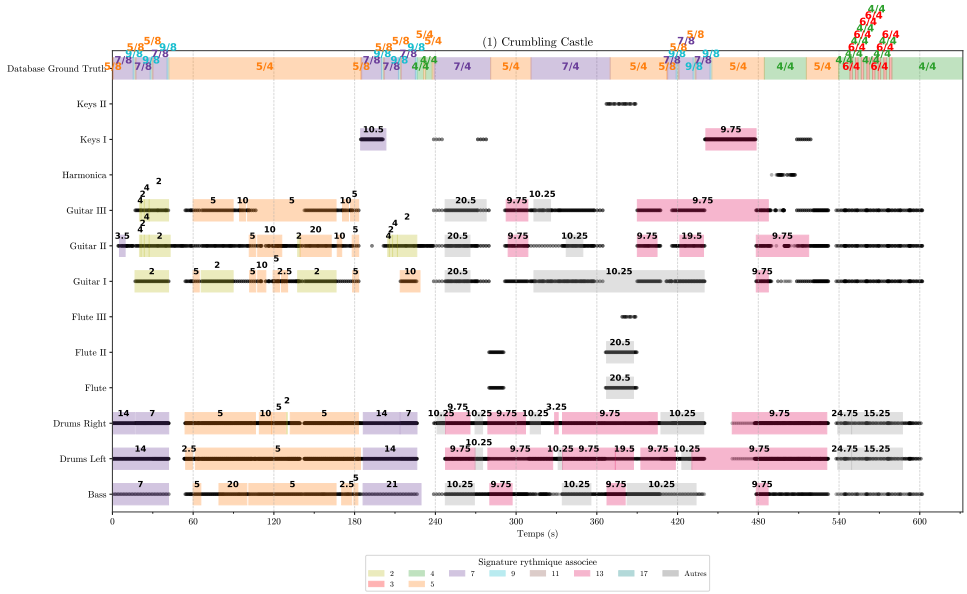


FIGURE 1 – Visualisation de l'évolution des signatures rythmiques dans *Crumbling Castle*

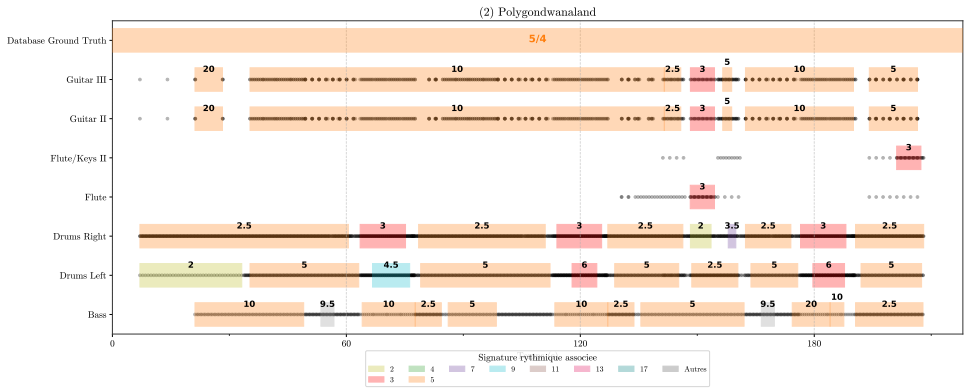


FIGURE 2 – Visualisation de l'évolution des signatures rythmiques dans *Polygondwanaland*

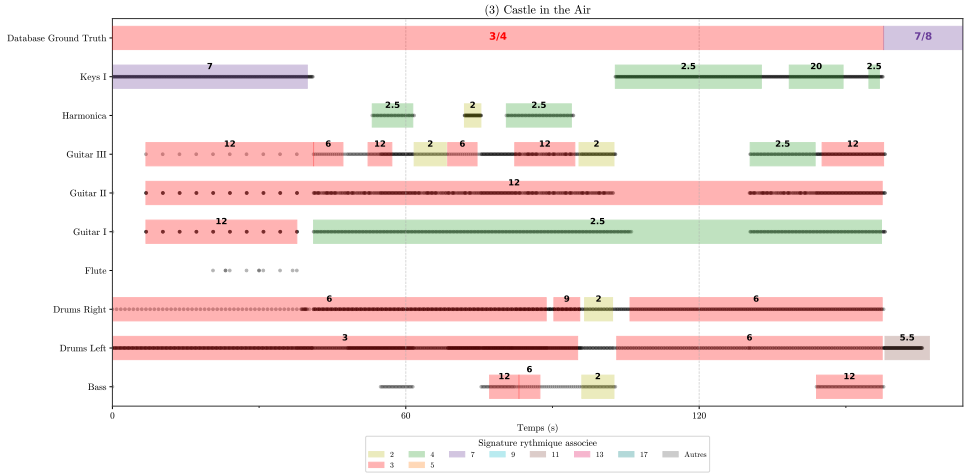


FIGURE 3 – Visualisation de l'évolution des signatures rythmiques dans *Castle in the Air*

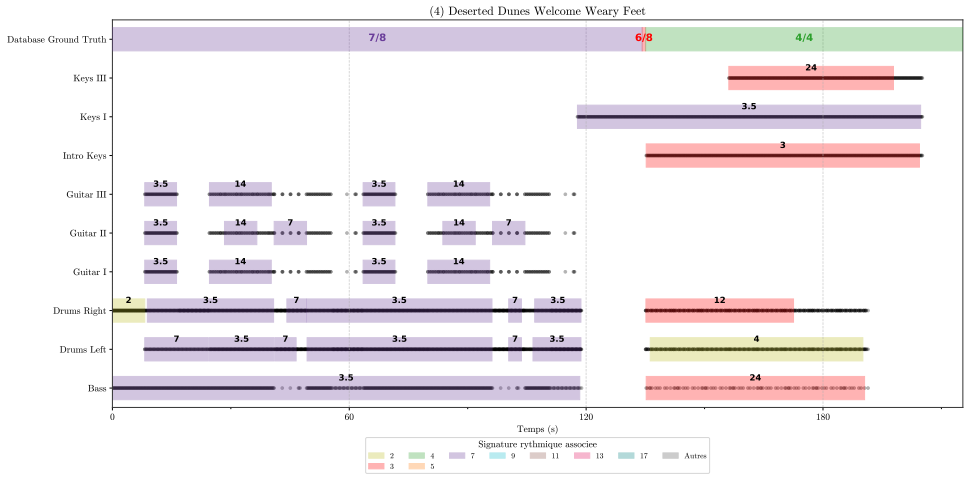


FIGURE 4 – Visualisation de l'évolution des signatures rythmiques dans *Deserted Dunes Welcome Weary Feet*

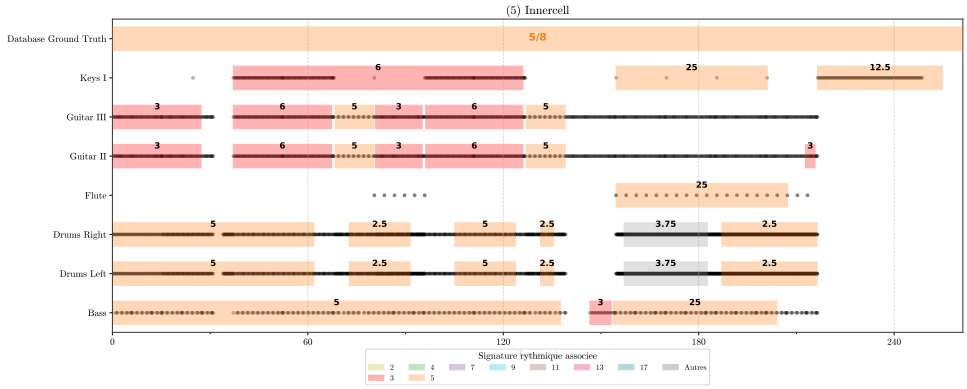


FIGURE 5 – Visualisation de l'évolution des signatures rythmiques dans *Innercell*

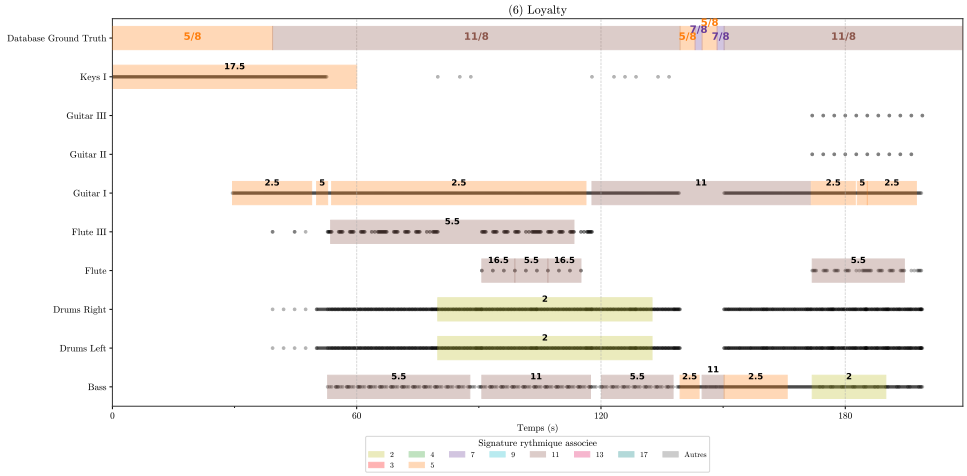


FIGURE 6 – Visualisation de l'évolution des signatures rythmiques dans *Loyalty*

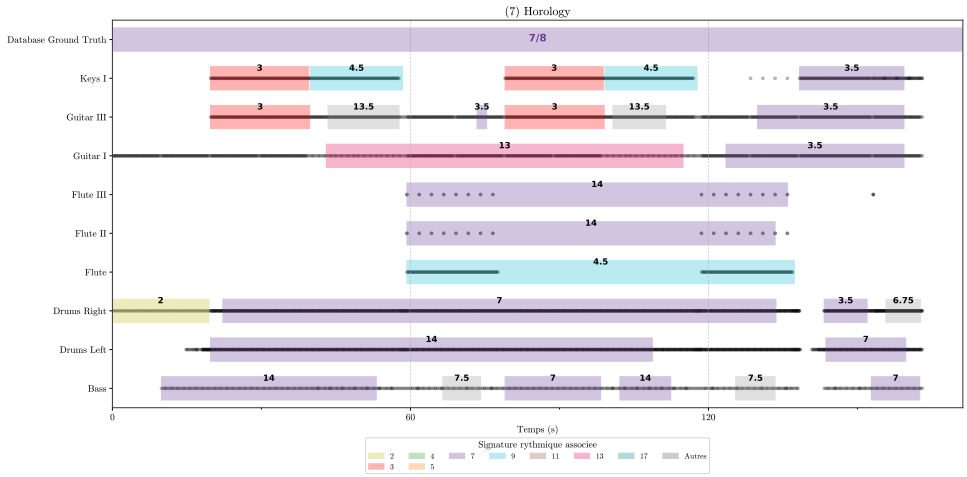


FIGURE 7 – Visualisation de l'évolution des signatures rythmiques dans *Horology*

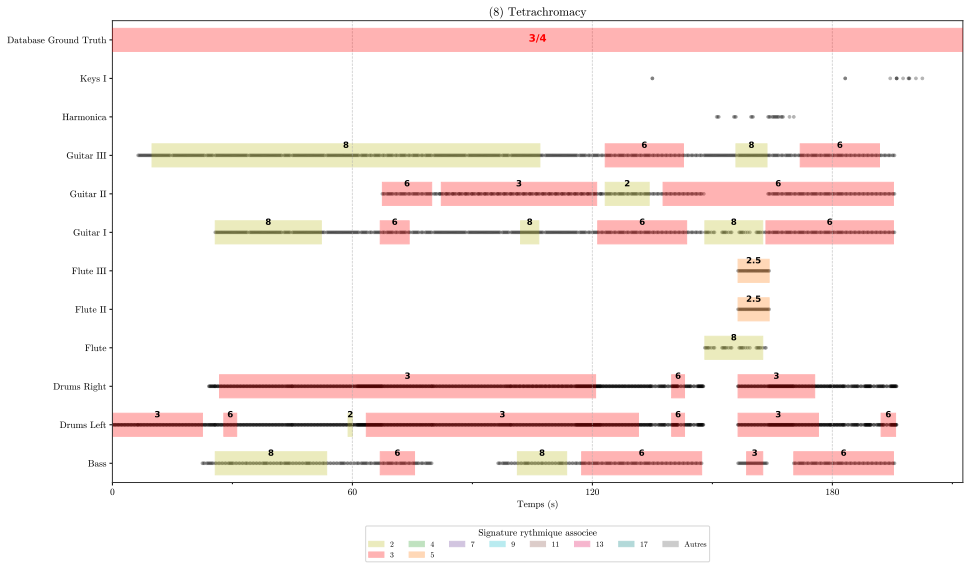


FIGURE 8 – Visualisation de l'évolution des signatures rythmiques dans *Tetrachromacy*

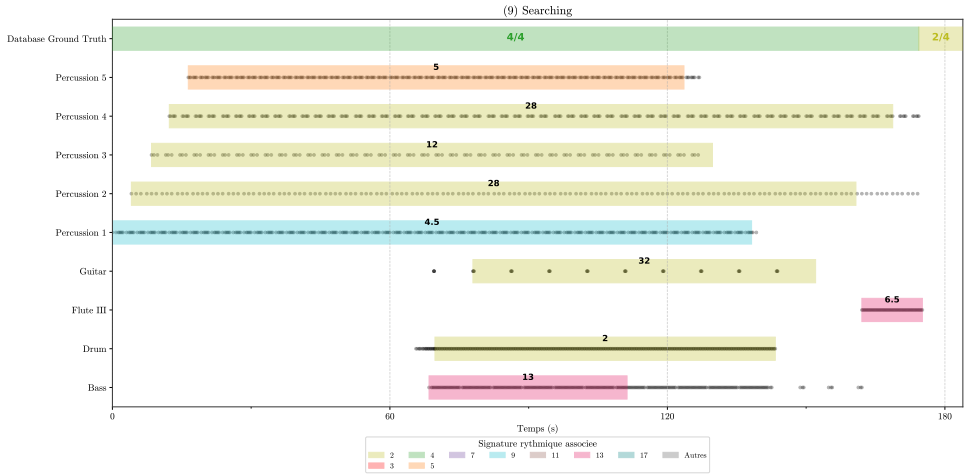
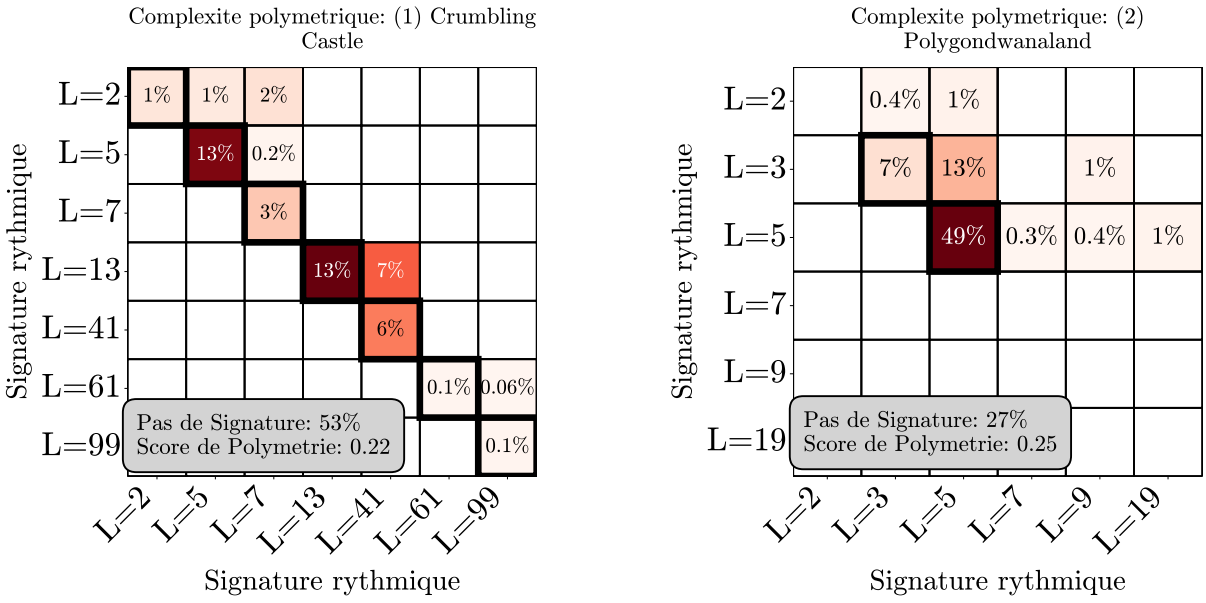


FIGURE 9 – Visualisation de l'évolution des signatures rythmiques dans *Searching*

FIGURE 10 – Visualisation de l'évolution des signatures rythmiques dans *Searching*

Annexe D : Matrices de complexités polymétriques

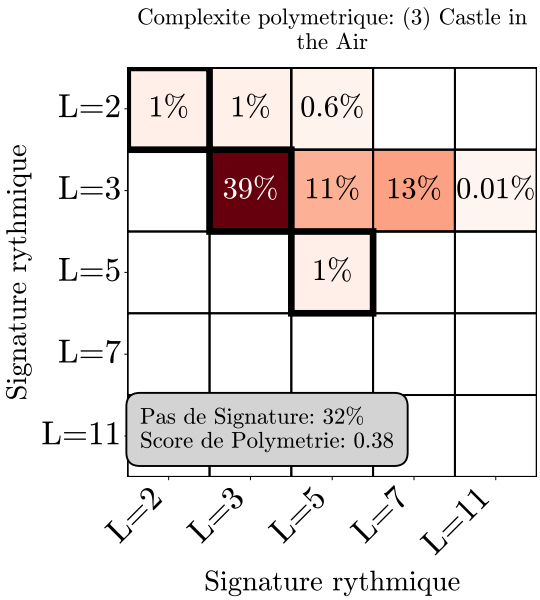


(1) *Crumbling Castle*

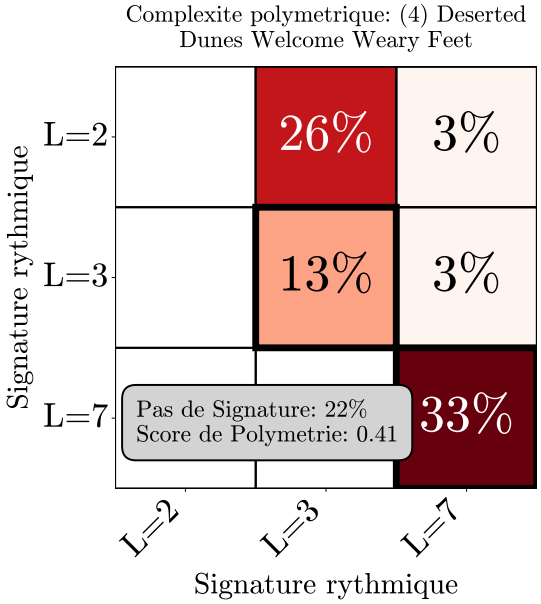
(2) *Polygondwanaland*

FIGURE 11 – Matrices de complexité polymétrique (1–2).
Diagonale en **gras** = homométrie, le reste = polymétrie.

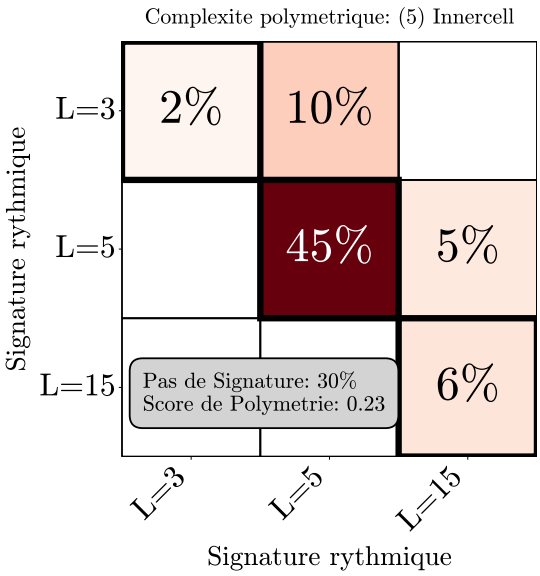
Remarque : Les valeurs de L supérieures à L_{max} viennent du fait que les valeurs non-entières de L sont associées à leur multiple entier le plus proche (exemple : $L = 30,5$ qui est associé à $L = 61$ $(= 30,5 \times 2)$.



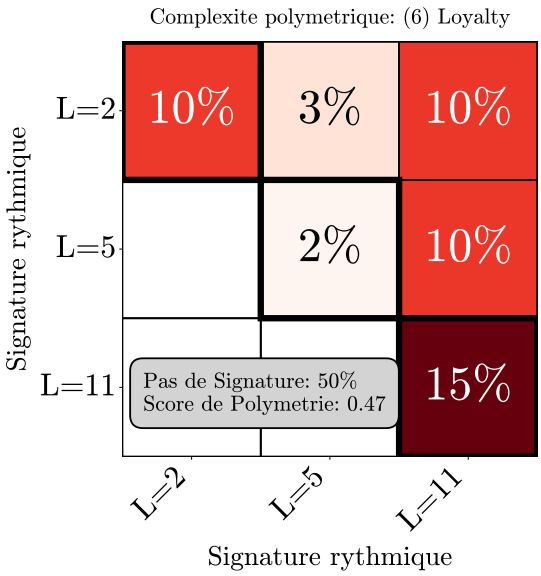
(3) *Castle in the Air*



(4) *Deserted Dunes Welcome Weary Feet*

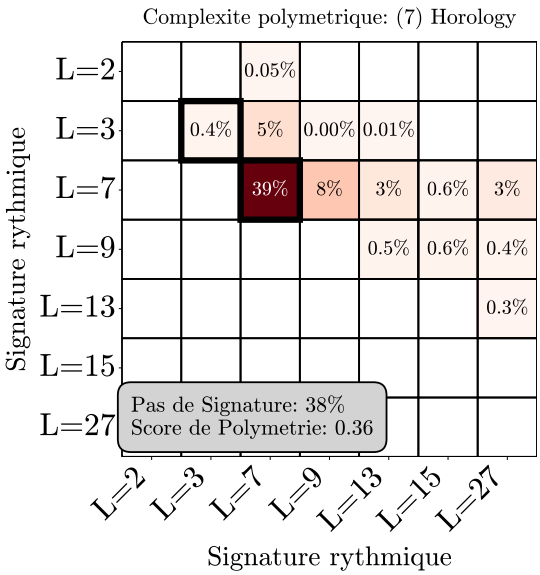


(5) *Innercell*

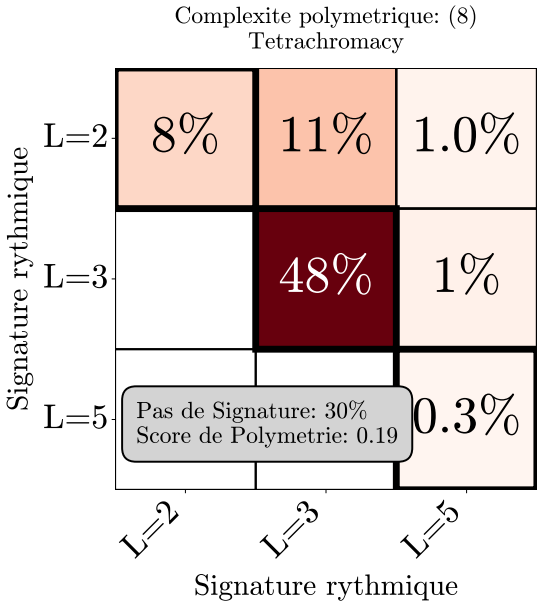


(6) *Loyalty*

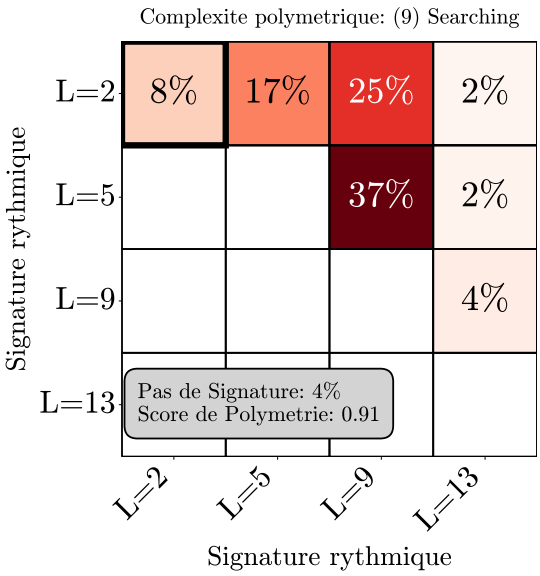
FIGURE 12 – Matrices de complexité polymétrique (3–6).



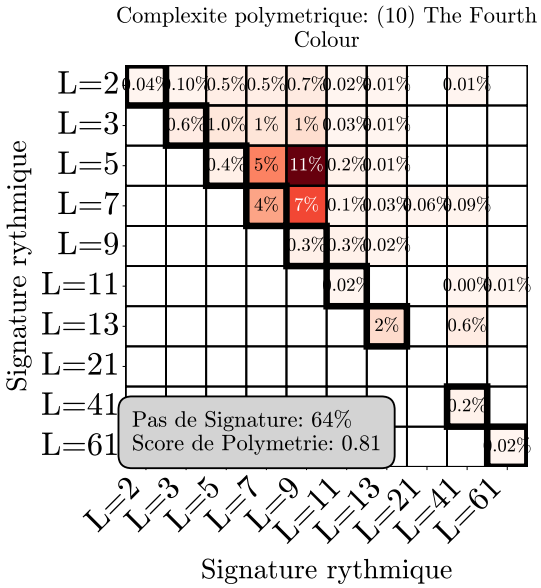
(7) *Horology*



(8) *Tetrachromacy*



(9) *Searching*



(10) *The Fourth Colour*

FIGURE 13 – Matrices de complexité polymétrique (7–10).