THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS VI

Spécialité : **Informatique**

Option : **Intelligence Artificielle**

présentée par

Pierre-Yves ROLLAND

pour obtenir le grade de

Docteur de l'Université Paris VI

Sujet de la thèse :

DECOUVERTE AUTOMATIQUE DE REGULARITES DANS LES SEQUENCES ET APPLICATION A L'ANALYSE MUSICALE

Soutenue le 20 juillet 1998

Devant le jury composé de :

Mme Bernadette BOUCHON-MEUNIER	*Examinatrice*
M. Maxime CROCHEMORE	*Examinateur*
M. Jean-Gabriel GANASCIA	*Directeur*
M. François PACHET	*Examinateur*
M. Bruce PENNYCOOK	*Rapporteur*
M. Henry SOLDANO	*Examinateur*
M. Gerhard WIDMER	*Rapporteur*

A Anne-Laure

REMERCIEMENTS

Ma profonde gratitude va en premier lieu à Jean-Gabriel Ganascia, qui a dirigé ma thèse avec constance, pertinence et gentillesse. Les conseils et les idées dont il m'a fait part, au plan tant scientifique que méthodologique, tout au long de ces années de doctorat ont été irremplaçables. Je lui exprime mes sincères remerciements pour le temps qu'il a toujours su consacrer à nos entrevues malgré un planning plus que chargé, pour sa passion de (toute) la science et son enthousiasme contagieux et pour sa confiance, qui m'a accompagné depuis les prémisses de ce travail.

Je remercie chaleureusement Gerhard Widmer d'avoir bien voulu être rapporteur de cette thèse. Ses commentaires élogieux et ses suggestions fécondes me sont précieux.

Bruce Pennycook a lui aussi accepté la difficile tâche de rapporteur. Qu'il trouve ici l'expression de mes sincères remerciements.

Je remercie vivement Maxime Crochemore de son intérêt pour mes travaux et d'avoir bien voulu présider le jury de ma thèse. J'en suis honoré et reconnaissant.

Un immense merci à Henri Soldano, qui m'a fait l'honneur de participer à mon jury de thèse. Sa relecture attentive du manuscrit, ses suggestions et ses encouragements ont joué un rôle clé dans la réussite du projet.

Mes remerciements vont aussi à François Pachet pour sa participation à mon jury de thèse et pour nos collaborations scientifiques depuis le DEA.

Je remercie Bernadette Bouchon-Meunier de l'intérêt qu'elle a manifesté à l'égard de mon travail en participant au jury.

J’adresse des remerciements particuliers à Gerald Faussurier, Christophe Meyer, Raphaël Rispoli et Jean-Daniel Zucker qui ont eu la gentillesse de relire mon manuscrit. Le présent document a pu mettre à profit leurs remarques toujours pertinentes.

L’équipe ACASA du LIP6 est un foyer scientifique actif au sein duquel il est stimulant de préparer une thèse de doctorat. J’en remercie ici les membres et ex-membres pour les discussions scientifiques enrichissantes que nous avons eues pendant ces années de thèse, et aussi pour leurs encouragements et leur sympathie : Quyên Kieu, Mourad Sefrioui, Geber Ramalho, Irina Akoulchina, Frédéric Thiré…

Merci aux ex-co-insulaires du couloir 66-56 pour ces années de cohabitation amicale : Ocaï Caetano, Alexis Drogoul, Guillaume Hutzler, Pierre Vinant,… Je n’oublie pas non plus les continentaux de 46-0, saluant en particulier Bruno Lesueur et Christophe Marsala pour leur dévouement sans pareil.

Merci à nos ingénieurs, techniciens et administratifs, dont le dévouement et la sympathie permettent que le laboratoire soit un ensemble de personnes et de machines à la fois connectées et administrées… Et merci à Marcelle Philiba pour son professionnalisme et sa gentillesse.

Merci à tous les membres de l’ex-LAFORIA qui, d’une manière ou d’une autre, ont pris une part active dans l’aboutissement de ma thèse : Jean-Pierre Briot, Zahia Guessoum, Patrice Perny, Jean-Charles Pomerol en particulier.

L’atelier de Bioinformatique est un endroit hors du commun à de nombreux points de vue. Ses « habitants », Alain Viari et Joël Pothier en particulier, savent le rendre aussi accueillant que scientifiquement excellent. Qu’ils en soient ici remerciés.

Une personne mérite elle aussi tous les honneurs. Cette personne extraordinaire qui a choisi de partager ma vie a notamment, dans les derniers mois, relevé le challenge de vivre au rythme de l’existence ordinaire du doctorant rédigeant… Je remercie Anne-Laure pour sa patience et son impatience qui ont tour à tour été d’indispensables ingrédients à l’aboutissement de ce travail.

Je remercie ma petite Ma*n*man et ma grande Soeurette pour tout ce que je n’aurais jamais pu être ou faire sans elles. J’ajouterai, de façon plus pragmatique, que leurs relectures attentives du manuscrit ont été utiles autant que réconfortantes.

Last but not *liste*, ma gratitude va a mes amis, pour l’intérêt qu’ils ont manifesté à mes travaux et aussi pour être mes amis.

C‘est avec un grand plaisir que je remercie également toutes celles et ceux qui ne sont pas cités nommément ici mais qui ont, d’une façon ou d’une autre, contribué à la réussite de mon travail.

Merci à mon Père, qui a quitté ce monde il y a si longtemps. Ce travail est un peu le sien.

SOMMAIRE

NOTATIONS ET CONVENTIONS

Notations

- **R** désigne l’ensemble des réels
- **Z** désigne l’ensemble des entiers relatifs
- **N** désigne l’ensemble des entiers naturels ;

  Si k est un entier naturel, $\mathbf{N}_k$ désigne l’intervalle entier {0, 1, … k}
- **Bool** désigne l'ensemble {VRAI, FAUX} des booléens

N.B. La plupart des notations de base, relatives à l'algorithmique sur les séquences, utilisées dans ce mémoire sont regroupées à la section 2.2.1.

Convention typographique

La fin d'un exemple est marquée par une puce † (sauf si l'exemple est très bref)

Chapitre 1. INTRODUCTION

L'extraction de connaissances à partir de données ou ECD (*knowledge discovery in databases*), et la fouille de données (*data mining*) sont des thèmes de recherche connaissant depuis quelques années un essor important. L’un des objectifs poursuivis est d’automatiser — ou du moins d’assister — l’extraction d’informations pertinentes (connaissances) dans les quantités de plus en plus immenses de données produites, qu’elles soient de nature scientifique, industrielle ou commerciale (au sens large). On prend toute la mesure du problème en notant par exemple que les satellites d’observation tels que ceux de la NASA produisent de l’ordre du teraoctet de données par jour [Berndt & Clifford 1996] ; globalement, il est estimé que la quantité mondiale de données double tous les 20 mois [Frawley, Piatetsky-Shapiro & Matheus 1991].

Face à de telles quantité de données, il n’est plus envisageable que l’analyse en soit faite [exclusivement] par des humains. Les recherches menées dans les champs de l’ECD et de la fouille de données visent à développer des théories et des outils permettant de faire passer des données brutes, de bas niveau, dans des formes « plus compactes plus abstraites ou plus utiles », pour reprendre la formulation de Fayyad, Piatetsky-Shapiro & Smyth (1996a). Un certain nombre de résultats probants ont été obtenus sur des applications tant scientifiques [Fayyad, Haussler & Stolorz 1996] que commerciales/industrielles [Agrawal et al. 1996 ; Mannila, Toivonen & Verkamo 1997].

D'une manière générale, l’ECD peut être définie comme :

> "the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data." [Fayyad, Piatetsky-Shapiro & Smyth 1996a,b]

Dans cette définition, un *pattern* décrit un sous-ensemble des données, ou un modèle conforme à un tel sous-ensemble. L'ECD est vue comme un **processus** prenant en entrée les collections de données brutes et agissant sur elles au travers des étapes suivantes :

- *Sélection*
- *Prétraitement* et/ou « nettoyage » des données sélectionnées
- *Transformation* : celui-ci consiste couramment en une projection ou une réduction des données issues de l'étape précédente
- *Fouille de données* : celle-ci met en œuvre des algorithmes spécifiques produisant une énumération de patterns
- *Evaluation* et/ou *interprétation* de ces patterns

La fouille de données n'est donc qu'une étape, certes critique, d'un processus global. Parmi les paradigmes généraux dans lesquels s'inscrivent les algorithmes accomplissant cette étape on peut citer la classification, la régression, le regroupement (*clustering*) ou encore la modélisation de dépendances.

1.1. Découverte de régularités dans les séquences

Nos travaux sur la découverte de régularités s'inscrivent dans le cadre général de l'ECD présenté ci-dessus. Nous nous concentrons, comme dans la grande majorité des travaux du domaine, sur la découverte, dans des séquences, de patterns qui sont eux aussi des entités séquentielles. Nous parlons alors de *patterns séquentiels*. (Un contre-exemple de ce cas de figure sera présenté au 2.1.3.1 au travers des travaux de Widmer (1993 ; 1994) sur l'apprentissage automatique de règles d'expression pianistiques — dynamique, rubato).

La découverte de patterns séquentiels intervient sur des séquences issues de domaines divers qui seront présentés à la section 2.1, notamment :

- La biologie moléculaire (acides nucléiques, protéines)
- La linguistique (textes)
- Les agents « intelligents » d'interface et les jeux (séquences d'action utilisateur ou de coups joués)

- La musique (monodies, polyphonies, séquences d'accords)

Une formalisation à laquelle il est fréquemment fait appel consiste à désigner chaque élément de séquence distinct par un symbole appartenant à un *alphabet* donné. A toute séquence, dont les éléments peuvent être représentés de manière quelconque, on associe de manière bijective une séquence sur l'alphabet correspondant. Un alphabet, noté Σ, est un ensemble fini de symboles, appelés également *caractères* ou *lettres* [Crochemore & Rytter 1994]. Par exemple, en biologie l'une des représentations possibles d'une séquence protéique (resp. d'une séquence d'acide nucléique) est une chaîne de caractères sur un alphabet d'acides aminés avec $|\Sigma|=20$ (resp. sur un alphabet de nucléotides avec $|\Sigma|=4$). On parle alors de séquences *sur* Σ, et l'ensemble de toutes les séquences sur Σ est noté Σ^*. La découverte de régularités dans les séquences est alors vue comme la découverte de régularités dans des chaînes de symboles.

Un pattern séquentiel (appelé tout simplement dans la suite *pattern,* lorsqu'il n'y a pas d'ambiguïté), exprimé sur un certain langage, représente de manière intensionnelle un ensemble de séquences ou *mots.* Cet ensemble est naturellement la représentation extensionnelle (ou *extension*) du pattern. Plaçons-nous dans le contexte d'un ensemble de séquences C={S1, …, SN} (appelé *corpus*) dans lesquelles on cherche, soit à localiser de tels patterns donnés au départ, soit à extraire de nouveaux patterns. Les *facteurs* (segments contigus) des séquences Su appartenant à l'extension d'un pattern P sont appelés *occurrences* de P dans C.

Un exemple

Dans cet exemple très simple, nous nous plaçons sur l'alphabet « latin » $\Sigma^\circ = \{a, b, \ldots, z\}$. Considérons le corpus C constitué des trois séquences sur Σ présentées à la Figure 1-1. Nous supposons que les patterns sont également des mots sur Σ°.

```
S1 : mqlsdcabvnmqlkfjtablfkj

S2 : qdfjmlcaaklf

S3 : qlscabmlkjecabkjml
```

Figure 1-1. Corpus exemple

Supposons dans un premier temps que les occurrences d'un pattern P sont définies comme étant les facteurs identiques (égaux) à P. Alors le pattern P° = cab possède 3 occurrences, dites *exactes*, dans C : une dans S1, zéro dans S2 et deux dans S3. Chacune de ces occurrences, soulignées sur la Figure 1-2, peut être désignée par les *coordonnées* au sein de C du facteur correspondant, c'est-à-dire le triplet d'entiers constitué du numéro de la séquence, du *rang* du premier élément et de la *longueur*[1] (nombre d'éléments) du facteur. On obtient respectivement, pour les trois occurrences, (1; 6, 3), (3; 4, 3) et (3; 12, 3).

```
S1 : mqlsdcabvnmqlkfjtablfkj

S2 : qdfjmlcaaklf

S3 : qlscabmlkjecabkjml
```

Figure 1-2. Occurrences exactes du pattern P°=cab dans le corpus C={S1,S2,S3}

Supposons à présent que l'on choisisse un autre critère d'occurrence, plus tolérant que le précédent : cette fois-ci, une *erreur* (discordance) est permise sur l'un des éléments (au maximum) entre un pattern et une occurrence. A ce moment là, deux nouvelles occurrences de P°, appelées occurrences *approximatives,* apparaissent en plus des trois occurrences précédentes. Ces occurrences, de coordonnées respectives (1; 17, 3) et (2; 7, 3), sont soulignées en pointillé sur la Figure 1-3)

```
S1 : mqlsdcabvnmqlkfjtablfkj

S2 : qdfjmlcaaklf

S3 : qlscabmlkjecabkjml
```

Figure 1-3. Occurrences exactes du pattern P°=cab dans le corpus C={S1,S2,S3}

†

Spécificités

Si la découverte de régularités dans les séquences peut être inscrite dans le champ de l'ECD, il existe néanmoins un clivage net entre la découverte de patterns dans des données organisées en séquences et la découverte de patterns dans des données organisées différemment [Wang et al. 1994].

---

[1] Bien entendu, dans ce cas précis, compte tenu de la forme de la condition définissant l'existence d'une occurrence il suffit de connaître le **rang**, la longueur des facteurs-occurrences étant nécessairement égale à celle du pattern.

Pour ces dernières, il peut s'agir de bases de données relationnelles classiques [Ullman 1988 ; Korth & Silberschatz 1991], ou plus simplement de *bases de faits* comme souvent en intelligence artificielle (symbolique). La caractéristique importante est celle d'une organisation par enregistrements (*record-oriented*), chacun étant décrit typiquement par un ensemble de couples attribut-valeur.

C'est la relation d'ordre (total) existant sur les éléments des séquences qui induit le clivage en question [Wang et al. 1994, p. 115]. Ajoutons que, dans le contexte de la découverte de régularités, ce clivage subsiste même lorsque les éléments des séquences sont eux-mêmes décrits sous forme attributs-valeurs. Le fait que les éléments soient ordonnés les uns par rapport aux autres interdit en effet d'assimiler ces séquences à des bases de données relationnelles, ou à des bases de faits, standard. La raison principale est la suivante. Les régularités recherchées dans les bases de données « non séquentielles » sont typiquement des *corrélations* entre différents attributs. Supposons par exemple que l'on dispose d'une base de données sur un ensemble de consommateurs. Trouver des *patterns d'achat* tels que : « Si un(e) consommateur(trice) a acheté le produit A, il (elle) a une forte probabilité d'acheter les produits B et C dans les 3 mois qui suivent. » [Agrawal et al. 1996] peut rapporter gros à un responsable de marketing !

Pour extraire des patterns séquentiels dans des séquences (appelées *séquences sources*), schématiquement on ne compare pas des éléments individuels mais plutôt des entités (structures) séquentielles. Celles-ci sont souvent des *segments* des séquences sources, mais peuvent être des structures plus abstraites comme des expressions régulières. La comparaison, c'est-à-dire l'évaluation de la similitude, entre ces entités séquentielles est de nature *topologique*, faisant intervenir une *structure de correspondance* (« *mapping* ») entre les éléments respectifs des entités qui prend en compte la relation d'ordre existant entre les éléments (voir section 3.2). Par contraste, la comparaison entre deux enregistrements décrits sous forme d'un ensemble de couples attribut-valeur se fait généralement en comparant les valeurs deux à deux, attribut par attribut.

1.1.1. Découverte, localisation et extraction

Dans l'ECD on peut distinguer deux types d'objectifs. La *vérification* consiste à tester des hypothèses émises explicitement par l'utilisateur [Fayyad, Piatetsky-Shapiro & Smyth 1996a] ; au contraire, la *découverte* correspond à la recherche par le système de « nouveaux » patterns, de manière plus ou moins autonome.

Aussi distinguons-nous, au sein de la découverte de régularités dans les séquences :

- La *localisation* de patterns (précédemment extraits ou donnés au départ ; cf. par exemple [Berndt & Clifford 1996]). La problématique de la localisation peut être vue comme une forme particulière de reconnaissance de formes (cf. par exemple [Milgram 1993]).

- La découverte de (nouveaux) patterns.

Au sein du processus global, l'étape « fouille de données » (cf. ci-dessus) est ici désignée par *extraction de patterns*. La Figure 1-4 propose un schéma général de la découverte de régularités dans les séquences, lequel s'inspire du schéma général de l'ECD proposé dans [Fayyad, Piatetsky-Shapiro & Smyth 1996a]. Localisation et extraction de patterns se traduisent par deux chemins différents dans le « Y » horizontal formé par les flèches grises qui matérialisent les étapes successives du processus sur la figure :

- Branche supérieure du « Y » horizontal pour la localisation. Le tableau figurant dans la case en haut à droite traduit, à titre d'exemple, les résultats du dénombrement des occurrences de différents patterns (un par ligne) au sein de différentes séquences (une par colonne).

- Branche inférieure du « Y » horizontal pour l'extraction. A titre d'exemple, les dessins figurant dans la case en bas à droite évoquent les informations associées aux patterns extraits, par exemple une évaluation numérique de la qualité des différentes occurrences (telles que 0.8 pour la 3e occurrence dans la 1e séquence).

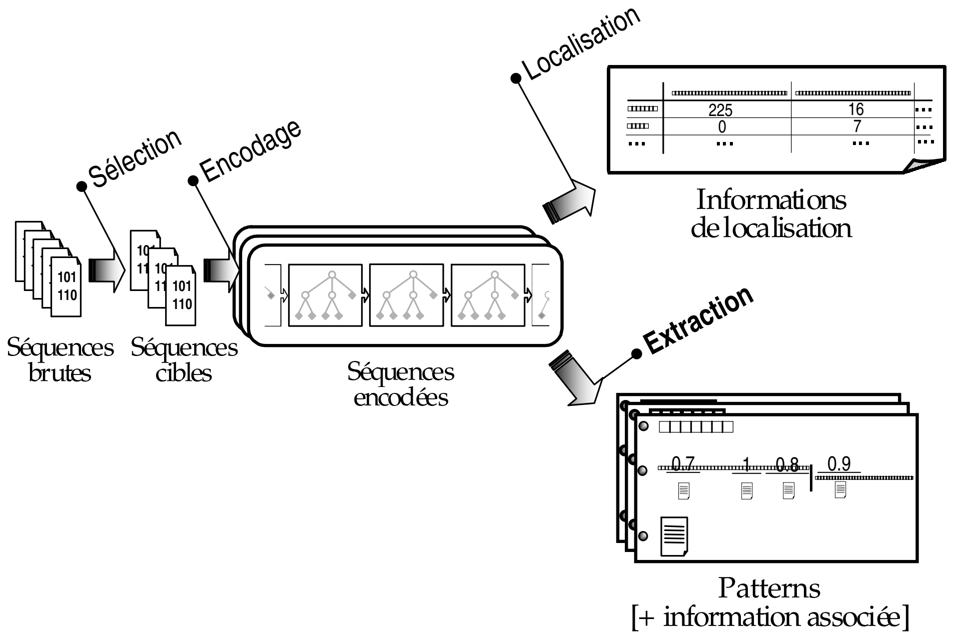


Figure 1-4. Schéma général de la découverte de régularités dans les séquences

Différentes approches de nature combinatoire ont été proposées précédemment dans la littérature, tant pour la localisation (3.3) que pour l'extraction (3.4) de patterns dans une ou plusieurs séquences. Un certain nombre d'entre eux ont une visée générale, indépendante de tout domaine particulier d'application, et de nombreux autres ont été conçus spécifiquement dans un contexte biologique. Quelques-uns ont par ailleurs été proposés pour traiter des séquences musicales (par exemple [Stammen & Pennycook 1993 ; Cope 1991a]).

La composante centrale de tous ces algorithmes est un *modèle de comparaison* entre paires d'entités séquentielles (cf. 3.2), explicite ou implicite suivant les cas. Différents modèles de comparaison ont été proposés, parmi lesquels nous distinguons :

- Les modèles pour lesquels la structure de correspondance entre les éléments respectifs des deux entités comparées est *fixe* (cf. section 3.2.4.1) ; des instances classiques en sont la distance euclidienne ou la distance de Hamming, laquelle a été employée dans l'exemple plus haut.

- Les modèles pour lesquels cette structure est *variable*, dont l'exemple central est le modèle d'édition (distance d'édition et variantes). Un modèle de similitude (resp. de dissimilitude) s'inscrivant dans le cadre du modèle d'édition valué (cf. 3.2.5.2) est caractérisé par l'ensemble des *types d'appariements* (resp. *types d'éditions*) permis et les *fonctions de contribution* (resp. *fonctions de coûts*) associées à chacun d'entre eux.

Le modèle de comparaison employé par les algorithmes de localisation ou de comparaison a une influence majeure à la fois sur la nature des résultats qu'il permettent d'obtenir et sur leur complexité algorithmique.

1.1.2. Représentation

Il y a presque toujours plusieurs façons, alternatives ou complémentaires, de décrire les collections d'entités dans lesquelles on cherche à découvrir des régularités. Pour donner une illustration simple, considérons la séquence X, de longueur L, contenant les valeurs successives d'une variable numérique, par exemple un historique d'indices boursiers. La représentation initiale n'est pas nécessairement la meilleure en vue de la découverte de régularités. En effet, il peut être plus pertinent de considérer par exemple la séquence des variations de ces valeurs entre chaque élément et le suivant. Cette variation étant exprimée sous forme d'une différence, on remplacera le *descripteur* (valeur de la description) initial de chaque[2] élément, noté X[i].D, par le descripteur X[i]. Δ — ou on adjoindra ce dernier à la représentation initiale — avec :

$$\forall i \in \mathbf{N}_{L-1},\ X[i].\Delta = X[i+1].D - X[i].D$$

Par exemple, remplacer, sur chaque élément de séquence, une description numérique (voire l'élément lui-même s'il s'agit d'une séquence numérique) par la valeur de sa **variation** entre chaque élément et son successeur, permet d'obtenir une invariance à toute *translation verticale*. (Une telle translation est caractérisée par l'ajout, pour chaque élément de la séquence, d'une même constante τ à la valeur de la description numérique considérée). Notons qu'en musique, cette redescription variationelle, courante, en *intervalles* permet de rendre compte de la notion de *contour mélodique* (succession des intervalles et/ou directions d'intervalles) dont il a été montré qu'elle joue notamment un rôle important dans la mémorisation de la mélodie. Notons par ailleurs que la redescription variationnelle présentée peut se généraliser de différentes façons, par exemple en remplaçant la différence par un quotient (*ratios*), ou en prenant en compte uniquement le signe de la différence (*directions de variation*) ou son ampleur qualitative ('grande variation', 'petite variation' ou 'pas de variation').

De même, il peut être pertinent de tenir compte, dans le cadre de la découverte de régularités, de l'existence de formes caractéristiques dans la séquence comme des *contours monotones*, stricts ou approximativement (voir Figure 1-5). Ce type de

---

[2] Bien sûr, cela pose un problème pour le premier élément. Des solutions telles que nous en proposons au 5.5.2.2.3 (page 252) peuvent être apportées.

caractéristiques locales peuvent alors être prises en compte dans le calcul de similitude entre deux segments : le fait que deux segments contiennent (ou soient contenus dans) un contour monotone de même direction, en particulier, contribue alors positivement à la similitude calculée entre eux.

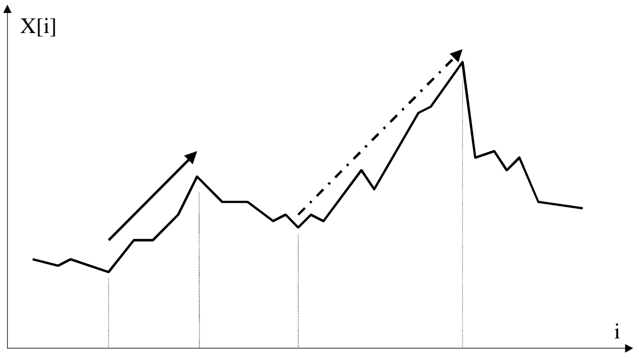


Figure 1-5. Contours ascendants dans une séquence numérique : strict (flèche continue) ou approximatif (flèche pointillée)

En matière de découverte de régularités et, plus généralement, de comparaison d'entités, la fixité et/ou la pauvreté des représentations peut avoir une influence significative sur la qualité des résultats obtenus. (Pour la biologie, voir par exemple [Sagot 1996]). Cette constatation se généralise d'ailleurs à d'autres domaines, par exemple celui de l'apprentissage automatique, où le choix de représentations adéquates pour les exemples est critique [Flann & Dietterich 1986 ; Widmer 1994]. Un rapprochement peut être fait avec une problématique plus générale incluant l'enrichissement des données, la *saturation* (à partir de connaissances du domaine — *background knowledge*), la *reformulation* ou encore le changement de représentation [Zucker 1996]. Le *principe de la connaissance* [Lenat & Feigenbaum 1991] postule pour sa part que plus on fournit de connaissances à une machine, plus elle est susceptible d'exhiber un comportement « intelligent » ; quoiqu'il demeure un postulat, ce principe s'est avéré valide dans un certain nombre de travaux applicatifs en intelligence artificielle.

Le problème de la représentation des séquences et de leurs éléments se pose en particulier lorsque l'on modélise (ou que l'on simule) informatiquement un processus de découverte de régularités ordinairement accompli par un « expert » humain, avec pour objectif d'obtenir des résultats les plus conformes possibles à ceux de l'expert. C'est le cas notamment pour les systèmes automatisant l'analyse musicale « orientée pattern » (voir 1.2.2. et 1.2.3 ci-après). Dans un tel contexte, la conformité du comportement du système quant à un certain nombre de composants du processus de découverte est critique, à commencer par celui du jugement (calcul) de similitude qui comme nous l'avons vu plus haut joue un rôle tout à fait central au sein de ce processus. Pour que le modèle de comparaison d'entités séquentielles intégré à l'algorithme combinatoire d'extraction ou de localisation produise des résultats conformes à ce qui est souhaité, il est important que ce modèle prenne le plus possible en compte les représentations que prend en compte l'expert dans son jugement de similitude. Par exemple, un spécialiste de l'analyse de courbes (séquences discrètes) de taux boursiers ne juge pas la similitude de deux segments de courbe uniquement sur la base de la séquence précise des valeurs qui les composent, mais prend aussi en considération, par exemple, la séquence des successions d'intervalles de valeur entre éléments successifs (cf. ci-dessus) ou encore de l'existence commune, au sein des deux segments, de contours monotones (Figure 1-5). Aussi, un système simulant — à des fins de prédiction par exemple — la détection de patterns récurrents dans ce telles courbes doit-il prendre ces deux types de descriptions supplémentaires dans son modèle de comparaison entre segments. Du moins cette nécessité constitue-t-elle une hypothèse de travail plausible.

Un autre lien doit être fait ici avec une notion générale appréhendée en psychologie : celle de point de vue ; typiquement, deux entités peuvent être semblables par rapport à un certain ensemble de caractéristiques (— traduisant un point de vue —) et différentes par rapport à d'autres (— traduisant un autre point de vue —) [Suppes et al. 1990 ; Rifqi 1996].

Aussi nous proposons d'insérer dans le processus de découverte de régularités une étape *d'enrichissement* des données qui suit celle de l'encodage et précède, suivant les cas, celle de l'extraction ou celle de la localisation de patterns. Le processus résultant est illustré sur la Figure 1-6. Précisons que, pour la localisation de patterns donnés au départ, l'étape d'enrichissement est également effectuée sur les patterns. L'automatisation de l'étape d'enrichissement suppose de concevoir des algorithmes spécialisés pour le calcul des *descriptions dérivées* à partir des descriptions de base, ce que nous avons fait pour le domaine musical. Indiquons que, dans le cadre d'un système d'ECD interactif, il est tout à fait envisageable que l'utilisateur complète ou modifie les descriptions ajoutées par le système durant la phase d'enrichissement, voire prenne celle-ci en charge complètement.

Deux remarques importantes doivent être faites ici. Nous choisissons délibérément, dans la présentation de nos travaux, de parler principalement d'**enrichissement** de la représentation, cela dit tout ce qui sera présenté est tout aussi valide dans le cadre d'un **changement** de représentation où tout ou partie de la représentation initiale est abandonné au profit d'une autre représentation. La seconde remarque est que, si l'enrichissement (ou le changement) de la représentation est schématiquement présenté comme **précédant chronologiquement** la localisation ou l'extraction de patterns, les choses peuvent avantageusement être différentes en pratique. Pour des raisons d'encombrement mémoire en particulier, rien n'empêche que l'enrichissement se fasse dynamiquement en fonction des besoins liés au déroulement de l'algorithme de localisation ou d'extraction, quitte à ce que les descriptions supplémentaires soient le cas échéant « oubliées » dès que les éléments concernés ont été entièrement traités par l'algorithme.

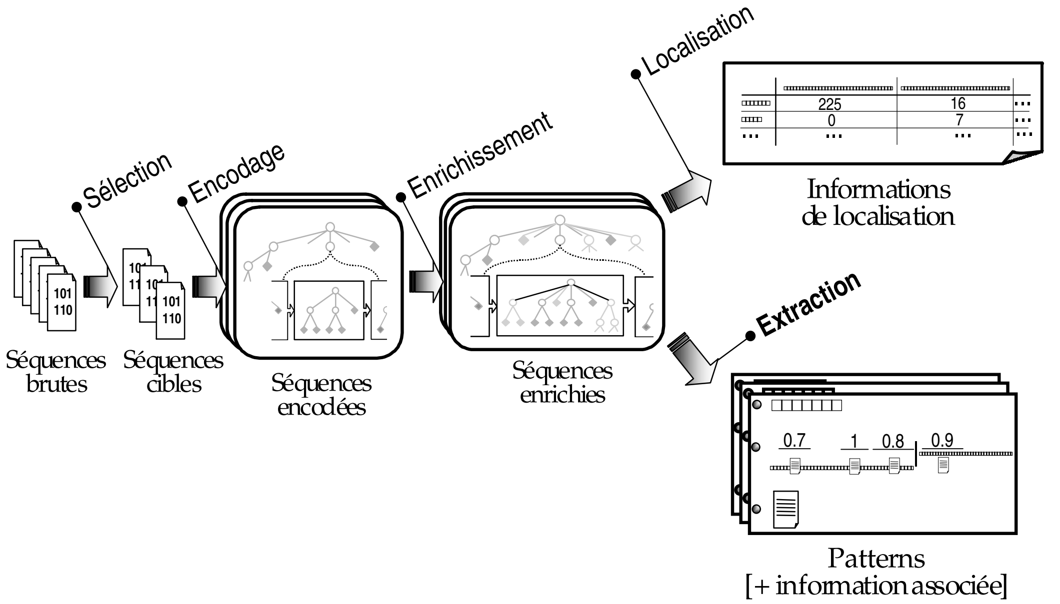


Figure 1-6. Vision enrichie du processus de découverte de régularités dans les séquences

Ainsi, l'insertion de nos travaux au sein du champ de l'intelligence artificielle recouvre deux aspects :

- Un premier aspect concerne la ***représentation de connaissances*** pour la découverte de régularités dans les séquences et, de manière plus générale, la comparaison d'entités séquentielles.
- Un second aspect, le cas échéant, concerne la **simulation** du **jugement de similitude humain**, avec tous les aspects relatifs à la modélisation cognitive que cela suppose. Du point de vue des objectifs fondamentaux de l'intelligence artificielle [Ganascia 1990; 1995] ce second aspect se situe dans une perspective plus directement liée à la modélisation de l'esprit humain et de son fonctionnement.

1.1.3. Modèle d'Édition Valué Multi-descriptions

Le *modèle d'édition valué multi-descriptions,* ou MEVM, que nous avons proposé et implémenté [Rolland & Ganascia 1998a ; 1996b] est un modèle général de comparaison entre entités séquentielles, sous-tendu par les considérations relatives à la représentation qui précèdent.

Le MEVM (section 4.7) reprend, en les généralisant, certaines caractéristiques des modèles d'édition proposés par Mongeau & Sankoff (1990) et par Stammen & Pennycook (1993) pour la comparaison de passages musicaux. Dans ces modèles (cf. 3.2.5.2.2), deux descriptions sont explicitement prises en compte simultanément dans le calcul de la contribution associée à chaque appariement. Le MEVM, pour sa part, fait intervenir un nombre arbitraire de descriptions simultanées. Ainsi, en musique, pour matérialiser le fait que l'omission d'une note perçue comme "importante" passe moins inaperçue que celle d'une note plus "neutre", nous pouvons rendre le coût de la suppression d'une note croissant en fonction :

- de sa durée — absolue ou relative (par rapport, par exemple, à celle des notes voisines)
- de son niveau métrique (cf. 4.7.7.1)
- de son degré de *saillance* (d'autant plus élevé que la note représente, par exemple, un maximum local de hauteur ou d'amplitude).

De même, le coût de substitution d'une note par une autre peut ne pas se limiter à la différence entre leurs hauteurs absolues, leurs durées et/ou leurs amplitudes respectives. Ce coût peut faire intervenir un ensemble éventuellement redondant de descriptions traduisant, sur les dimensions temporelle et/ou fréquentielle, les propriétés de chacune des notes et de leur voisinage. Le niveau d'abstraction, variable, des différentes descriptions confère à cet ensemble une structure partiellement hiérarchique.

Nous distinguons trois niveaux imbriqués de descriptions, en fonction de l'étendue horizontale des descriptions :

- Une description *individuelle* (ou atomique) concerne un élément et un seul. Les descriptions initiales des éléments des séquences en font typiquement partie, par exemple les valeurs X[i] pour le cas de la séquence numérique vue plus haut en exemple. Pour des séquences mélodiques (cf. section 1.2.3), les descriptions de base sont généralement la hauteur absolue, la durée et l'amplitude, exprimées sur des échelles discrètes[3]. Au-delà de celles-ci, une large variété de descriptions individuelles, dites *dérivées*, peuvent être prises en compte [Rolland & Ganascia 1998a]. Chaque note peut par exemple être également décrite par :

- sa position métrique,
- son degré (diatonique) dans la tonalité globale du morceau auquel elle appartient ou encore dans les tonalités associées à l'accord local, ou le cas échéant à une cellule harmonique (schéma d'accord type tel que II-V ou VI-II-V-I [Baudoin 1992a; b]).
- l'intervalle, diatonique ou chromatique, formé avec la note précédente, la note suivante ou encore la note située deux éléments avant (« précédente de la précédente »)

- Une description *locale* concerne un facteur (segment de séquence) donné. Les contours ascendants ou descendants de la Figure 1-5 entrent par exemple dans ce cadre. Nous y incluons aussi, en musique, des structures locales telles que schémas mélodico-rythmiques caractéristiques [Narmour 1989] ou encore, dans le cadre plus spécifique du jazz tonal, les cellules harmoniques (cf. plus haut). La structuration des séquences en groupes (*phrases* ; cf. par exemple [Lerdahl & Jackendoff 1983]), est également prise en compte au moyen de descriptions locales.

- Enfin, une description *globale* concerne une séquence dans sa totalité. Sur un plan général, on peut par exemple prendre en compte, dans l'extraction de patterns, le numéro d'ordre de chaque séquence dans le corpus. En tant que description globale, ce numéro d'ordre peut être employé comme biais favorisant l'extraction de patterns spécifiques (locaux) à une séquence particulière du corpus. Sur un plan musical, le tempo, l'armure et le mode (définissant la tonalité globale d'un morceau), ou encore la grille d'accord dans le cas du jazz tonal, sont des descriptions globales.

La prise en compte de propriétés locales et globales est ce qui permet de dépasser le paradigme du "note par note", qui — cf. [Widmer 1993 ; 1994 ; Rolland & Ganascia 1998a] — représente une limitation majeure des approches antérieures pour la comparaison d'entités séquentielles.

---

[3] Dans le cas, fréquent, où les séquences sont issues de fichiers MIDI [SMF 1988], ces échelles sont fixées par la norme MIDI [Moog 1986] — nous reviendrons plus loin sur ces aspects.

En plus des types d’appariements (éditions) intervenant dans le modèle d’édition standard — *insertion, délétion* et *remplacement*, le MEVM peut, comme nous le verrons, en faire intervenir un ensemble plus riche, par exemple : *consolidations*, *fragmentations*, *permutations* (*swaps*) ou encore *substitutions multiples.* A notre connaissance ces deux derniers types n’ont jamais été employés dans un contexte musical.

Pour tout type d'appariement, sa fonction de contribution associée est une combinaison pondérée de fonctions de contribution "spécialisées", chacune se rapportant rapport à une description (voir 4.7.4). Cette pondération permet une prise en compte effective de la notion de point de vue dont nous avons souligné plus haut l’importance. La possibilité, dans le MEVM, de choisir la pondération de chaque description ou groupe de descriptions matérialise l’adoption de points de vue différents.

1.1.4. FlExPat

L’algorithme[4] FlExPat (*FlExible Extraction of Patterns* — Chapitre 5) que nous avons conçu et implémenté [Rolland 1998] est un algorithme de type combinatoire permettant d’utiliser tout modèle d’édition valué à fonctions de coût non constantes tel que le MEVM. A notre connaissance, tous les autres algorithmes d’extraction de patterns à portée générale obligent à travailler avec des modèles plus pauvres.

Deux phases algorithmiques principales interviennent :

- La première, dite *phase d’appariement de facteurs*, permet d’identifier tous les couples de facteurs similaires, définis comme ceux dont la similitude numérique a été déterminée comme étant au-dessus d'un seuil fixé. Pour chaque couple, les 2 facteurs jugés similaires peuvent soit appartenir à la même séquence, soit à 2 séquences différentes. Cette phase produit un graphe dont chaque noeud représente un facteur et chaque arc représente une relation de similitude. Généralement, ce graphe, appelé *graphe de similitude*, est valué en vue de pouvoir différencier les arêtes suivant la force de la similitude correspondante.

- La seconde, dite *phase de catégorisation*, permet d'extraire les patterns à partir du graphe de similitude. Plusieurs approches sont possibles pour effectuer cette extraction. Les méthodes classiques de recherche de cliques (maximales) dans un graphe peuvent être appliquées ; nous en proposons d’autres, moins problématiques au plan algorithmique et donnant tout de même des résultats

---

[4] En toute rigueur, il s’agit en réalité d’une **famille** d’algorithmes, comme nous le verrons plus bas.

satisfaisants. En particulier, la méthode que nous nommons "Centre d'Etoile", s'inspire d'une méthode proposée dans le cadre de l'alignement multiple en biologie moléculaire [Gusfield 1993].

1.2. Application à l'analyse musicale

1.2.1. Informatique, intelligence artificielle et musique

L'informatique musicale (au sens large), située au croisement de plusieurs disciplines, est aujourd'hui considérée comme un domaine de recherche à part entière. Des conférences annuelles y sont consacrées tant au plan national (Journées d'Informatique Musicales [Pachet & Chemiller 1998]), qu'au plan international (par exemple, *International Computer Music Conference* [Pennycook & Simoni 1995]).

Plus spécifiquement, la musique s'affirme au fil des années comme un champ d'investigation privilégié pour l'intelligence artificielle[5], au même titre que les autres disciplines artistiques [Ramalho 1997a]. Là aussi, cela apparaît au travers de l'existence d'un certain nombre d'ouvrages (par exemple [Balaban, Ebcioglu & Laske 1992]) et de manifestations scientifiques consacrés aux liens entre les deux disciplines :

- *International Conference on Music and Artificial Intelligence* (1995)
- *Workshops* organisés dans le cadre de conférences majeures sur l'intelligence artificielle : *International Joint Conference on Artificial Intelligence* (IJCAI), *National Conference on Artificial Intelligence* (AAAI) et *European Conference on Artificial Intelligence* (ECAI). Les deux dernières éditions ont été organisées dans le cadre de l'IJCAI [Widmer 1995 ; Hirata 1997].

Les travaux présentés dans le présent mémoire s'inscrivent dans la lignée d'un ensemble de recherches en informatique musicale effectuées au sein du LAFORIA (ancienne dénomination du Pôle Intelligence Artificielle du LIP6). Ces recherches couvrent un large éventail de sujets, allant — pour les plus récents — de l'analyse musicale au sens large [Pachet, Ramalho & Carrive 1996 ; Pachet 1998] à l'aide à la synthèse sonore [Rolland & Pachet 1996 ; 1998], en passant par l'accompagnement improvisé automatique [Ramalho & Pachet 1994 ; Ramalho, Rolland & Ganascia 1998].

---

[5] Pour un panorama des travaux effectués en France dans le champ « intelligence artificielle et musique », voir [Pachet & Assayag 1995].

1.2.2. Découverte de régularités et analyse musicale

Lorsqu'on écoute de la musique, on remarque fréquemment des structures récurrentes ou régularités de différentes natures. Quoique cela ne soit pas vrai de tous les types de musique, cela est vrai pour une large variété de genres, notamment la musique tonale occidentale (baroque, classique, etc.) et le jazz, du moins de l'ère du « New Orleans » à celle du « Be-bop ». Au sein de la notion de régularité en musique, la *répétition* occupe une place privilégiée. L'illustre musicologue Heinrich Schenker dit de la répétition qu'elle constitue « la base de la musique en tant qu'art. Elle crée la forme musicale, tout comme l'association d'idées à partir d'un pattern dans la nature crée les autres formes d'art[6] » [Schenker 1954, pp. 4-5, cité par Stech]. Sur le plan des rapports existants entre répétition et syntaxe musicale, Ruwet (1972b) précise, lui, que celle-ci « repose sur des rapports d'équivalence—autrement dit, sur la répétition, au sens le plus général du terme. ». Notons que l'emploi de matériau récurrent est un procédé extrêmement répandu dans les différentes disciplines artistiques. Pressing (1988) et Kernfeld (1989b) citent notamment la poésie homérique/épique et les troubadours.

Comprendre et caractériser le style de compositeurs ou d'improvisateurs — ou les techniques mises en œuvres par ceux-ci — est la motivation centrale de l'analyse musicale « orientée pattern » (cf. 2.1.3.4). On cherche à mettre en évidence des structures récurrentes et la manière dont elles sont agencées au sein des œuvres [Rolland & Ganascia 1998a]. En matière de musique composée, on peut donner comme exemple d'une telle investigation basée sur les patterns, l'analyse faite par Ruwet (1972a) d'œuvres de Debussy. Ruwet a cherché à y détecter des phrases ou passages répétés avec des variations simples. Les improvisations de plusieurs grands musiciens de jazz ont fait l'objet d'analyses de type analogue [Baker 1980 ; Kernfeld 1983 ; Owens 1974 ; Smith 1983].

Un certain nombre d'outils informatiques ont été développés dans le but d'automatiser ou du moins d'assister l'analyse orientée-patterns (cf. 2.1.3.4.1). Il s'agit notamment d'épargner au spécialiste humain les tâches fastidieuses et répétitives (en raison de la combinatoire) que ce genre d'analyse requiert — surtout lorsqu'il est effectué de manière quantitative (cf. par exemple [Owens 1974]) :

- Sélectionner et comparer entre eux un nombre potentiellement immense de fragments musicaux

---

[6] "(…) the basis of music as art. It creates musical form, just as the association of ideas from a pattern in nature creates the other forms of art."

- Faire, tester et éventuellement réviser des hypothèses concernant les patterns candidats
- Compter et localiser les occurrences de patterns, etc.

Nous avons développé un système baptisé Imprology[7] [Rolland & Ganascia 1996a ; 1998a] capable d'effectuer automatiquement une analyse orientée-patterns sur un ensemble de séquences mélodiques au format standard Midi File [SMF 1988]. Imprology permet à la fois de localiser des patterns connus à l'avance et d'extraire de nouveaux patterns (algorithme FlExPat). Le modèle de comparaison intégré dans Imprology est notre modèle d'édition valué multi-descriptions présenté plus haut.

Nous avons effectué un certain nombre d'expérimentations sur un corpus qui est un (large) sous-ensemble de celui de l'étude d'Owens (1974) — 250 solos du saxophoniste de jazz *Be-bop* Charlie Parker transcrits par le musicologue. Le jazz improvisé, surtout dans le genre *Be-bop*, est un type de musique très intéressant pour la découverte de régularités pour tout un ensemble de raisons que nous expliciterons dans la suite (en particulier à la section 4.2). Par ailleurs, la richesse et le caractère quantitatif de l'étude d'Owens cette dernière fournit un support de validation de grande qualité pour Imprology. Soulignons à cet égard l'une des caractéristiques importantes de nos travaux, du point de vue de leur contribution au domaine de « l'analyse musicale assistée par ordinateur ». Contrairement à un grand nombre de travaux dans ce domaine, nous avons souhaité nous attaquer à des problèmes en « vraie grandeur » ; sur le corpus d'Owens, nous travaillons sur un corpus constitué d'environ 150 solos parmi les 250 transcrits par le musicologue (cf. 4.4.1), représentant un total de l'ordre de 10000 mesures ou 60000 notes. A ce titre notre recherche s'inscrit dans le courant de la « musicologie quantitative » soutenu par des organismes tels que le *Center for Computer Assisted Research in the Humanities* (*CCARH*). Des projets tels que *Humdrum* [Huron 1994] ou *the Music Library of the Future* [Pennycook 1995] s'y inscrivent également.

En termes d'applications musicales de nos travaux (concepts, modèles, algorithmes et implémentation), nous nous concentrerons dans ce mémoire sur l'analyse musicale orientée-patterns au sens large. Néanmoins d'autres applications musicales directes ou indirectes existent [Rolland & Ganascia 1998a], par exemple :

---

[7] Concernant l'origine du nom Imprology, voir la section 4.2.

- L’extraction automatique de matériau mélodique récurrent dans des œuvres existantes, en vue de sa réutilisation par un système informatique de composition, d’improvisation ou d’accompagnement (cf. 2.1.3.3).

- L’anticipation dans un système musical interactif [Rowe 1993 ; 1998], cf. 2.1.3.5

1.2.3. Représentation musicale

L’étude du jugement de similitude musicale humain, tel qu’il intervient dans des tâches d’analyse musicale « orientée-patterns », ressortit aux sciences de la perception et de la cognition musicales. Celles-ci constituent aujourd’hui un domaine de recherche très actif, avec en particulier des conférences internationales annuelles ou bi-annuelles telles que l’*International Conference on Music Perception and Cognition* (ICMPC), dont les localisations des deux dernières éditions sont Montréal [Pennycook & Costa-Giomi 1996] et Séoul [Won Yi 1998], ou encore des revues internationales (par exemple *Music Perception*). Qu’il s’agisse de résultats de psychologie expérimentale, ou de *théories* possédant une connotation cognitive plus ou moins forte (par exemple Lerdahl & Jackendoff 1983 ; Narmour 1989, 1992]), ces travaux mettent entre autres en évidence que le traitement (perception, mémorisation, création…) d’une mélodie ne se résume pas au traitement d’une séquence de hauteurs, durées et amplitudes (et le cas échéant de timbres) ; les « activités musicales » au sens large — qu’il s’agisse de perception, d’écoute, de mémorisation, de composition, d’improvisation ou encore d’interprétation — font intervenir un ensemble divers de structures cognitives et de multiples représentations**.** Celles-ci sont d’ailleurs, en grande partie, liées à la culture dont est issu l’auditeur (par exemple, la culture occidentale par opposition à la culture indienne au sens large).

Pour simuler le jugement de similitude intervenant dans les tâches liées à la localisation approximative de passages mélodiques ou à l’extraction de patterns (passages récurrents), une hypothèse de travail raisonnable — et que nous faisons — est de considérer que le plus grand nombre de ces descriptions doit être intégré au modèle de comparaison d’entités séquentielles du système [Rolland & Ganascia 1996b]. Cela ne signifie pas l’on doive toujours utiliser simultanément la totalité de ces descriptions, que l’on doive accorder la même importance à chacune d’entre elles, ou encore que les emphases relatives placées sur les différentes descriptions doivent demeurer identiques quels que soient les objectifs analytiques poursuivis. Nous traduisons cela par l’introduction des paramètres de pondération, ajustables à souhait par l’utilisateur, au sein du modèle d’édition valué multi-descriptions, comme introduit plus haut. Une typologie de descriptions musicales intégrées ou intégrables dans le MEVM sera proposée au 4.7.7.

En informatique musicale, un ensemble de recherches, s'appuyant sur les résultats issus des travaux qui viennent d'être mentionnés, soulignent la nécessité absolue de conférer un « sens commun » ou « bon sens » musical aux systèmes en introduisant le plus possible de connaissances musicales dans leur 'raisonnement' (entendu au sens large) [Widmer 1992 ; 1994 ; Horowitz 1995a ;b ; Pachet 1991a ; Ramalho & Pachet 1994]. Notamment, dans le cadre de la conception d'un système d'interprétation musicale, Widmer (1994) propose un changement de représentation (*transformation strategy*) présentant des analogies avec celui que nous avons illustré sur la Figure 1-6. Il est important de noter que, dans le cas du système de Widmer — mais aussi dans d'autres cas, par exemple celui du système ImPact [Ramalho & Ganascia 1994 ; Ramalho, Rolland & Ganascia 1998] — ce changement de représentation est *destructif* au sens où il se traduit par un « oubli » de la représentation initiale, c'est-à-dire celle au niveau des notes individuelles. Dans le cadre de l'analyse musicale telle que celle que nous visons à simuler, ce type d'oubli peut s'avérer pertinent suivant les objectifs visée. Néanmoins cette pertinence n'est pas systématique : on constate que manifestement tout ou partie des descriptions de base (initiales) sont susceptibles d'entrer en ligne de compte dans le processus d'extraction de patterns. En particulier, tant le niveau des notes individuelles que celui des groupes de notes et des structures plus abstraites ont un rôle à jouer dans le jugement de similitude. Une confirmation empirique de cette constatation nous a été fournie, tant par l'observation qualitative, dans le cadre de l'étude d'Owens (1974), de la forme des patterns du lexique (cf. 4.4.2) et de leurs occurrences, que les expérimentations que nous avons conduites et qui seront, pour partie, présentées dans ce mémoire.

Pour donner dès maintenant un aperçu des résultats expérimentaux, énonçons la conclusion majeure suivante qui en est issue : un **enrichissement** (ou un **changement**) **de la représentation** conduit, à condition d'être pertinent, à une **amélioration très significative des résultats** de la découverte de régularités dans les séquences. L'amélioration se situe tant sur un plan qualitatif que sur un plan quantitatif.

1.3. Plan de la thèse

Dans le premier chapitre qui suit cette introduction nous considérerons la problématique de la découverte de régularités dans les séquences, et plus particulièrement celle de la découverte de patterns séquentiels, sous deux angles différents : celui de l'application puis celui de l'informatique théorique. Au-delà d'une présentation des différentes problématiques, nous proposerons (section 2.2) une formalisation précise des différents concepts et problèmes relatifs à la comparaison d'entités séquentielles, la localisation de patterns (connus à l'avance) et l'extraction de (nouveaux) patterns.

Cette formalisation nous permettra ensuite de proposer une présentation unifiée de travaux utilisant des terminologies et des concepts différents de l'un à l'autre. Le Chapitre 3 est consacré à un passage en revue de l'état de l'art en matière de concepts et techniques pour la comparaison d'entités séquentielles (3.2), puis pour la localisation (3.3) et l'extraction (3.4) de patterns séquentiels.

Au chapitre suivant, nous présenterons un algorithme de localisation de patterns mélodiques et son implémentation dans le système Imprology que nous avons conçu en vue de la découverte de régularités dans les séquences musicales (localisation et extraction de patterns). A la lumière des résultats expérimentaux qui seront présentés, nous introduirons un nouveau modèle de comparaison entre entités séquentielles, le Modèle d'Edition Valué Multi-descriptions.

Après avoir mis en évidence les limitations , principalement dans le contexte applicatif qui est le nôtre, des algorithmes d'extraction présentés au 3.4, nous présenterons au Chapitre 5 l'algorithme FlExPat. L'implémentation au sein du système Imprology et un certain nombre de résultats expérimentaux seront également exposés.

Enfin, au Chapitre 6 nous présenterons les conclusions de cette thèse et dessinerons un certain nombre de perspectives.

Chapitre 2. PROBLÉMATIQUE

La problématique de la découverte de patterns dans les séquences peut être posée de différentes manières suivant le point de vue dont on se place (les termes spécifiques employés, par exemple "occurrence", seront définis ultérieurement, principalement en section 2.2) :

- Du point de vue de l'application considérée, il s'agit de définir précisément quel but est poursuivi au travers de la découverte de patterns, et d'indiquer les modalités de cette découverte :
    - Quelle est la forme des patterns recherchés ? Comment sont-ils représentés ? Sont-ils nécessairement des segments (*facteurs*) des séquences à l'étude, ou peuvent-ils aussi être des structures différentes, plus abstraites ?
    - Quelles contraintes impose-t-on sur les patterns pouvant être extraits, par exemple : (i) contrainte de quorum — le nombre minimal q de séquences différentes dans lequel le pattern doit avoir des occurrences ; (ii) "taille" minimale des patterns.
    - Quelles contraintes impose-t-on sur les relations pouvant exister entre les différents patterns extraits et/ou leurs occurrences (recoupements maximum autorisés, etc.) ?
    - Quelles autres informations que les patterns eux-mêmes cherche-t-on (par exemple, les caractéristiques de leurs ensembles d'occurrences) ?

- Quelle interprétation fait-on des patterns et des informations supplémentaires extraits ?

  Nous verrons à la section 2.1 les similitudes et les différences qui existent d'un domaine à l'autre dans les réponses à ces questions générales.

- Par ailleurs, vue sous l'angle de l'informatique théorique la découverte de patterns dans les séquences s'inscrit dans le cadre plus général de l'*algorithmique sur les séquences* ou encore *sur les textes* [Crochemore & Rytter 1994], comme nous le verrons à la section 2.2. Une remarque importante doit être faite ici : le découpage fait ici entre différentes sous-problématiques de la découverte de régularités dans les séquences, ne vaut que par la clarté d'exposition qu'il permet (du moins nous l'espérons !). En particulier, la frontière entre informatique théorique d'un côté, et intelligence artificielle, reconnaissance des formes ou ECD, de l'autre, est particulièrement floue dans le contexte de la découverte de régularités.

2.1. Vue sous l'angle de l'application

Dans un grand nombre et une grande variété de domaines, les données manipulées (ou du moins une partie d'entre elles) sont intrinsèquement séquentielles. Il existe sur ces données un ordre naturel, qui dans de nombreux cas est le temps (courbes de taux ou d'indices boursiers, enregistrements d'activité sismique, séquences d'actions utilisateur, pièces musicales, etc.). Parmi les autres cas figure celui de la biologie moléculaire, dont l'importance est à la mesure des enjeux scientifiques, médicaux et économiques colossaux liés à la recherche sur la génétique et en particulier sur le génome humain.

2.1.1. Séquences non-temporelles[8]

2.1.1.1. Biologie moléculaire

---

[8] Dans certains cas il est possible de donner une interprétation temporelle de l'ordre sur lequel sont basées ces séquences. Considérons par exemple en biologie les procédures de transcription ou de traduction d'acides nucléiques. Au sein de ces procédures intervient un parcours des macromolécules (ou d'une partie de ces macromolécules) qui s'inscrit dans le temps. Dans le contexte de ce parcours plus ou moins linéaire, l'ordre temporel se confond ainsi avec l'ordre naturel existant sur ces séquences. Néanmoins, cette perspective temporelle n'est valide que dans un tel contexte. Aussi jugeons-nous pertinent de distinguer clairement les séquences considérées dans cette section des séquences intrinsèquement temporelles qui font l'objet de la section suivante.

2.1.1.1.1. Nature des séquences

En biologie moléculaire, deux types de macromolécules (polymères) jouent un rôle primordial :

- Les acides nucléiques, acide désoxyribonucléique (ADN) et acide ribonucléique (ARN), chaînes moléculaires dont les maillons (monomères) sont appelés *nucléotides* ; il existe 4 nucléotides différents.
- Les protéines sont des chaînes dont les maillons sont des *acides aminés* — il en existe 20 différents.

Les protéines résultent de la traduction de l'ARN, lui-même issu de la transcription de l'ADN. Acides nucléiques et protéines peuvent être représentées de différentes manières. On distingue :

- la structure primaire, qui décrit la succession linéaire des monomères, traduite par une séquence (chaîne de caractères) sur un alphabet de 4 ou 20 symboles, respectivement. L'analyse de la structure primaire est donc de nature lexicale.
- la structure secondaire et la structure tertiaire, décrivant la morphologie de la macromolécule dans l'espace. (La structure tridimensionnelle d'une protéine joue un rôle déterminant dans la fonction que celle-ci exerce ou contrôle dans la cellule).

2.1.1.1.2. Objectif et modalités de la découverte

Dans un cadre d'analyse de ces macromolécules, un objectif majeur est d'identifier les parties (régions) des macromolécules qui jouent un rôle dans le fonctionnement ou la reproduction des organismes vivants [Sagot 1996]. Cette identification est rendue possible principalement par le fait que ces régions sont généralement les mieux *conservées* au fil de l'évolution moléculaire. La *comparaison multiple de séquences* en biologie moléculaire vise à découvrir des régularités, régions identiques ou similaires au sein d'un ensemble de séquences nucléiques ou protéiques. Cela a fait l'objet d'un nombre significatif de travaux, dont un nombre important porte sur l'*alignement multiple* (cf. par exemple [Chan, Wong & Chiu 1992]).

2.1.1.2. Autres

Parmi les autres types de séquences non-temporelles dans lesquelles intervient la découverte de régularités, on peut citer les textes (vus par exemple comme une séquence de mots — au sens usuel, non informatique, du terme…). A l'Université Pierre et Marie Curie, dans ce domaine une thèse a été soutenue en 1997 par M. Guerdoud, et d'autres travaux sont en cours au LIP6 (cf. par exemple [Zaragoza & Gallinari 1998]).

2.1.2. Séquences temporelles

2.1.2.1. Panorama général

Les types de séquences temporelles pour lesquels le besoin de découverte de régularités se fait sentir sont nombreux. On peut citer (cf. par exemple [Berndt & Clifford 1996]) :

- Les données économiques et boursières (historiques de taux, d'indices, de prix d'actions)
- Les mesures scientifiques telles que la séquences télémétriques (satellites) ou sismiques
- Les données issues d'instruments de surveillance médicaux ou industriels

En dehors de ceux qui viennent d'être cités, nous considérons ici trois autres domaines d'application de manière plus détaillée. Les deux premiers, la conception d'agents d'interfaces adaptatifs et l'algorithmique des jeux, font l'objet de travaux au sein de l'équipe Acquisition de Connaissances et Apprentissage Symbolique Automatique du LIP6 à laquelle nous appartenons. Le troisième est la musique, qui constitue le domaine d'application privilégié de nos travaux et fera l'objet de la section 2.1.3.

2.1.2.2. Agents d'interface adaptatifs

Dans la conception d'un agent d'interface "intelligent" [Akoulchina & Ganascia 1997 ; Edwards et al. 1996 ; Maglio & Barett 1997], par exemple un agent d'aide à la navigation sur le Web ou dans un système hypermédia, il est souvent indispensable que l'agent soit *adaptatif*. Dans ce but, l'agent enregistre les séquences d'actions de l'utilisateur (par exemple, les clics de souris avec leur localisation précise ou les choix dans les menus). Il peut alors y extraire, de préférence en temps-réel, des régularités de différentes natures (par exemple des successions d'actions récurrentes) afin de prédire les prochaines actions de l'utilisateur et agir en conséquence, par exemple :

- Proposer à l'utilisateur d'effectuer les prochaines actions à sa place (voire prendre l'initiative de les effectuer sans lui demander son avis!…)

- Suggérer au contraire à l'utilisateur qu'il fait probablement fausse route, et lui proposer une séquence d'actions alternatives permettant d'atteindre le but que l'agent lui prête.

2.1.2.3. Jeux

Dans la conception d'algorithmes de jeux, l'une des approches possibles consiste à concevoir des algorithmes adaptatifs dont la stratégie de jeu évolue en fonction de l'historique des coups joués par les deux adversaires.

Par exemple, pour le jeu "Pierre Ciseaux Papier", Minasi (1991) a proposé un algorithme ad hoc (et très redondant au plan combinatoire) qui à chaque coup base son choix sur l'examen de l'historique des coups joués précédemment par lui-même et par l'adversaire. En simplifiant, cet examen consiste à trouver dans l'historique une successions de coups identique à celle qui vient d'être jouée. L'algorithme suppose alors que l'adversaire jouera au prochain coup de la même manière qu'il avait joué dans le passé (par exemple :"Ciseaux"), et fait son choix pour le prochain coup en fonction de cette prédiction (en l'occurrence : "Pierre", qui l'emporte sur "Ciseaux").

Meyer, Ganascia & Zucker (1997) mettent en oeuvre un algorithme plus élaboré, pouvant faire appel (suivant les paramètres choisis) à un algorithme génétique, et basé sur une modélisation de l'adversaire qui évolue au fil de la ou des partie(s). Ici aussi, les règles de jeu issues de la modélisation de l'adversaire et servant à l'algorithme pour faire son choix à chaque coup, sont basées (indirectement) sur les régularités existant dans l'historique des coups.

2.1.3. Séquences musicales

La problématique générale de la découverte de patterns en musique a été introduite au 1.2.2. Après avoir présenté un cas de découverte de patterns non séquentiels, nous nous concentrerons — pour le reste de ce mémoire — sur la découverte de pattern séquentiels.

2.1.3.1. Découverte de patterns non séquentiels

Dans certains travaux sur des séquences musicales les régularités recherchées ne sont pas, elles-mêmes, de nature séquentielle. Ainsi, dans le domaine musical, Widmer [1993 ; 1994] a dévéloppé un système qui extrait, dans des séquences mélodiques des régularités qui prennent la forme de règles d'expression. Plus précisément, les séquences source sont des interprétations de morceaux de piano (classique et jazz notamment) jouées par l'auteur sur un clavier MIDI, et les règles concernent les variations expressives de dynamique et de tempo relatif (rubato) appliquées par le pianiste. Ensuite, le système applique les règles qu'il a apprises pour appliquer des variations expressives : règles de crescendo, règles de diminuendo, règles d'accelerando, règles de ritardando, et les résultats obtenus sont extrêmement probants. Les règles ont un champ d'action d'une note. Autrement dit, si on considère par exemple une règle de crescendo, le déclenchement d'une instance de cette règle résultera en la modification de la dynamique d'une note bien précise. La prémisse d'une règle consiste en la description plus ou moins abstraite d'une *situation* dans laquelle l'expression correspondant à la règle peut être appliquée (par exemple un certain niveau de crescendo). La description d'une situation est faite en termes des composants du *modèle qualitatif de connaissance musicale* intégré au système. Celui-ci intègre un réseau hiérarchique de concepts issus notamment de travaux en théorie et psychologie de la musique au sens large. Parmi ces concepts, citons par exemple les structures métrique et de groupement [Lerdahl & Jackendoff 1983] ou les processus [Narmour 1989 ; 1992].

Ainsi, dans le cadre des travaux de Widmer, les régularités recherchées sont, schématiquement, des associations Description de Situation → Niveau d'expression.

2.1.3.2. Découverte de patterns séquentiels

An sens le plus large, un pattern (séquentiel) est une structure récurrente pouvant porter sur n’importe quelle combinaison de dimensions musicales à n’importe quel niveau d’abstraction. Typiquement, ces dimensions musicales sont la hauteur des notes, leur support temporel (instant de début et durée), le groupement vertical (accords) et le déroulement harmonique, les contours et le mouvement, la tension, etc. Par exemple, les patterns monodiques, comme les motifs en musique baroque ou classique, correspondent à des séquences de hauteurs et de durées de notes. De même, les patterns harmoniques, comme les *cellules* II-V ou VI-II-V-I en jazz tonal [Baudoin 1992a; b] correspondent à des séquences de fonctions d’accords. Dans la suite de ce mémoire, sauf mention particulière les patterns musicaux que nous considérerons sont des patterns *monodiques*. Cela dit la majeure partie des résultats théoriques et expérimentaux présentés sont transposables à d’autres types de patterns musicaux, en particulier polyphoniques ou harmoniques.

Une grande diversité de termes sont employés pour désigner des structures mélodiques récurrentes, dépendant du genre musical (tonal occidental, jazz, contemporain etc.) : pattern, motif, figure, idée, cliché, formule. Viennent s’y adjoindre des termes ou expressions anglo-saxons difficilement traduisibles, comme « hot lick » ou « gesture ». Ces différents signifiants sont souvent utilisés de manière interchangeable, néanmoins des distinctions ont été proposées[9]. Par exemple, Kernfeld (1989) distingue les *motifs* des *formules* en jazz (tonal) : dans l'improvisation de type « motivic », un morceau entier – ou du moins une section entière – est construit(e) à partir de quelques motifs (c'est le cas d'une certain nombre de solos du pianiste Benny Carter ou du saxophoniste John Coltrane). En revanche, dans l'improvisation de type « formulaic » (que Charlie Parker illustre superbement), les patterns sont des formules plus diverses, plus sporadiques, combinées pour former des lignes continues.

Dans les sections qui suivent, nous examinons les différentes situations musicales dans lesquelles intervient la découverte de patterns, qu'il s'agisse d'un contexte humain ou informatique :

- Création musicale au sens large (composition, improvisation, accompagnement)
- Analyse musicale, là aussi dans une acception large du terme [Pachet 1998]
- Autres (par exemple, l'anticipation dans un système musical interactif)

2.1.3.3. Création musicale

L'utilisation récurrente de structures mélodiques et rythmiques particulières au sein d'œuvres ou ensembles d'œuvres a fait l'objet, en tant que procédé compositionnel, d'une littérature abondante : de nombreux livres et traités abordent les notions fondamentales de répétition, de développement et de variation. Pour la musique improvisée et en particulier le jazz, le rôle joué par les patterns a été considéré à la fois dans le cadre de travaux de psychologie cognitive (cf. [Pressing 1988], et plus généralement [Sloboda 1988]) et dans celui de la pédagogie [Coker 1964; 1970]. De nombreux musiciens de jazz créent ou « empruntent », tout au long de leur carrière, des patterns (mélodiques, rythmiques et même harmoniques) en vue de leur réutilisation dans de nouvelles improvisations.

2.1.3.3.1. Systèmes et programmes

Conséquence naturelle de ce qui vient d'être dit, une proportion significative de systèmes de composition ou d'improvisation font intervenir des patterns [Ramalho 1997b]. Dans certains cas le rôle joué par ceux-ci est tout à fait central, un ensemble de patterns mélodiques et/ou rythmiques étant enchaînés ou combinés pour constituer la

---

[9] Des distinctions du même genre sont faites en biologie. Cf. par exemple la distinction établie entre motif et pattern par Kruskal & Sankoff (1983 – p. 78).

totalité de la production du système. Dans d’autres, l’utilisation des patterns est plus partielle, ceux-ci étant essentiellement employés comme ferment ou comme complément au sein d’un processus génératif complexe.

Il existe de nombreux systèmes de ce type, produisant de la musique dans différents genres :

- Musique tonale occidentale : système EMI, pour Experiments in Musical Intelligence [Cope 1991a]
- Jazz : Neuerswing [Baggi 1989], ImPact [Ramalho & Ganascia 1994 ; Ramalho, Rolland & Ganascia 1998], et les systèmes d’Hodgson (1996), Pennycook, Stammen & Reynolds (1993), Spector & Alpern (1995), et Ulrich (1977)
- Multi-genre : Cypher [Rowe 1993], Flavors Band [Fry 1991], et les logiciels du commerce comme Band-In-A-Box, The Jammer, Style Trax, SuperJAM Jr., etc.

Dans la quasi-totalité des systèmes existants, les patterns doivent être introduits initialement, ajoutés ou modifiés « manuellement », et cela est effectué :

- par le concepteur/programmeur uniquement (c’est le cas du système d’Hodgson ou d’ImPact, dans la forme actuelle), ou :
- par l’utilisateur également (cas de Band-In-A-Box ou de The Jammer)

Dans les deux cas, cela est difficile et fastidieux si la *base de patterns* est large. Par exemple, plus de 350 fragments ont été extraits de transcriptions de lignes de basse jouées par des contrebassistes (humains) et entrés dans ImPact. Automatiser l’extraction de patterns permet d’éviter cette phase manuelle, ou du moins de l'amoindrir.

Une autre application importante de l’extraction de patterns se situe dans le domaine de l’interaction entre musiciens humains et informatiques. Une manière efficace de faire en sorte que les musiciens informatiques construisent réellement sur le jeu des humains est en effet de les doter de capacités d’extraction et de réutilisation de patterns en temps-réel [Rowe 1993] : chaque système extrait des structures récurrentes dans le discours musical des autres musiciens, et les réutilise (après adaptation si nécessaire) pour produire son propre discours. Il y a eu quelques (rares) tentatives explicites d’automatiser l’extraction de patterns. Il s'agit, parmi les systèmes mentionnés ci-dessus, de celui de Pennycook, Stammen & Reynolds (1993), d'EMI [Cope 1991a] et de Cypher [Rowe 1993] (cf. section 3.4.1.8).

2.1.3.4. Analyse musicale

L’analyse musicale basée sur la recherche de structures récurrentes (typiquement, de nature mélodique) a été introduite au 1.2.2. Nous détaillons ici les caractéristiques principales de ce type d’analyse au travers de l’analyse faite par Owens [1974] du jeu improvisé de Charlie Parker (1920-1954), l’un des fondateurs du mouvement Be-bop (ou « Bop ») au milieu du siècle.

- Les patterns ou formules sont extraits à partir d’un ensemble de matériau musical (typiquement, des compositions ou improvisations d’un artiste ou d’un style particulier). Owens a extrait, à partir d’un corpus de 250 solos de Charlie Parker, un lexique de quelque 200 formules qu’il appréhende comme les briques (‘building blocks’) utilisées par le saxophoniste dans la construction de ses solos. La Figure 2-1 montre quelques formules issues de ce lexique ; les identificateurs hiérarchiques (1Aau, 1Bb, etc.) sont basés sur ceux employés par Owens dans la présentation de son lexique (voir aussi Annexe A).

- Des informations statistiques et structurelles sont éventuellement produites à partir des patterns individuels ou des groupes de patterns. Owens, pour sa part, a étudié les caractéristiques d’usage des différentes formules (fréquence, emplacement dans les solos), en les détaillant suivant le solo ou le groupe de solo considérés. Ce groupement est effectué :

    - soit sur la base de la tonalité des morceaux (si bémol majeur, fa majeur, do majeur, sol majeur et mi bémol majeur étant par exemple les tonalités les plus courantes des morceaux joués par Parker)

    - soit sur la base de leur schéma harmonique (ou type de grille d’accord): Blues, « I Got Rhythm », etc.

- Des conclusions d’ordre plus général sont enfin tirées des informations des deux catégories précédentes. Ainsi l’étude d’Owens aboutit à une caractérisation du style de Parker et des techniques employées par lui, caractérisation dont la portée musicologique fait elle aussi l’objet d’une réflexion spécifique [Owens 1974, pp. 267-272].

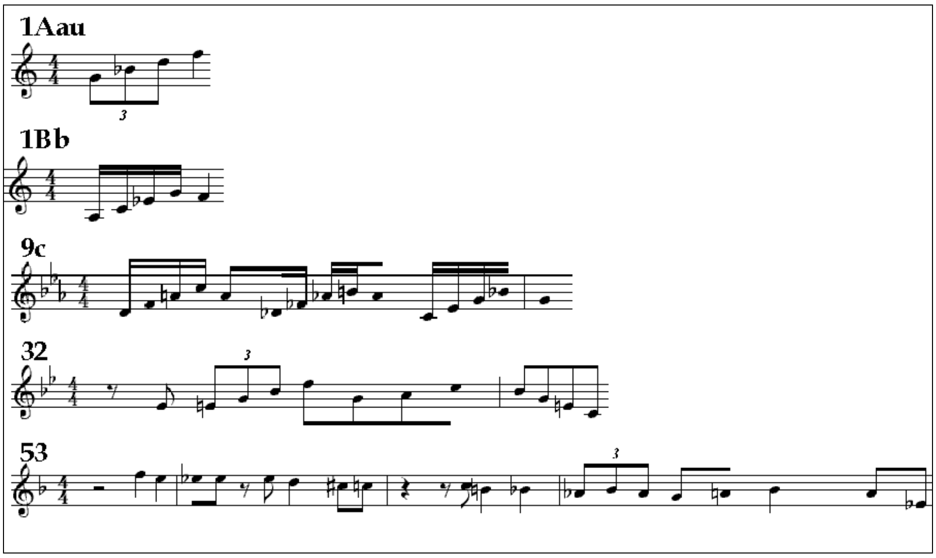



Figure 2-1. Quelques-uns des patterns (“formules”) trouvés par Owens (1974) au sein des solos de Charlie Parker.

2.1.3.4.1. Systèmes et programmes

Comme évoqué plus haut (1.2.2), le développement de systèmes informatiques pour l’analyse orientée-patterns vise typiquement à éviter à l’analyste humain les tâches fastidieuses et répétitives que ce genre d’analyse implique.

- Le composant analytique d’EMI identifie des patterns stylistiques appelés « signatures » au sein d’échantillons de compositions tonales données [Cope 1991a].

- Le système Imprology, outil général d’analyse orientée patterns sera décrit en détail au Chapitre 4. Il permet en particulier d’analyser sous un angle technique et stylistique de la musique improvisée, par exemple un corpus de transcriptions de solos de jazz [Rolland & Ganascia 1996a, 1996b]. Dans une configuration typique, le schéma analytique d’Imprology est inspiré de celui employé par Owens (cf. Figure 2-2) : l’étape principale (flèche continue) est l’extraction, à proprement parler, de patterns ou *formules* (A). Elle est éventuellement suivie (flèches pointillées) d’analyses statistiques (B1) et structurelles ou encore topologiques (B2).

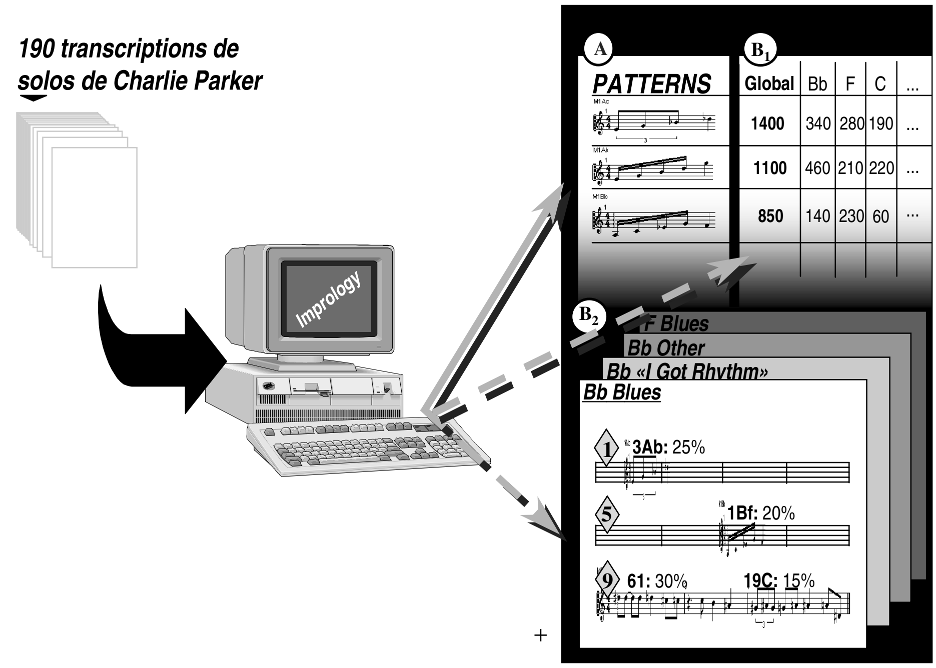


Figure 2-2. Dans sa configuration type, le schéma analytique d'Imprology est semblable à celui de l'étude d'Owens (1974).

- Le système de Stech (1981) peut être considéré comme effectuant un genre particulier d'analyse basée sur les patterns. Il recherche en effet des répétitions de segments, à un type de variation donné près, au sein de musique composée. 26 types de variation sont autorisés, basés notamment sur les concepts traditionnels de rétrograde (miroir horizontal) et d'inversion (miroir vertical) qui sont appliqués au rythme et/ou à la mélodie (cf. 3.2.4.1.2 et 3.4.1.8).

- De même, un type très spécifique d'analyse basée sur les patterns fait l'objet du travail de Smaill, Wiggins & Harris (1993) où l'objectif majeur est de valider un paradigme de représentation original dans un contexte analytique particulier (analyse à la manière de Ruwet).

- Enfin, certains outils d'analyse d'usage général comme Humdrum [Huron 1994] comportent quelques fonctionnalités de base en vue de l'analyse orientée-patterns.

2.1.3.5. Autres

La majorité des recherches touchant à l'usage – et, corrélativement, à l'extraction – de patterns dans les activités musicales des humains ou des ordinateurs se sont rapportés à la création musicale (composition, improvisation) ou à l'analyse. Cela dit, quelques autres aspects ont fait l'objet de réflexions ou de travaux. En particulier, il a été suggéré que les patterns jouent un rôle dans les processus cognitifs liés à *l'attente* (*expectation*) [Simon & Sumner 1968]. Les principes sous-tendant le mécanisme d’anticipation de Cypher se situent dans cette perspective : afin de prédire le jeu (au moins partiellement improvisé) d’un musicien, Cypher tente de détecter, en temps-réel, les débuts de patterns monodiques découverts précédemment. Dès lors que *la moitié* des notes d’un pattern ont été reconnues, Cypher anticipe que le musicien va poursuivre en jouant l’autre moitié du pattern, et ainsi le système peut entreprendre différentes actions sur la base de cette hypothèse. Remarquons que le principe est très proche de celui mis en œuvre par Minasi dans le domaine des jeux (cf. 2.1.2.3). La différence majeure tient à ce que le système de Minasi n'est pas soumis aux "vraies" contraintes de temps-réel auxquelles est soumis Cypher ; cela lui permet de recommencer à chaque coup joué la recherche de répétitions dans l'historique. Celle-ci est effectuée à partir de rien, c’est-à-dire sans réutiliser les résultats obtenus au coup précédent. Une telle réitération de la recherche à chaque coup (qui est d'ailleurs extrêmement inefficace au plan algorithmique) serait totalement exclue dans le cas de Cypher, sauf si l'on imposait que le débit de notes soit très faible…

2.1.3.6. Remarque

Par souci de clarté, nous avons fait une distinction nette, dans ce qui précède, entre différentes situations musicales. Cependant du point de vue de la découverte de patterns ces différentes situations sont fortement reliées ; en effet l’extraction de patterns à des fins de réutilisation ou d’anticipation dans les systèmes compositeurs ou improvisateurs peut être vue comme une forme particulière d’analyse – analyse en temps-réel ou différé suivant les cas.

2.2. Vue sous l’angle de l’informatique théorique

Au sein de l’informatique théorique (*theoretical computer science*), un sous-domaine spécialisé concerne l’algorithmique sur les séquences. Suivant le point de vue et le contexte, on parle aussi d'algorithmique sur les chaînes (néologisme anglais "stringology" [Galil 1994]), sur les mots ou sur les textes [Crochemore & Rytter 1994], ou encore de "combinatorial pattern matching" (une conférence annuelle porte ce nom).

Cette section, consacrée à donner une vision de la découverte de régularités dans les séquences dans le cadre de l'algorithmique sur les séquences, est aussi l'occasion pour nous de fixer un certain nombre de concepts, de termes et de notations qui seront utilisées dans la suite. Soulignons que, si nous nous attachons à indiquer les termes standard désignant les différents concepts lorsqu’ils existent, nous sommes également amené, quand cela est nécessaire, à proposer nos propres termes tels que ceux, par exemple, de *corpus* ou encore *séquence source* (2.2.3.1).

2.2.1. Définitions et notations

2.2.1.1. Séquences et éléments

Une *séquence* est une collection ordonnée d'entités appelés *éléments* de la séquence.

La *longueur* d'une séquence X, c'est-à-dire son nombre d'éléments, est notée $|X|$.

La séquence vide (de longueur 0) est notée λ.

Le i-ème élément d'une séquence X est noté X[i]. On appelle i le *rang* (ou *l'indice*, ou encore la *position*), de cet élément.

La formalisation classique d'une séquence en termes d'alphabets a été vue dans le chapitre d'introduction. Etant donnée une séquence X de longueur L sur un alphabet Σ, nous appelons *alphabet effectif* [relativement à X et Σ] l'ensemble des éléments (symboles) distincts contenus dans X. C'est donc le plus petit sous-ensemble de Σ tel que :

$$\{\forall\alpha\in\Sigma\}\ \{\exists i\in\mathbf{N}_L\}\{X[i]=\alpha\}$$

Plus X contient d'éléments distincts, plus l'alphabet effectif est grand, ce qui, comme nous le verrons, peut avoir des conséquences majeures sur la complexité en temps et/ou en espace des algorithmes de localisation et d'extraction de patterns. (Voir à ce sujet les discussions proposées au 5.3.8.3.4 et au 5.6.4)

2.2.1.2. Facteurs

Etant données deux séquences X et F, on dit que F est un *facteur* de X si et seulement si il existe deux séquences Y et Z, éventuellement vides, telles que :

$$X=YFZ$$

Equation 2-1

Y et Z n'étant pas nécessairement uniques.

Exemple

Sur Σ={a,b}, F=aab est un facteur de X=babaabb †

Remarques :

- On emploie parfois le terme alternatif de *segment.* C'est le cas par exemple pour des séquences musicales, où l'on parle aussi de *passages* ou encore de *fragments*.

2.2.1.2.1. Concaténation

Σ* est muni de l'opération classique de concaténation :

$$(X,Y) \mapsto X\,.\,Y \text{ (ou en raccourci XY),}$$

dont la séquence vide est l'élément neutre :

$$\{\forall X\in \Sigma^*\}\ \{X.\lambda = \lambda.X = X\}$$

(***Remarque*** : Le mot "facteur" se réfère bien sûr à la structure multiplicative conférée à Σ* par l'opération de concaténation)

2.2.1.2.2. Identification

Un facteur F d'une séquence X peut être identifié par le couple d'entiers (i, m) formé du rang du premier élément de F dans la séquence S (resp. dans la séquence quelconque X) et de sa longueur. Ce couple d'entiers forme les *coordonnées* de F dans X. Le *rang* d'un facteur (ou encore sa *position*) désignera alors naturellement celui de son premier élément. Ainsi le facteur F de rang i et de longueur m de la séquence X, désigné par X[[i,m]] est la séquence :

$$X[i]X[i+1]\ldots X[i+m-1].$$

Lorsque cela sera plus pratique, nous écrirons alternativement (notation double-pointée de type langage Pascal) :

$$X[i\ ..\ i+m-1]$$

Exemple

Sur Σ={a,b}, le facteur de rang 4 et de longueur 3 de X=babaabb est :

X[[4,3]]=X[4 .. 6]=aab †

2.2.1.2.3. Rang terminal

Nous appelons *rang terminal* d'un facteur le rang de son dernier élément. Le rang terminal du facteur de coordonnées (i,m) est i+m-1.

2.2.1.2.4. Préfixe et suffixe

Dans l'Equation 2-1, lorsque le facteur Y est vide (i.e. lorsque X=FZ) nous disons que F est un *préfixe* de X. De même, lorsque Z est vide (X=YF) nous disons que F est un *suffixe* de X. Lorsque F est un préfixe (resp. un suffixe) de X et que F≠X, F est qualifié de *préfixe strict* (resp. de *suffixe strict*).

Exemple

Sur Σ={a,b}, bab et b sont des préfixes de X=babaabb, et bb en est un suffixe. †

2.2.1.2.5. Bordure

La *bordure* ("*border*") de X est définie comme le plus long facteur de X qui en soit à la fois un préfixe et un suffixe.

Exemple

La bordure de X=baabbabbaab est baab.

N.B.: baab n'est pas le seul préfixe qui soit à la fois préfixe et suffixe de X : b l'est aussi †

Remarque : c'est la notion duale de celle de *période* d'une séquence, qui joue un rôle prédominant dans bon nombre d'algorithmes de localisation de patterns, du moins pour ce qui est de la localisation d'occurrences exactes (cf. 2.2.2 et 3.3.1)

2.2.1.2.6. Enumération

Nous notons facts(X) l'ensemble des facteurs *non vides* de la séquence X.

2.2.2. Localisation de patterns donnés

La localisation des occurrences d'un ou plusieurs patterns connus à l'avance est une variante classique de l'un des problèmes les plus étudiés au sein de l'algorithmique sur les séquences. Celui-ci, appelé génériquement *pattern matching* en anglais, peut se définir sous la forme d'un test que nous appelons *test d'occurrence* :

Etant donné un pattern P et une séquence X, renvoyer VRAI s'il existe, une (ou plusieurs) occurrence(s) de P dans X, FAUX sinon.

Remarque terminologique : lorsque les patterns sont eux-mêmes des séquences (typiquement, sur le même alphabet que les séquences dans lesquelles on recherche les occurrences de patterns), on parle plus spécifiquement de *string matching.*

La définition générale que nous nous donnons pour la localisation, ou *repérage*, d'un pattern dans une séquence est la suivante :

Définition 2-1

Etant donné un pattern P et une séquence X, localiser *le pattern P (sous-entendu : dans X) consiste à rechercher, s'il y en a, toutes les occurrences de P dans X.*

Une occurrence de pattern étant par définition un facteur (cf. 2.2.2.1), pour caractériser chaque occurrence on emploiera généralement ses coordonnées dans X.

Suivant l'application visée l'appellation peut varier. On parle par exemple d'opération de *word spotting* en traitement de la parole [Kruskal 1983]. La généralisation de la Définition 2-1 à un ensemble de patterns P1, … Pk est triviale : la localisation consiste dans ce cas à localiser successivement chacun des patterns dans la séquence et cela de façon indépendante pour chaque pattern. De même, la généralisation à un corpus de séquences {S1,…,SN} est immédiate : le résultat de la localisation s'obtient par la réunion des ensembles d'occurrences localisées au sein de chaque séquence. En pratique, ici il peut être plus efficace de concaténer S1,…SN puis de rechercher des occurrences de patterns dans la séquence globale obtenue. Il faut pour cela mettre en œuvre un mécanisme permettant d'écarter les éventuelles occurrences se trouvant à cheval sur deux séquences successives.

Nous rappelons ci-après la définition du mot 'occurrence'. Nous le faisons en premier lieu pour des patterns de forme quelconque. Puis nous envisageons la situation, fréquente, où les patterns sont des séquences sur le même alphabet que les séquences sources.

2.2.2.1. Définition d'une occurrence

2.2.2.1.1. Patterns de forme quelconque

Définition 2-2

Une occurrence *d'un pattern P dans une séquence X est un facteur de X. On note $Occ_X(P)$ l'ensemble des occurrences de P dans X.*

Lorsqu'un pattern P possède une ou plusieurs occurrences dans une séquence X, on dit que P est *actif*, ou encore *présent*, dans X.

Quelle que soit la forme prise par les patterns, il est possible de définir de manière opérationnelle ses occurrences. Cette définition fait intervenir une fonction booléenne, que nous pouvons par exemple nommer $EstOcc_X$, dont la valeur indique si le premier argument, un facteur F de la séquence X, est une occurrence du second argument, un pattern P. On a donc l'égalité booléenne :

$$EstOcc_X(F,P) = (F \in Occ_X(P))$$

Exemple

Soit Σ° l'aphabet latin. On suppose que les patterns sont des n-uplets non-ordonnés de symboles (caractères) tous distincts. Soient P = {C1, …, Cn} un tel pattern, X une séquence sur Σ°, et F un facteur de X. On définit la fonction $EstOcc_X$ par la relation booléenne suivante :

$$EstOcc_X(F,P) = \{\text{F contient une fois et une seule chaque caractère présent dans P}\}$$

Supposons que P={p,a,t} et que X="etpapatattrapera". La définition de $EstOcc_X$ permet de déterminer que $Occ_X(P)$ est composé des facteurs suivants :

- X[[1,4]] = "etpa" ;
- X[[2,3]] = "tpa" ;
- X[[5,3]] = "pat" ;

- X[[10,4]] = "trap" ;
- X[[10,5]] = "trape" ;
- X[[10,6]] = "traper" ; †

La formalisation à l'aide d'une telle fonction est particulièrement naturelle lorsque les patterns sont des séquences sur le même alphabet que les séquences sources.

2.2.2.1.2. Patterns — séquences sur le même alphabet

Dans ce cas de figure usuel, les patterns sont alors appelés *mots* plutôt que séquences, afin :

- de les distinguer, dans l'appellation, des séquences source
- d'évoquer le fait que les patterns sont généralement bien plus courts que les séquences sources.

La notion d'occurrence se définit alors à partir de celle d'*équipollence* entre mots :

Définition 2-3

Soit Σ *un alphabet. Une* fonction d'équipollence entre mots — *en abrégé* fonction d'équipollence —*est une fonction booléenne sur* Σ*×Σ*. *Deux mots (séquences) X et X' sur* Σ *sont dits* équipollents *au regard d'une fonction d'équipollence donnée Eq si et seulement si Eq(X,X') = VRAI.*

L'interprétation est la suivante : dire que deux mots sont équipollents signifie que leur ressemblance est significative au regard d'un critère donné. Nous choisissons d'utiliser un terme, "équipollence", distinct du terme usuel d'« équivalence ». La raison en est qu'une relation d'équivalence doit posséder des propriétés, en particulier la transitivité, qui ne sont généralement pas toutes avérées dans les situations où nous nous plaçons.

Soulignons qu'« équipollence » (ainsi que les adjectifs associés) est un terme imparfait dans la mesure où, dans son interprétation en géométrie euclidienne, il traduit une relation d'équivalence (entre bipoints). Nous estimons néanmoins préférable d'avoir recours à un tel terme plutôt que d'employer un terme ambigu comme « ressemblance » voire « similitude » qui ne traduirait pas l'aspect « quantique », non-graduel, de la notion à exprimer.

Nous ferons l'hypothèse qu'une fonction d'équipollence est symétrique, autrement dit que :

$$\{\forall\ (X,X') \in \Sigma^*\times\Sigma^*\ \}\ \{Eq(X,X') \Rightarrow Eq(X',X)\}$$

Néanmoins, cela n'est pas strictement nécessaire ; notamment, moyennant des adaptations mineures, les algorithmes que nous avons conçus — cf. Chapitre 4 et Chapitre 5 — peuvent s'accomoder de fonctions d'équipollence asymétriques.

Lorsque les patterns sont des séquences sur le même alphabet que les séquences sources, la fonction EstOcc (asymétrique par essence) du 2.2.2.1.1 est remplacée par une fonction d'équipollence (symétrique). On peut alors donner la définition suivante d'une occurrence de pattern.

Définition 2-4

Etant donnés un alphabet Σ, un pattern P qui est un mot sur Σ; une séquence X sur Σ et une fonction Eq d'équipollence entre facteurs, une occurrence de P dans X est un facteur F de X vérifiant :

$$Eq(P,F)$$

Occurrences exactes, occurrences approximatives

Définition 2-5

Soient Σ un alphabet, P un pattern qui est un mot sur Σ; X une séquence sur Σ et F une occurrence de P dans X. On dit que F est une occurrence exacte *de P dans X si et seulement si* ***F=P****, l'égalité entre séquences étant définie classiquement de la manière suivante :*

$$\{X=Y\} \Leftrightarrow \{\{|X|=|Y|\}\ \text{et}\ \{\{\forall\ i \in \mathbf{N}_{|X|}\}\ X[i]=Y[i]\}\}$$

Dans le cas contraire on dit que F est une occurrence approximative *de P dans X.*

Equipollence définie comme l'égalité

Un cas particulier très important pour la fonction d'équipollence est celui où cette fonction est un simple *test d'égalité* sur Σ*, autrement dit :

$$\{\forall\,(X,Y) \in \Sigma^*\times\Sigma^*\}\ \{Eq(X,Y)\} =\{X=Y\}\}$$

Dans ce cas, bien sûr, toute occurrence est nécessairement exacte.

Fonction d'équipollence induite par une fonction de similitude

Une fonction d'équipollence peut être obtenue en faisant un seuillage sur la valeur d'une fonction de similitude numérique entre séquences (mots). Un nombre *Seuil* étant fixé, pour un couple de facteurs (F,F') la fonction résultante vaudra VRAI si *Simil*(F,F')>*Seuil*, FAUX sinon (cf. 3.2.3). Notons que, dès lors que *Simil* est symétrique, c'est-à-dire (∀ (F,F')) (Simil(F,F')=Simil(F',F)) Eq l'est aussi. Or, rappelons que la grande majorité des fonctions de similitude entre entités séquentielles sont symétriques, du moins celles que nous passons en revue au 3.2.

Dans la suite nous faisons — sauf précision contraire — l'hypothèse implicite qu'une fonction d'équipollence Eq a été choisie. Ainsi lorsque nous dirons que deux mots sont équipollents nous entendrons : "équipollents au regard de Eq".

2.2.2.2. Localisation de mots

Nous avons donnés plus haut une définition générale de la localisation de pattern(s) donné(s). Supposons à présent que les patterns sont des mots (séquences) sur le même alphabet que les séquences sources. On peut alors préciser la définition de la façon suivante :

Définition 2-6

Etant donné un pattern P; une séquence X et une fonction Eq d'équipollence entre facteurs, localiser P dans X consiste à rechercher, s'il en existe, les coordonnées de tous les facteurs F de X vérifiant :

$$Eq(P,F)$$

2.2.3. Extraction de patterns

2.2.3.1. Concepts de base, terminologie et notations

2.2.3.1.1. Corpus et séquences sources

Les données séquentielles sur lesquelles on travaille forment un *corpus*, désigné par *C*, composé d'une ou plusieurs *séquences sources*.

Suivant le nombre de séquences sources auxquelles on a affaire, il y a deux cas de figure possibles :

- Celui, général, où le corpus est composé de deux séquences ou plus. On parle alors de *corpus multiple*.
- Celui, cas particulier important, où le corpus ne comprend qu'une seule séquence. On parle alors de *corpus unitaire*. L'importance de ce cas particulier provient non seulement du fait que dans les domaines d'applications on est souvent amené à rechercher des régularités **internes** dans une seule séquence (cf. par exemple [Mongeau et Sankoff 1990]), mais aussi du fait que les algorithmes traitant les corpora multiples sont souvent des généralisations d'algorithmes ne traitant que des séquences individuelles (cf. par exemple [Landraud, Avril & Chretienne 1989]).

On désigne par N le nombre de séquences composant un corpus multiple. Celui-ci est alors noté en extension $\{S_1, ..., S_N\}$, l'ordre dans lequel sont rangées les différentes séquences étant fixé une fois pour toutes. On appelle *séquence globale*, et on désigne par S, la séquence résultant de la concaténation de S_1, S_2, … S_N, autrement dit :

$$S=S_1S_2\ldots S_N$$

Dans le cas d'un corpus unitaire, l'unique séquence source (qui se confond avec la séquence globale) est également désignée par S.

La longueur totale du corpus, c'est-à-dire la somme des longueurs de chaque séquence source, est notée L. Dans le cas d'un corpus unitaire, cette longueur totale se confond bien sûr avec celle de l'unique séquence source.

Le *rang initial* de la n-ième séquence source est le rang de son premier élément, noté RgInit(n). On a bien sûr, pour tout $n \in \mathbf{N}k$:

$$\mathrm{RgInit}(n) = |S_1|+\ldots+|S_{n-1}|+1$$

2.2.3.1.2. Désignation des facteurs

Etant donné un corpus C ou une séquence quelconque X, il s'agit de désigner de manière univoque un facteur quelconque F de [l'une des séquences de] C, ou de X.

Si l'on a affaire à un corpus unitaire, F est tout simplement identifié par ses coordonnées (i, m) dans l'unique séquence S. Si en revanche le corpus est multiple, il y a deux possibilités :

- soit F est identifié par ses coordonnées (i, m) dans la séquence globale ;
- soit F est identifié par le triplet d'entiers (u; i, m) où u désigne le numéro de la séquence au sein de C et (i,m) est le couple des coordonnées de F dans S_u.

Exemple

Sur $\Sigma=\{a,b\}$, supposons que :

$$C=\{S_1,S_2\},$$

avec :

$$S_1=aabaaabab$$
$$S_2=baabbba,$$

Le facteur bbb est désigné :

l soit par S[13,3] (13 étant son rang dans la séquence globale S=aabaaababbaabbba),

l soit par C[[2 ;4,3]] †

2.2.3.2. Définition de l'extraction de patterns

Plusieurs formalisations explicites ou implicites ont été proposées dans la littérature. Nous adoptons la suivante, qui nous semble permettre un compromis satisfaisant entre généralité et précision :

Etant donnés :

- un corpus C sur un alphabet Σ ;
- un espace E de patterns, par exemple Σ* ou l'ensemble des facteurs de C ;

l'extraction de patterns dans C consiste à rechercher dans E un ou plusieurs patterns vérifiant des critères fixés à l'avance et liés, pour chaque pattern P :

- aux caractéristiques de l'ensemble des occurrences de P :
    - nombre d'occurrences,
    - "qualité" des occurrences. Celle-ci peut se situer à la fois au plan des occurrences individuelles et au plan de leur ensemble. Supposons par exemple que les patterns sont des mots sur Σ et que la fonction d'équipollence employée est un seuillage sur une fonction de similitude numérique (cf. 2.2.2.1.2). On pourra alors exiger non seulement que la différence de longueur entre P et chacune de ses occurrences reste inférieure à un plafond fixé au départ, mais aussi que la somme des similitudes entre P et chaque occurrence excède un seuil lui-aussi fixé initialement. N.B. : Ce double critère fait partie de ceux que nous mettons typiquement en œuvre dans FlExPat / Imprology (cf. Chapitre 5).
- aux caractéristiques intrinsèques de P. Lorsque les patterns sont des mots, on prend par exemple en compte leur longueur. Typiquement, on exige alors que celle-ci soit supérieure à un seuil donné, pour éviter d'extraire des patterns trop courts — à l'extrême, des symboles.

2.2.3.3. Notion de bloc

Dans la littérature, en particulier dans celle qui a trait à la biologie moléculaire, un ensemble de mots identiques ou équipollents deux à deux, ou encore l'ensemble des occurrences d'un pattern, sont souvent appelé *bloc*.

2.2.3.4. Notion de représentant

Dans certains contextes (cf. par exemple Chapitre 5), il est souhaitable de marquer très nettement la distinction entre les aspects intensionnel et extensionnel des patterns. Dans un tel cas, nous appelons la représentation intensionnelle d'un pattern P : *représentant* (ou *prototype*) de P. La représentation extensionnelle de P, relativement au corpus C considéré, est (rappel) l'ensemble des occurrences de P dans C, ensemble de facteurs également appelé bloc.

Chapitre 3. COMPARAISON, LOCALISATION ET EXTRACTION

3.1. Introduction

3.1.1. Articulation du chapitre

Il ressort de ce que précède que la comparaison d'entités séquentielles joue un rôle déterminant aussi bien au sein du processus de localisation de patterns donnés au départ que de celui de l'extraction de (nouveaux) patterns. Corrélativement, le tour d'horizon effectué dans le présent chapitre, de concepts et algorithmes — de nature principalement combinatoire — proposés précédemment dans la littérature, concernera d'abord la **comparaison** d'entités séquentielles (3.2), puis la **localisation** (3.3) et l'**extraction** (3.4) de patterns.

3.2. Comparaison de séquences et/ou de facteurs

Qu'il s'agisse de localisation ou d'extraction de patterns, la comparaison, explicite ou implicite, entre deux séquences ou facteurs (cf. 3.1.1) joue un rôle central.

Notons cela dit que l'évaluation de la ressemblance entre entités en général, a fait l'objet de nombreux travaux dans un large éventail de disciplines ou contextes, allant de la psychologie à la théorie des sous-ensembles flous en passant par l'analyse de données (cf. par exemple [Tversky 1977 ; Zwick, Carlstein & Budescu 1987 ; Diday et al. 1982]).

Au sein de plusieurs paradigmes en intelligence artificielle la comparaison d'entités apparaît comme une tâche centrale. En apprentissage, citons par exemple le raisonnement à partir de cas [Kolodner 1993; Rifqi 1995], où l'évaluation de similitude entre cas — du moins la *partie problème* des cas, par opposition à leur *partie solution* — joue un rôle crucial dans la phase de récupération de cas dans la base.

Nous nous centrons ici sur la comparaison d'entités séquentielles. Nous verrons au 3.2.4.3 des modèles de comparaison de séquences se rapprochant davantage de modèles de comparaisons d'entités non séquentielles. Soulignons par ailleurs que nous nous concentrons sur les modèles à caractère combinatoire, par opposition, par exemple, à des modèles purement statistiques ou connexionistes (voir par exemple les travaux d'Anagnostopoulou (1997), ceux de Hörnel & Ragg (1996), ou un ouvrage général comme [Todd & Loy 1991]). Formellement, il s'agit de définir une fonction, que nous appelons *fonction de comparaison*, renvoyant le résultat de la comparaison entre deux séquences (en particulier deux facteurs de séquences). Une large majorité de travaux ont concerné le développement de fonctions *numériques*, privilégiant ainsi une vision quantitative de la comparaison de séquences. Néanmoins, dans certains cas on peut obtenir des informations qualitatives sur la ressemblance entre deux séquences. Par exemple, dans le cadre du modèle d'édition (3.2.4.2.1) il peut s'agir de la structure de mise en correspondance (*alignement*) optimale entre les différents éléments de chaque séquence (3.2.4.2.7).

Précisons que nous attachons à comparer deux entités séquentielles X et X' dans leur totalité. Il s'agit de comparaison *globale*, par opposition notamment à la comparaison *locale* où l'on rechercherait typiquement les deux facteurs de X et X' les plus similaires entre eux. (La distinction sera considérée au 3.2.6).

Précisons également que nous nous intéressons, dans la présentation qui suit, à comparer des entités de même nature, c'est-à-dire des séquences (sur le même alphabet typiquement). La comparaison, dans ce cadre, s'effectue selon la philosophie d'une *mesure de ressemblance* [Rifqi 1995]. En revanche, la comparaison entre une séquence à proprement parler et une entité séquentielle plus abstraite, comme par exemple dans les algorithmes d'extraction de patterns HPoivre et LePoivre [Sagot, Viari & Soldano 1995] présentés au 3.4, exige une généralisation (éventuellement triviale) des modèles et algorithmes présentés dans la présente section. La comparaison, dans ce cadre, s'effectue selon la philosophie d'une *mesure de satisfiabilité* [Rifqi 1996].

Toute fonction de comparaison est sous-tendue par ce que nous appelons un *modèle de comparaison de séquences*, correspondant à une vision, une conception spécifiques de la ressemblance entre entités séquentielles. Par exemple, le modèle d'édition correspond à une vision transformationnelle : deux séquences sont déclarées similaires si on peut obtenir l'une à partir de l'autre en effectuant des modifications (*éditions*) "raisonnables" en nombre et/ou en nature. La prise en compte de la philosophie sous-jacente à chaque modèle de comparaison fournit un critère pour le choix d'un tel modèle en vue d'une application particulière.

Dans la suite nous nous concentrerons exclusivement sur les fonctions de comparaison numériques, sauf lorsque cela sera explicitement mentionné. Deux types de fonctions de comparaison existent :

- Les fonctions mesurant la dissimilitude entre deux séquences, en particulier les *distances* (cf. ci-dessous). L'opération consistant à déterminer l'image d'un couple de séquences par une telle fonction sera appelée *calcul de dissimilitude*.
- Les fonctions mesurant la similitude entre deux séquences, l'opération associée étant nommée *calcul de similitude*.

La Figure 3-1 visualise cette taxinomie.

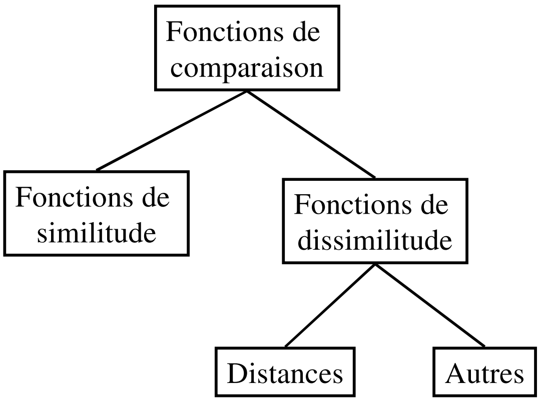


Figure 3-1. Taxinomie de fonctions de comparaison entre entités séquentielles

Dans le cadre le plus général il est impossible de donner une définition discriminante de ces deux types de fonctions sans introduire une certaine récursivité. En effet dans ce cadre général, une fonction de comparaison est définie par rapport à l'interprétation que l'on fait des valeurs (numériques) qu'elle renvoie :

- pour une *fonction de dissimilitude* Dissim celles-ci sont supposées croissantes avec la "vraie" dissimilitude existant entre les séquences comparées.

- pour une *fonction de similitude* Simil, les valeurs renvoyées sont supposées croissantes avec la "vraie" similitude existant entre les séquences comparées.

Il n'existe en général pas de définition objective ou unique de cette "vraie" similitude/dissimilitude. En pratique cependant, cela ne pose pas de problème et les seules contraintes que l'on impose éventuellement (cf. ci-dessous) sont des contraintes objectives comme, par exemple :

- dans le cas des fonctions de similitude, la positivité (i.e. $\mathrm{Simil}(\Sigma^* x \Sigma^*) \subset \mathbf{R}^+$)
- dans celui des fonctions de dissimilitude, la nullité pour toute autocomparaison, i.e. $(\forall X\ \Sigma^*)\ (\mathrm{Dissim}(X,X)=0)$.

Remarque : ces deux contraintes interviennent dans la définition, respectivement, d'un *indice de similarité* et d'un *indice de dissimilarité* [Diday et al. 1982 ; Saporta 1990]. Souhaitant conserver un degré de généralité maximal, nous ne faisons pas appel à ces notions classiques et parlons plutôt, plus généralement, de fonctions de similitude/dissimilitude. Parmi les modèles que nous présenterons ou que nous proposerons, certains font par exemple intervenir des fonctions de similitude ne vérifiant pas la propriété de positivité (par exemple [Mongeau & Sankoff 1990 ; Rolland & Ganascia 1998a ; 1996b] — cf. 4.6.1.4).

3.2.1. Distances

Au sein des fonctions de dissimilitude, les distances sont des fonctions particulières caractérisées par les classiques *propriétés métriques.* Soit Dist une fonction de comparaison :

$$\mathrm{Dist}: \Sigma^* x \Sigma^* \rightarrow \mathbf{R} \text{ (ou } \mathbf{Z}\text{, ou } \mathbf{N} \ldots)$$

Dist est une distance sur Σ* si et seulement si elle possède les quatre propriétés suivantes :

- Positivité

$$\{\forall\ (X,Y) \in \Sigma^* x \Sigma^*\}\ \{\mathrm{Dist}(X,Y) \geq 0\}$$

- Commutativité (ou symétrie)

$$\{\forall\ (X,Y) \in \Sigma^* x \Sigma^*\}\ \{\mathrm{Dist}(X,Y)=\mathrm{Dist}(Y,X)\}$$

- Condition nécessaire et suffisante de nullité

$$\{\forall\ (X,Y)\in \Sigma^{*}x\Sigma^{*}\}\ \{\{Dist(X,Y)=0\} \Leftrightarrow \{X=Y\}\}$$

- Inégalité triangulaire

$$\{\forall\ (X,Y,Z)\in (\Sigma^{*})^{3}\}\ \{Dist(X,Y)=Dist(X,Z)+Dist(Z,Y)\}$$

En pratique, on exige un ensemble de propriétés moindre que pour des distances au sens strict, en particulier en ce qui concerne l'inégalité triangulaire [Mongeau & Sankoff 1990]. Cette dernière est particulièrement problématique en musique, du fait de l'existence de domaines de valeur cycliques [Rolland & Ganascia 1998a].

Le fait d'assimiler ces pseudo-distances à des distances permet notamment d'utiliser certaines techniques d'analyse de données exigeant les propriétés métriques (cf. par exemple [Mongeau & Sankoff 1990])

Soulignons que la majorité des fonctions de comparaison qui seront présentées possèdent la propriété de symétrie.

3.2.2. Fonctions de similitude versus fonctions de dissimilitude/distances

Il existe une dualité évidente entre les notions de similitude et de dissimilitude. En particulier, il est toujours possible de définir une fonction de similitude à partir d'une fonction de dissimilitude et vice versa. Trivialement, étant donnée une fonction de similitude Dissimil, la fonctions (-Simil) — et, plus généralement, les fonctions de la forme (C-Simil) où C est une constante numérique positive typiquement — sont des fonction de similitude candidates, et inversement.

Les distances présentent des avantages, déjà évoqués plus haut, liés à leurs propriétés métriques. Cependant, lorsqu'on se place dans le cadre du modèle d'édition présenté ci-dessous, tout problème solvable avec des distances peut aussi être résolu avec des fonctions de similitudes, l'inverse n'étant pas vrai [cf. par exemple Waterman 1995, chap. 9]. Dans la suite, les fonctions présentées dans le cadre des différents modèles seront en majorité des fonctions de similitude, les exceptions correspondant à des distances "importantes" (soit parce qu'elles sont très utilisées, soit parce qu'elles ont une proéminence théorique particulière).

3.2.3. Fonction d’équipollence associée à une fonction de comparaison numérique

A partir d’une fonction de comparaison numérique, par exemple une fonction de similitude Simil, il est possible de définir une fonction d’équipollence (cf. 2.2.2) par un test sur les valeurs prises par Simil, typiquement un seuillage. Plus précisément, étant donné un nombre *Seuil* nous associons à Simil la fonction d'équipollence Eq définie par :

$$\text{Eq} : \Sigma^* \times \Sigma^* \rightarrow \mathbf{Bool}$$

$$\text{Eq}(X,Y) = \text{VRAI si Simil}(X,Y) \geq \text{seuil, FAUX sinon}$$

3.2.4. Différentes catégories de modèles

Un certain nombre de modèles ont été proposés pour comparer des entités séquentielles. Nous les groupons en deux catégories, en fonction des possibilités offertes par les modèles quant à la *structure de correspondance* (cf. 1.1) entre les éléments respectifs des deux séquences :

- Les modèles pour lesquels cette structure, que l’on pourrait appeler aussi structure d’appariement, est *fixe*, imposée au départ.
- Les modèles pour lesquels cette structure peut varier (dans un intervalle de possibilités clairement délimité, bien entendu).

Pour cette seconde catégorie, nous présenterons le *modèle d'édition* et sa famille de variantes. Ce modèle, utilisé dans de nombreux domaines, est celui sur lequel opèrent les algorithmes d'extraction que nous proposons (Chapitre 5).

La catégorisation proposée ci-dessus n'est pas la seule possible. Alternativement nous aurions pu, par exemple, grouper les modèles en fonction de la complexité des algorithmes de calcul de la similitude (cf. 3.2.6) associés à chacun d'entre eux.

L’ordre de notre présentation va du plus spécifique au plus général. Autrement dit, dans cette succession hiérarchique de modèles, chacun subsume son prédecesseur. Nous traduisons l’existence des liens de subsomption (généralité) en introduisant le terme *sous-modèle* ; par exemple (voir plus bas) le modèle de Hamming classique est un sous-modèle du modèle d’édition à coûts/contributions constants, lui-même sous-modèle du modèle d’édition à coûts/contributions variables. Par ailleurs, la plupart des modèles de ces deux catégories font intervenir un certain nombre de paramètres — au sens large du terme — par exemple, pour le modèle d'édition, l'ensemble des *types d'éditions* ou des *types d'appariements* autorisés et les différentes *fonctions de contribution* associées à ces types. Un modèle de comparaison dont les paramètres sont spécifiés sera qualifié d'*instance* du modèle. Ainsi (cf. ci-après), la classique distance de Hamming est une instance du modèle de Hamming, et la *distance de Levenshtein avec substitutions* est une instance du modèle d'édition à coûts constants.

3.2.4.1. Modèles à structure de correspondance fixe

Le principe sous-tendant un certain nombre de modèles consiste, schématiquement, à mettre les deux séquences en regard direct l'une de l'autre et à en évaluer la similitude (resp. la dissimilitude) sur la base de l'alignement, dit vertical, obtenu. Il est important de noter que cela exige l'égalité des longueurs des deux séquences X et Y. Le résultat final s'obtient à partir de l'ensemble des similitudes (resp. dissimilitudes) individuelles, résultats de la comparaison des éléments X[i] et Y[i] composant chaque colonne, en général par sommation.

Plusieurs exemples classiques de distances peuvent être donnés qui instancient la catégorie des modèles à structure de correspondance fixe. Dans tous les cas il y a sommation, sur l'ensemble des indices, d'un terme d(X[i],Y[i]) où d est elle-même une distance sur l'alphabet Σ.

- La *distance de Hamming* mesure le nombre d'indices auxquels les deux séquences diffèrent :

$$(X,Y) \mapsto \text{distHamming}(X,Y) = \sum_{i=1}^{L} \delta\,(X[i],Y[i])$$

Equation 3-1

avec :

$$(\forall i \in \mathbf{N}_L)\ (\delta\ (X[i],Y[i])=0 \text{ si } X[i]=Y[i],\ 1 \text{ sinon})$$

Equation 3-2

Par exemple, la distance de Hamming entre les chaînes de caractères « musical » et « médical » est égale à 2.

Les distances qui suivent exigent de plus que les éléments comparés soient des réels.

- La *distance euclidienne* est définie par :

$$(X,Y) \mapsto \text{distEucl}(X,Y) = (\sum_{i=1}^{L} (X[i] - y[i])^2)^{1/2}$$

- La distance « des pâtés de maison » (*city block distance*) est définie par :

$$(X,Y) \mapsto \text{distCityBl ock}(X,Y) = \sum_{i=1}^{L} |X[i] - Y[i]|$$

Remarque : Ces deux distances sont des instances particulières — ayant l'avantage d'avoir chacune un nom d'usage en raison de leur large utilisation — d'une famille de distances paramétrée par un entier p (distance de Minkowski) :

$$(X,Y) \mapsto \text{dist } p(X,Y) = (\sum_{i=1}^{L} |X[i] - Y[i]|^{p})^{1/p}$$

On a ainsi : distEucl=dist$_2$ et distCityBlock=dist$_1$.

Nous appelons *modèle de Hamming* le modèle général de comparaison de séquences fondé sur une structure de correspondance verticale. Les distances que nous venons de présenter s'inscrivent donc dans le cadre de ce modèle.

3.2.4.1.1. Utilisation musicale

Dans le système EMI, Cope (1992,1990) utilise une variante de la distance de Hamming pour comparer des segments mélodiques. Ceux-ci sont représentés par des séquences (facteurs) dont les éléments sont, suivant le paramétrage choisi, soit des intervalles chromatiques — nombres algébriques de demi-tons entre chaque note et la suivante —, soit des durées, soit encore des couples (intervalle chromatique, durée). Cette classique redescription en intervalles chromatiques rend la similitude entre deux séquences invariante aux *tranpositions chromatiques* (par opposition à *diatonique*) pouvant être appliquée à l'une de ces séquences ou au deux.

Remarque : les propriétés d'invariance de différentes descriptions aux transformations telles que la transposition font l'objet du 4.7.7.3.3.

Avec les valeurs de paramètres les plus « strictes », le critère d'équipollence deux segments F et F' est leur **égalité**, ce qui exige en particulier qu'ils aient même longueur.

Cope offre la possibilité, via un ajustement de paramètres, d'apporter un raffinement à ce schéma basique en remplaçant, dans la distance de Hamming, l'égalité stricte des éléments appariés verticalement par une égalité approximative. Cela donne lieu à une fonction de dissimilitude (qui n'est plus une distance) comme suit :

$$Dissim_{EMI}(X,Y) = \sum_{i=1}^{L} \delta'(X[i], y[i])$$

δ' étant définie à partir d'une relation d'égalité approximative ≈ par :

$$(\forall i \in \mathbf{N}_L)\ (\delta'(X[i],Y[i])=0 \text{ si } X[i] \approx Y[i], 1 \text{ sinon})$$

La relation ≈ est basée sur une "tolérance" t, paramètre fixant la dissimilitude maximale pouvant exister entre X[i] et Y[i]. Par exemple, lorsque le paramétrage fixe une redescription en 'intervalles chromatiques seulement', t représente le nombre maximal de demi-tons dont peuvent différer les deux notes. Pour ce paramétrage, t est typiquement réglé à la valeur 1. Un tel réglage permet de tolérer les substitutions entre tons et demi-tons couramment employées en musique tonale par les compositeurs "*in order to remain within a diatonic framework when sequencing*" [Cope 1991b].

Par ailleurs, la condition de nullité de la distance ou de la fonction de dissimilitude peut être assouplie par le biais d'un seuillage paramétré. Au lieu d'exiger que la fonction $\mathrm{Dissim}_{\mathrm{EMI}}$ soit égale à zéro, un paramètre k, entier strictement positif, fixe pour cette fonction un seuil maximal en dessous duquel les deux segments sont jugés équipollents.

3.2.4.1.2. Modèle de Hamming valué

Une variante de la distance de Hamming est obtenue en attribuant des valeurs spécifiques aux différentes substitutions intervenant dans la mise en correspondance verticale des deux séquences comparées (cf. par exemple [Schuler, Altschul & Lipman 1991]). En clair, l'Equation 3-1 ne change pas mais en revanche la fonction δ ne prend plus nécessairement des valeurs binaires (cf. Equation 3-2). Au contraire, $\delta(X[i],Y[i])$ peut prendre des valeurs graduelles censées refléter la dissemblance existant entre *X[i] et Y[i]*. Les contraintes à imposer à δ sont la positivité et la nullité pour tout couple d'éléments identiques :

$$\{\forall x\}\ \{\delta(x,x)=0\}$$

Il est important de noter que suivant la fonction δ choisie, la fonction de dissimilitude obtenue n'est plus nécessairement une distance. Bien entendu, moyennant l'utilisation, à la place de δ, d'une fonction σ traduisant la *ressemblance* entre éléments, on obtient une fonction de *similitude*.

Un tel modèle peut être vu comme un cas singulier (dégénéré) de modèle d'édition valué dans lequel le seul type d'édition autorisé est le type Substitution (cf. 3.2.5). (Plus généralement, le modèle de Hamming peut à l'extrême être décrit comme un cas particulier de modèle d'édition où seules les substitutions sont autorisées).

3.2.4.1.3. Modèles particuliers

En dehors des modèles à portée générale que nous venons de présenter, quelques autres modèles ont été proposés qui se basent sur une structure de correspondance fixe. Nous ne présenterons ici qu'un seul de ces modèles destinés à un domaine précis, en l'occurrence la musique. A proprement parler, il s'agit non d'un seul mais d'une famille de modèles de comparaison. Ce sont les 26 modèles implémentés dans le système de Stech (1981) que nous mentionnions plus haut (2.1.3.4). Donnant lieu à des fonction de similitude booléenne, ils sont pour la plupart fondés sur la notion de *transformation globale*.

Critiquant le biais de "*fixed content*" dont étaient entachés les systèmes antérieurs, Stech a développé un système qui ne recherche pas seulement des répétitions exactes de segments. Il recherche aussi des répétitions de nature particulière, dans lesquelles le second segment (la "copie") se déduit du premier ("l'original") par une transformation globale. Stech a défini 26 types de telles transformations. Ce sont des combinaisons des opérations traditionnelles de rétrograde (symétrie horizontale) et d'inversion (symétrie verticale) appliquées à la séquence des hauteurs/intervalles ou à celle des durées. Ainsi, moyennant l'une des 26 transformations une séquence renversée dans le temps de directions d'intervalles peut être jugée équipollente à la séquence originale.

A chaque type de transformation correspond une structure d'appariement fixe. Par exemple, pour le 'rétrograde par rapport les durées' la structure de correspondance est celle illustrée à la Figure 3-2 et traduite par l'équation suivante :

$$\{\forall i \in \mathbf{N}m\}\ \{F[i].\text{durée} = F'[m-i+1].\text{durée}\}$$

où F et F' sont les deux segments concernés, de longueur commune m et où la durée d'une note N est écrite N.durée.

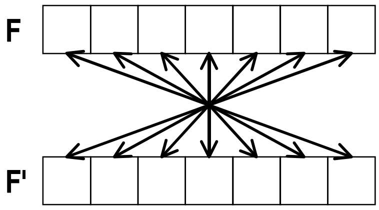


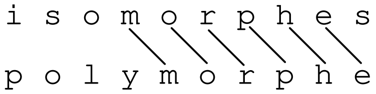
Figure 3-2. Structure de correspondance fixe pour une transformation globale de type "rétrograde"

3.2.4.1.4. Discussion

Les distances que nous avons présentées dans cette section n’ont de sens que si la structure de mise en correspondance imposée – et ses conséquences en termes contraintes sur les longueurs – en ont. Considérons par exemple la distance de Hamming entre les deux chaînes de caractères suivantes, que nous disposons naturellement en regard l'une de l'autre en vertu de la structure de correspondance verticale qu'impose ce genre de distance :

```
i s o m o r p h e s
p o l y m o r p h e
```

Cette distance est égale à 8, qui est la valeur maximale pour des chaînes de cette longueur, en dépit de leur évidente ressemblance liée à la présence commune du facteur "morphe". Une structure de correspondance intuitivement plus pertinente ici serait par exemple celle-ci:



voire celle de la Figure 3-3, tenant également compte de la présence commune du "o" initial :

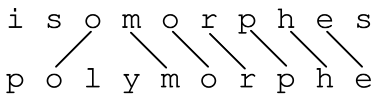


Figure 3-3. Structure possible de mise en correspondance des chaînes de caractères 'isomorphes' et 'polymorphe'

En musique des notions aussi importantes que la variation ou l'ornementation au sens large font fréquemment intervenir des changements locaux résultant en une structure de correspondance non verticale. Ainsi, les méthodes à structure de correspondance fixe ne sont susceptibles d’être appliquées qu’à des problèmes musicaux très particuliers, situés en tout état de cause dans le cadre de musique *composée* et non *improvisée* (voir à ce sujet la section 4.2.1).

3.2.4.2. Modèles à structure de correspondance variable

La *distance d’édition* est un concept désormais classique en algorithmique sur les séquences ainsi que dans d'autres disciplines scientifiques comme la reconnaissance des formes [Milgram 1993]. (En toute rigueur, il serait plus exact de parler plutôt d’une famille de distances, comme nous allons le voir.) Cette distance s'inscrit dans le cadre d'un modèle plus général que nous appelons *modèle d'édition*. Ce modèle a été utilisé, et continue de l'être, dans un panorama très large de domaines d'application, allant de la biologie (cf. par exemple [Sankoff & Kruskal 1983, Waterman 1995, ou Crochemore & Rytter 1994] à la musique ([Dannenberg 1984, Mongeau & Sankoff 1990, Stammen & Pennycook 1993; Rowe 1993] et, indirectement, [Chemiller 1994]). Un passage en revue succinct de l’application du modèle d’édition en musique est proposé dans (Stammen et Pennycook 1993). D'autres domaines sont par exemple (cf [Kruskal 1983]) :

- Le traitement de la parole
- L'informatique (comparaison de fichiers, en particulier de textes ; correction d'erreurs de saisie au clavier)
- Le codage de l'information

3.2.4.2.1. Le modèle d'édition : formulation générale

Le modèle d’édition est, en termes très schématiques, sous-tendu par la considération suivante : ***"pour évaluer la similitude entre deux séquences, on caractérise de quelle(s) façon(s) on peut transformer ("éditer") la première séquence***

***pour obtenir la seconde*[10]"**. Dans cette formulation générale, deux précisions fondamentales doivent être données :

- La *transformation* d'une séquence, désignée par *séquence origine*, en une *séquence destination* consiste en un ensemble d'*éditions*, également appelées opérations ou transformations élémentaires, appartenant à un ou plusieurs types fixés au départ (en fonction notamment du genre de séquences dont on s'occupe). Les types standard, de loin les plus employés, sont les suivants:
  - *Insertion* [d'un élément dans la séquence destination].
  - *Délétion* [d'un élément de la séquence origine]. N.B. on utilise aussi le mot "suppression", voire le mot "destruction". Nous choisissons d'employer dans ce mémoire le néologisme délétion (basé sur l'anglais, et de plus en plus répandu — cf. par exemple [Sagot 1996]), pas seulement parce qu'il est plus court mais aussi parce qu'il permet l'emploi d'un autre néologisme tiré de l'anglais : indel (cf. ci-dessous).
  - *Substitution* (ou *changement*) [d'un élément de la séquence origine en un élément différent de la séquence destination]
- La **caractérisation** dont il est question dans la formulation générale suppose qu'on se soit fixé une fonction permettant d'évaluer chaque transformation, c'est-à-dire chaque ensemble d'éditions permettant de passer d'une séquence à l'autre. Là aussi, la manière dont cette fonction est définie dépend du genre de séquences auxquelles on a affaire. La plus simple consiste à **compter** le nombre d'éditions composant la transformation.

Exemple 3-1

Il est possible de transformer la chaîne de caractères "isomorphe" en la chaîne "polymorphe" en utilisant une substitution (i→p), deux insertions (l et y) et une délétion (s). Notons qu'il s'agit cette fois-ci de séquences de longueurs différentes. La Figure 3-4 propose une représentation graphique simplifiée de cette transformation appelée *trace.* Une trace est un diagramme sur lequel figurent :

- La séquence origine et, au dessous, la séquence destination.

---

[10] Ou l'inverse, en vertu de la symétrie intrinsèque qui intervient ici et sur laquelle nous reviendrons ultérieurement.

- Un certain nombre de traits reliant :
  - les couples d'éléments identiques mis en correspondance (*identités*), et
  - les couples d'éléments faisant l'objet d'une substitution — ici i et p.

(Nous reviendrons plus loin sur la notion de trace en tant que mode de représentation d'une transformation). En supposant que toute transformation soit évaluée en comptant le nombre d'éditions qui la compose, la transformation représentée sur la Figure 3-4 a une évaluation égale à 4.

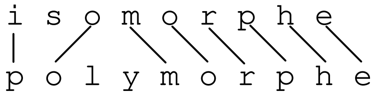


Figure 3-4. Représentation graphique (trace) de la transformation de la chaîne de caractères "isomorphe" en "polymorphe"

Cet ensemble d'éditions n'est pas le seul possible. Par exemple, une autre transformation consiste à supprimer un à un tous les éléments de "isomorphe" puis à insérer un à un tous ceux de "polymorphe", l'évaluation étant alors égale à 19 (9 délétions + 10 insertions). †

Il existe en fait un très grand nombre d'ensembles différents d'éditions transformant "isomorphe" en "polymorphe", dont les évaluations diffèrent. Il apparaît assez naturel de définir la similitude (ou la dissimilitude) entre les deux séquences en fonction de l'éventail des valeurs prises par ces différentes évaluations. On peut ainsi décider que la dissimilitude entre "isomorphe" et "polymorphe" est le *nombre mimimal* d'éditions permettant de transformer la première séquence en la seconde, ou, ce qui est équivalent, la seconde en la première. On pourrait montrer, en utilisant l'un des algorithmes qui seront présentés au 3.2.6, que ce nombre minimal est égal à 4, et est donc atteint (entre autres) pour la transformation représentée en Figure 3-4. Ici encore, une fois l'ensemble des types de transformation fixé, cette définition de la dissimilitude entre séquences n'est pas unique. On pourrait par exemple la définir comme le *nombre moyen* d'éditions pris sur l'ensemble de toutes les transformations possibles (même si, compte tenu de ce que nous verrons au 5.6.3, cela serait algorithmiquement irréaliste en raison du nombre gigantesque de transformations différentes).

Dans cette section, nous nous sommes surtout attachés à donner l'intuition générale du modèle d'édition. Il nous faut à présent fixer un certain nombre de concepts.

Nous supposons que X et X' sont les séquences à comparer. Dans les exemples que nous donnerons, nous prendrons les séquences suivantes sur l'alphabet latin Σ°={a..z} : X="isomorphe" et que X'="polymorphe".

3.2.4.2.2. Ensemble des types d'édition

Nous faisons une distinction claire entre, d'un côté, les types d'éditions, dont nous notons le nom avec une majuscule, et, de l'autre, leurs instances. Ainsi dans l'alignement de la Figure 3-4 la mise en correspondance du "i" de la première séquence avec le "p" de la seconde séquence se traduit par la substitution $\begin{pmatrix} i \\ p \end{pmatrix}$, instance particulière du type Substitution.

Dans le modèle d'édition les types d'édition autorisés doivent être clairement fixés au départ. Différents types d'édition peuvent être utilisés. Dans un très grand nombre de travaux, quel qu'en soit par ailleurs le domaine d'application, l'ensemble des types d'éditions autorisées est composé du type insertion, du type délétion et du type substitution. Insertion et délétion sont clairement des opérations duales, une insertion pouvant être vue comme une délétion suivant la séquence que l'on considère comme origine et celle que l'on considère comme destination. Il est souvent utile de désigner collectivement les deux types d'édition, nous employons pour cela le terme Indel (un néologisme qui n'est plus si nouveau que ça.…)

Au-delà de ces types standard, Substitution, Insertion et Délétion, d'autres types peuvent être proposés dont la pertinence dépend du domaine où l'on se place :

- *InsertionMultiple*, correspondant à une suite contiguë de k insertions (resp k délétions) où k est un entier strictement positif non nécessairement égal à 1.
- *DélétionMultiple*, qui correspond bien sûr à l'édition inverse de l'insertion multiple.
- *Consolidation-Fragmentation* ou — dans le formalisme du Dynamic Timewarping, — *Compression-Expansion* [Kruskal & Liberman 1983; Mongeau & Sankoff 1990]. Le diagramme de trace de la Figure 3-5 illustre le concept de fragmentation sur des séquences monodiques dans le genre jazz Be-bop : la première note du premier passage est *fragmentée* en un groupe de 3 notes. (Ce genre de répétition de note — attaque multiple —est un effet technique/stylistique courant chez un saxophoniste comme Charlie Parker). Les traits reliant, sur la trace, les 4 dernières paires de notes traduisent des identités. Il existe une

différence relativement subtile entre compression-expansion et délétion-insertion, mais sa présentation sortirait du cadre de ce mémoire (voir par exemple [Kruskal & Liberman 1983, p. 150]).

- *Permutation* (*"swap"*), correspondant à l'échange de deux[11] éléments adjacents [Wagner & Fisher 1974]. (Le terme de *transposition* est également utilisé, mais nous l'évitons en raison de sa signification musicale.)
- *Substitution multiple* (ou généralisée), correspondant au remplacement d'un groupe d'éléments contigüs (facteur) par un autre groupe distinct. Ce type d'édition peut être vu comme une généralisation des deux types qui précèdent.

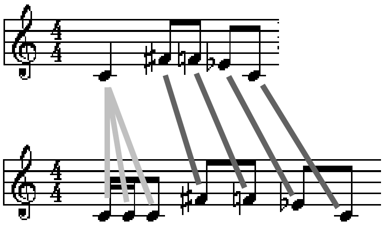


Figure 3-5.Exemple de *fragmentation* dans le domaine musical

3.2.4.2.3. Notion d'appariement

Nous introduisons la notion d'appariement, qui généralise utilement la notion d'édition. Intuitivement, un appariement traduit la mise en correspondance de deux facteurs dont l'un est éventuellement vide. Lorsqu'une édition se traduit par la mise en correspondance de deux facteurs non vides, elle est représentée par un appariement *propre*. C'est le cas par exemple de toute substitution (cf. ci-dessus), pour laquelle les deux facteurs sont de longueur 1. Lorsqu'en revanche une édition se traduit par l'insertion ou la délétion d'un facteur, elle est représentée par un appariement *impropre*. C'est le cas par exemple de toute insertion ou délétion, pour lesquelles le facteur est de longueur 1. Métaphoriquement, une insertion ou une délétion sont vues comme l'appariement d'un facteur non vide avec un facteur vide.

L'un des intérêts de la notion d'appariement, particulièrement dans le cadre d'un modèle de **similitude**, est de permettre la représentation explicite des **identités** déjà évoquées, qui traduisent la mise en correspondance de facteurs (ou d'éléments)

---

[11] N.B. : Rien n'empêche d'envisager plus généralement des permutations de k éléments, où k=3 ou davantage. Néanmoins la combinatoire qui en résulte est rapidement problématique, puisqu'il faut envisager 6 permutations différentes pour k=3, 24 pour k=4 …

identiques. Nous verrons que, contrairement à celle que nous avons utilisée dans l'Exemple 3-1, les fonctions d'évaluation de transformations plus riches peuvent prendre en compte explicitement les identités, en plus des substitutions et des indels. Un autre intérêt central est que la notion d'alignement (3.2.4.2.7), centrale dans le modèle d'édition, se formule très naturellement à partir de celle d'appariement — plus qu'à partir de celle d'édition.

Appariement propre

X et X' étant 2 séquences, nous appelons appariement propre sur (X,X') tout couple (F,F') où F est un facteur non vide de X et F' un facteur non vide de X'. Un tel appariement est visualisé par une matrice colonne :

$$\begin{pmatrix} F \\ F' \end{pmatrix}$$

Exemple 3-2

$\begin{pmatrix} mor \\ morph \end{pmatrix}$, $\begin{pmatrix} i \\ p \end{pmatrix}$ et $\begin{pmatrix} o \\ o \end{pmatrix}$sont des appariements propres pour X="isomorphe" et X'="polymorphe". †

Appariement impropre

X et X' étant 2 séquences, nous appelons appariement impropre sur (X,X') tout couple (F, λ) ou (λ, F), F et F' étant deux facteurs non vides de X. La séquence vide λ étant visualisée par le symbole "-" (lequel, comparé à "λ", se distingue mieux, à l'œil, des autres symboles), un tel appariement est visualisé par l'une des deux matrices colonnes suivantes :

- $\begin{pmatrix} - \\ F \end{pmatrix}$, qui correspond à l'insertion de F. Lorsque |F|>1, nous parlons de multiInsertion et lorsque |F|=1, de monoInsertion ou plus simplement d'insertion, comme ci-dessus.
- $\begin{pmatrix} F \\ - \end{pmatrix}$, qui correspond à la délétion de F, qui est une multiDélétion si |F|>1 et une [mono]délétion sinon.

Unification

En vue de donner une définition unifiée d'un l'appariement, notons $\underline{\text{facts}}(X)$ l'ensemble de tous les facteurs de la séquence X, y compris le facteur vide, c'est-à-dire :

$$\underline{\text{facts}}(X)=\text{facts}(X)\cup\{\lambda\}$$

(Le trait horizontal rajouté sur 'facts' évoque le tiret utilisé pour visualiser graphiquement la séquence vide). Alors un appariement sur les séquences X et X', qu'il soit propre ou impropre, peut être défini comme étant un élément de $\underline{\text{facts}}(X) \times \underline{\text{facts}}(X')$, avec la restriction que (λ,λ) n'est pas autorisé. Nous donnons ci-dessous la définition finale :

Définition 3-1

X et X' étant deux séquences, un appariement *a sur (X,X') est un élément de* $(\underline{\text{facts}}(X)\times\underline{\text{facts}}(X')) - \{(\lambda,\lambda)\}$

Sequences origine et destination

Soit a=(Ꞙ,Ꞙ') un appariement.

- Ꞙ est appelé *séquence origine* de a, éventuellement vide, noté seqOrig(a)
- Ꞙ', est appelé sa *séquence destination*, noté seqDest(a).

3.2.4.2.4. Correspondance (univoque) entre éditions et appariements

De même que nous avons défini les types d'éditions et les instances de ces types, nous définissons les *types d'appariement,* dont le nom est noté avec une majuscule, et leurs instances (appelées simplement appariements). Naturellement, pour chaque type d'édition décrit plus haut nous introduisons le type d'appariement qui lui est isomorphe et auquel nous attribuons le même nom : Insertion, Délétion, Substitution, Permutation, MultiInsertion, MultiDélétion, Fragmentation / Consolidation, SubstitutionGénéralisée.

Comme nous l'avons dit, la notion d'appariement est plus générale que celle d'édition parce que la mise en correspondance de deux éléments identiques, traduite par une instance d'un type particulier d'appariement, n'a pas de traduction en terme d'édition (c'est pour ainsi dire une "non-édition"). Ce type particulier est appelé *Identité* ("*continuation*"*).* Ainsi, nous introduisons le type d'appariement Remplacement qui possède deux sous-types, Identité et Substitution. La Figure 3-6 illustre les différents types d'édition standard.

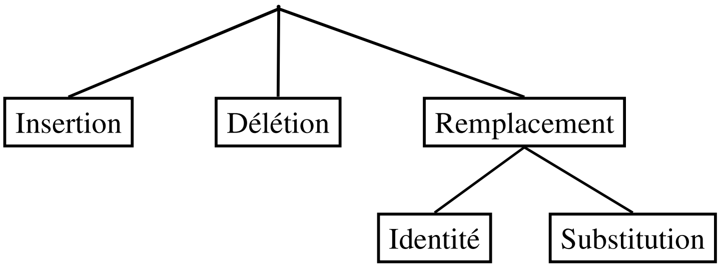


Figure 3-6. Les trois types d'appariements standard, et les deux sous-types de Remplacement

Nous introduisons une notion fort utile, celle de *signature*, qui permet de décrire de manière compacte un type d'appariement.

3.2.4.2.5. Signature d'un [type d'] appariement

Commençons par définir cette notion pour une instance d'un type d'appariement. Nous définissons la *signature* d'un appariement *a* comme étant le couple d'entiers formé du nombre d'éléments de la séquence origine de *a* et du nombre d'éléments de la séquence destination de *a*. Ainsi :

- La signature de $\begin{pmatrix} - \\ y \end{pmatrix}$ est (0,1) et il est clair qu'il en est de même pour toute autre (mono)insertion.
- Celle de $\begin{pmatrix} s \\ - \end{pmatrix}$ est (1,0) et il en est de même pour toute autre (mono)délétion.
- Celles de $\begin{pmatrix} i \\ p \end{pmatrix}$ et $\begin{pmatrix} o \\ o \end{pmatrix}$ sont toutes deux égales à (1,1), et il en est de même pour tout autre remplacement
- Celle de $\begin{pmatrix} mor \\ morph \end{pmatrix}$ est (3,5), celle de $\begin{pmatrix} iso \\ - \end{pmatrix}$ est (3,0).

Remarque : pour caractériser les liens de subsomption évoqués plus haut entre le type SubstitutionGénéralisée et d'autres types, observons que :

- Une consolidation de k éléments en 1 peut être vue comme une substitution généralisée de signature (k,1)

- Une fragmentation d'un élément en k éléments peut être vue comme une substitution généralisée de signature (1,k).
- Une permutation peut être vue comme une substitution généralisée de signature (2,2). Les permutations de k éléments évoquées au 3.2.4.2.2 ont la signature (k k).

Définir la signature d'un **type** d'appariement requiert une légère généralisation du fait que certains types, comme le type InsertionMultiple ou le type RemplacementGénéralisé, peuvent avoir des instances dont les signatures diffèrent. Nous nous contenterons de poser que :

- La signature, qualifiée de *variable*, d'un tel type d'appariement est égale à l'ensemble des signatures que peuvent posséder ses différentes instances.
- La signature, qualifiée de *fixe*, de tout autre type d'appariement est égale à celle de l'une quelconque de ses instances.

Le concept de signature est compatible avec l'organisation hiérarchique de l'ensemble des types d'appariements (représentée partiellement sur la Figure 3-6) : par exemple, les types Identité et Substitution héritent de leur parent Remplacement la signature fixe (1,1).

3.2.4.2.6. Ensemble des types d'appariements autorisés

Nous avons vu qu'un paradigme d'analyse et d'évaluation de transformation est, entre autres, déterminé par l'ensemble des types d'appariements que l'on autorise. Cet ensemble sera désigné dans la suite par ETAA. Nous notons $ETAA_{standard}$ = {Insertion, Délétion, Remplacement}, et qualifions de *standard* tout modèle d'édition pour lequel $ETAA=ETAA_{standard}$.

3.2.4.2.7. Notion d'alignement

D'une manière générale, un *alignement* est une collection ordonnée d'appariements (c'est-à-dire une séquence d'appariements, mais pour des raisons évidentes nous préférons éviter d'employer ici aussi le terme "séquence"). En pratique, nous nous intéresserons non pas à des alignements dans l'absolu, mais à des alignements entre deux séquences données.

Définition

Soient X et X' deux séquences. Un alignement entre X et X' *est une collection ordonnée A=a1..ap d'appariements sur (X,X') telle que :*

- *La concaténation (dans cet ordre) de seqOrig(a1),..., seqOrig(ap) est égale à X.*
- *La concaténation (dans cet ordre) de seqDest(a1),..., seqDest(ap) est égale à X'.*

Si tous les appariements composant A sont des appariements propres, on dit que A est un alignement sans trou *entre X et X'.*

Exemple 3-3

L'alignement dont la suite des appariements est indiquée ci-dessous est un alignement entre 'isomorphe' et 'polymorphe', description possible de la transformation visualisée sur la Figure 3-4 page 72 :

$$\begin{pmatrix} i \\ p \end{pmatrix}, \begin{pmatrix} s \\ - \end{pmatrix}, \begin{pmatrix} o \\ o \end{pmatrix}, \begin{pmatrix} - \\ l \end{pmatrix}, \begin{pmatrix} - \\ y \end{pmatrix}, \begin{pmatrix} m \\ m \end{pmatrix}, \begin{pmatrix} o \\ o \end{pmatrix}, \begin{pmatrix} r \\ r \end{pmatrix}, \begin{pmatrix} p \\ p \end{pmatrix}, \begin{pmatrix} h \\ h \end{pmatrix}, \begin{pmatrix} e \\ e \end{pmatrix},$$

†

Visualisation :

Plutôt que de visualiser la succession de tous les appariements — ce qui veut dire beaucoup de parenthèses, cf. Exemple 3-3 ! — on omet les parenthèses des matrices colonnes représentant les appariements pour n'en garder que les colonnes de contenu. Pour l'alignement de ce même exemple, on obtient :

$$\begin{bmatrix} i & s & o & - & - & m & o & r & p & h & e \\ p & - & o & l & y & m & o & r & p & h & e \end{bmatrix}$$

Séquence origine, séquence destination

Comme nous l'avions fait pour les appariements, nous définissons ici ces deux notions très utiles, en particulier lorsque nous avons affaire à des alignements locaux (voir ci-dessous) : la *séquence origine* (resp. *destination*) d'un alignement A est le résultat de la concaténation des séquences origine (resp. destination) des appariements qui le composent.

Remarque : Bien entendu, les éventuels symboles "-" figurant dans certains appariements disparaîssent, en vertu du fait que λ est élément neutre pour la concaténation de séquences.

Alignements particuliers

Nous définissons ici deux genres particuliers d'alignements qui nous seront utiles dans la suite.

Alignement vertical : un tel alignement est constitué uniquement de remplacements (identités ou substitutions), donnant lieu à une structure de correspondance identique à celles vues au 3.2.4.1). Soulignons qu'au travers de cette notion, nous donnons une évidence concrète du lien de subsomption existant entre modèles de comparaison à structure de correspondance fixe et modèles de comparaison à structure de correspondance variable.

Alignement à l'identique : *c'est un alignement vertical dont tous les appariements sont des identités. Ses séquences origine et destination sont égales.*

Alignement local et « loco-global »

Depuis le début de cette section, nous avons implicitement traité d'alignements *globaux*, c'est-à-dire mettant en correspondance la séquence origine Sorig avec la séquence destination Sdest dans leur totalité. La notion *d'alignement local* existe également, dans laquelle un facteur de Sorig est mis en correspondance avec un facteur de Sdest. La *comparaison locale* de deux séquences X et Y dans le cadre du modèle d'édition (voir 3.2.6.3) consiste alors, dans sa version usuelle, à trouver le(s) meilleur(s) parmi tous les alignements locaux entre X et Y. Ceci signifie trouver le(s) meilleur(s) parmi tous les alignements **globaux** entre F et F' où (F,F') décrit facts(X)×facts(Y).

Nous définissons de même un type d'alignement moins usuel, intermédiaire en quelque sorte entre les deux précédents : l'*alignement loco-global,* dans lequel un facteur de Sorig est mis en correspondance avec [la totalité de] Sdest. Un tel alignements est dit loco-global dans la mesure où il est local du point de vue de Sorig et global du point de vue de Sdest. Dans le cadre de la comparaison loco-globale d'un pattern Pat avec une séquence X (voir 3.2.6.2), on recherche le(s) meilleur(s) parmi tous les alignements loco-globaux entre X et Pat. Ceci signifie rechercher le(s) meilleur(s) parmi tous les alignements **globaux** entre F et Pat, où F décrit facts(X).

3.2.4.2.8. Représentation et paradigmes d'analyse

Il est possible de donner différentes représentations du résultat de l'opération d'alignement de deux séquences. A chaque mode de représentation correspond en fait un paradigme d'analyse et d'évaluation :

- La trace, dont des exemples ont été vus plus haut (cf. par exemple Figure 3-4).
- L'alignement, qui vient d'être présenté (section 3.2.4.2.7).
- La *derivation* (ou *listing*), qui détaille, dans l'ordre, la succession des éditions permettant de passer de la séquence origine à la séquence destination.

Les trois modes ont été donnés ci-dessus en allant du plus pauvre au plus riche : une trace peut correspondre à plusieurs alignements différents, et un alignement peut correspondre à plusieurs dérivations différentes. La représentation de type « alignement » étant de loin la plus usitée [Kruskal 1983], c'est essentiellement sur elle que nous nous concentrerons. Les 3 modes de représentation donnent lieu à trois paradigmes d'évaluation a priori différents. Ainsi dans le cas général il n'y a pas égalité entre distance de trace, distance d'alignement et distance de dérivation ; néanmoins moyennant quelques conditions couramment remplies il y a, au minimum, égalité entre distance de trace et distance d'alignement [Kruskal 1983]. Cette égalité justifie l'usage que nous faisons, dans ce mémoire, des traces comme moyen de représentation graphique clair et intuitif de transformations et d'alignements.

3.2.4.2.9. Formalisme du Dynamic Timewarping

Une formalisation légèrement différente de celle présentée ci-dessus existe également, et est employée notamment en traitement automatique de la parole [Kruskal & Liberman 1983 ; Sakoe & Chiba 1990]. Dans ce dernier domaine, les problèmes tels que la reconnaissance de mots isolés ou connectés exigent une prise en compte des possibles distorsions de l'axe du temps. Ces distorsions sont dues à des variations substantielles qui existent dans la vitesse d'élocution humaine : accélération (compression de l'axe du temps), décélération (expansion). Timewarping (ou encore *elastic matching*) est le nom, donné dans cet autre formalisme, à la comparaison de séquences dans le cadre du modèle d'édition. L'une des différences porte sur les types d'édition / d'appariement standard, qui dans le cas du Timewarping comprennent les types *Compression* (de plusieurs éléments en un) et *Expansion* (d'un élément en plusieurs). Ces deux types, présentant des analogies avec les types Consolidation et Fragmentation introduits plus haut, peuvent aussi être vue en termes d'indels et de remplacements [Kruskal & Sankoff 1983]. Les algorithmes de calcul de [dis]similitude et d'alignement sont, pour le Dynamic Timewarping aussi, basés sur la programmation dynamique (cf. 3.2.6).

3.2.4.3. Modèles sans structure de correspondance explicite

De tels modèles sont notamment mis en œuvre dans les situations où toute séquence (ou facteur) est préalablement (re)décrite, moyennant une opération d'abstraction, uniquement en termes d'attributs-valeurs. Il en est ainsi, par exemple, dans le système ImPact [Ramalho 1997]. Il peut s'agir aussi de modèles se plaçant dans des paradigmes de représentation très particuliers tels que celui d'Horowitz (1995a,b). A titre d'exemple, nous décrirons ici la comparaison dans le cadre des *PACTs* (« Potential musical ACTions ») d'ImPact. Le modèle des PACTs, trouvant son origine dans les travaux de Pachet (1990), a été proposé dans le cadre de la simulation de l'accompagnement à la contrebasse en jazz [Ramalho & Ganascia 1994; Ramalho & Pachet 1994 ; Ramalho, Rolland & Ganascia 1998]. Dans le système IMPACT, les PACTs sont utilisées entre autres pour représenter des fragments au sein d'une « mémoire musicale ». En simplifiant, l'utilisation de cette mémoire peut être décrit comme suit. Le système peut puiser dans la mémoire musicale pour compléter ou préciser les phrases incomplètement spécifiées qu'il est en train d'élaborer, par exemple une phrase dont la succession des hauteurs est spécifiée mais pas le rythme. Les fragments stockés dans la mémoire sont, eux, **totalement spécifiés** (mélodie et rythme), et à la description de la séquence de notes qui y correspond est ajoutée un ensemble hiérarchisé de descriptions plus ou moins abstraites et plus ou moins redondantes : densités temporelle et fréquentielle, registre, « syncopicité » par exemple. La récupération d'un fragment en mémoire pour compléter ou préciser une phrase donnée prend en compte **uniquement ces descriptions abstraites**, et non les descriptions de bas niveau que sont la succession précise des hauteurs ou celle des durées. (Techniquement parlant, cette mémoire est basée sur le principe du raisonnement à partir de cas [Kolodner 1993]). La comparaison de passages mélodiques intervient donc au niveau de la récupération d'une PACT (cas) en mémoire, et se base sur une hiérarchie de descriptions abstraites dont une partie sont d'ailleurs automatiquement déterminées par le système. Du point de vue du modèle de comparaison, une comparaison numérique classique, attribut par attribut, est effectué. La combinatoire intrinsèque à la comparaison d'entités séquentielles, liée à la multiplicité des structures de correspondance possibles, est donc totalement escamotée.

3.2.5. Modèle d'édition : différents sous-modèles et instances

Pour quantifier le modèle et le rendre opérationnel, nous en décrirons ou proposerons dans la suite plusieurs sous-modèles, de raffinement croissant, allant du modèle de Levenshtein avec substitutions (section suivante) au modèle valué multi-descriptions (4.7). Les applications musicales de différentes instances de modèle seront présentées à la suite de chacune d'entre elles.

3.2.5.1. Modèle non valué

Levenshtein (1965) a proposé une fonction de comparaison entre séquences, que nous appelerons *distance de Levenshtein avec substitutions*, égale au nombre minimum de substitutions et d'indels requis pour transformer la première séquence en la seconde. (Ces travaux de Levenshtein constituent le premier usage connu d'une fonction de comparaison prenant en compte les possibilités de délétion et d'insertion).

La traduction de cette distance en terme de distance d'alignement peut se faire de la manière suivante :

- On fixe ETAA = $\text{ETAA}_{\text{standard}}$.
- La valeur d'un alignement est égale au nombre d'appariements le composant qui ne sont pas des identités (c'est-à-dire au nombre cumulé d'indels et de substitutions).

Une fonction de similitude possible s'inscrivant dans le modèle d'édition est le nombre maximum d'appariements identité de tout alignement entre les deux séquences comparées. On peut par exemple montrer (cf. 3.2.6) que la valeur prise par cette fonction de similitude pour les chaînes de caractères 'isomorphe' et 'polymorphe' est égale à 7, correspondant (entre autres) à la trace de la Figure 3-4. On peut aussi prouver que cette trace (resp. cet alignement) est *strictement optimale*, c'est-à-dire est la seule trace (resp. le seul alignement) permettant d'atteindre cette valeur maximale.

Définition 3-2

Nous qualifions de non valué *tout modèle d'édition dans lequel l'évaluation d'une transformation est basée sur un comptage des appariements (ou des éditions).*

Remarque : L'expression *distance de Levenshtein* est souvent employée comme synonyme de distance d'édition. Cette appellation est en fait ambiguë car Levenshtein

(1965) a proposé deux distances différentes pour comparer des séquences, la seconde étant le nombre minimum d'indels requis (pas de substitutions permises).

La Définition 3-2 assure manifestement la symétrie de toute instance de modèle d'édition non valué : en notant Compar la fonction de comparaison correspondante :

$$\{\forall\ (X,Y)\in \Sigma^* x \Sigma^*\}\ \{\text{Compar}(X,Y)= \text{Compar}\ (Y,X)\}$$

3.2.5.1.1. Limitations

S'il peut s'avérer utile dans des contextes applicatifs particuliers, le modèle non valué est bien souvent trop simpliste. En particulier, il favorise les substitutions au détriment des indels, au sens où il est deux fois moins "coûteux" de faire une substitution qu'une délétion plus une insertion :

$$\begin{pmatrix} x \\ y \end{pmatrix} \quad \text{versus} \quad \begin{pmatrix} x \\ - \end{pmatrix} + \begin{pmatrix} - \\ y \end{pmatrix}$$

3.2.5.2. Modèle valué

Au lieu de comptabiliser des nombres d'appariements ou d'éditions, dans le modèle valué chaque type d'appariement se voit associer une fonction numérique et un alignement est évalué par la somme, sur l'ensemble des appariements qui le composent, des valeurs prises par les fonctions associées (cf. Définition 3-3 ci-dessous). Illustrons tout de suite cela :

Exemple 3-4

Nous fixons les fonctions constantes suivantes :

- +4 pour toute identité
- -3 pour toute substitution
- -2 pour toute indel

La valeur de l'alignement de l'Exemple 3-3 est alors égale à +25 (= (-3)+(-2)+4+(-2)+(-2)+4+4+4+4+4+4), dont on peut montrer qu'elle n'est pas maximale. Ceci illustre un fait important : même en gardant le même ensemble ETAA, l'utilisation d'une **fonction d'évaluation d'appariement différente** peut conduire à un ou des **alignement(s) optimal(aux) différents**. †

Le terme que nous emploierons pour désigner ces fonctions est différent suivant que l'on se place dans le cadre d'un modèle de similitude ou dans le cadre d'un modèle de dissimilitude.

- Pour un modèle de similitude : *fonction de contribution* associée à un type d'appariement. La valeur prise par cette fonction pour un appariement particulier a (instance du type associé) sera appelée en raccourci *contribution de a*.
- Pour un modèle de dissimilitude, *fonction de coût* ("poids" est aussi utilisé dans la littérature).

Nous pouvons maintenant donner la définition de la valeur d'un alignement. Nous nous plaçons pour cela dans le cadre d'un modèle de similitude ; la définition homologue dans le cadre d'un modèle de dissimilitude s'obtient en remplaçant "contributions" par "coûts".

Définition 3-3. Valeur d'un alignement dans le cadre d'un modèle de similitude

La valeur *de l'alignement A=a1..ap est égale à la somme des contributions des différents appariements dont il est constitué, soit :*

$$Valeur(A) = \sum_{i=1}^{p} contrib(ai)$$

Remarque : dans la littérature, certains emploient plutôt le terme de longueur (*length*), que nous préférons éviter dans la mesure où il désigne déjà le nombre d'éléments d'une séquence. D'autres parlent de *score*, mais — manque de chance — ce mot qui pourrait avantageusement faire l'affaire est lui aussi susceptible d'engendrer une ambiguïté dans un contexte musical, puisque *score* signifie en anglais partition musicale…

Très fréquemment, les fonctions de contribution (resp. de coût) sont choisies de manière « symétrique », ce qui, en prenant l'exemple du modèle d'édition standard, signifie que :

- La fonction de contribution associée au type d'appariement Remplacement est elle-même symétrique, c'est à dire vérifie, pour tout couple d'éléments (a,b) :

$$contrib\begin{pmatrix} a \\ b \end{pmatrix} = contrib\begin{pmatrix} b \\ a \end{pmatrix}$$

- Les fonctions de contribution associées au types d'appariement Insertion et Délétion vérifient la relation suivante, pour tout élément a :

$$contrib\begin{pmatrix} a \\ - \end{pmatrix} = contrib\begin{pmatrix} - \\ a \end{pmatrix}$$

La **fonction de similitude (resp. de dissimilitude)** correspondant à l'instance de modèle est alors **elle-même symétrique**. C'est notamment le cas des fonctions de similitude employées dans les différentes expérimentations qui seront présentées (sections 4.6 et 5.5).

Une différence nette peut être marquée entre les modèles pour lesquels les fonctions d'évaluation des différents types d'appariement sont constantes, et ceux pour lesquels elles ne le sont pas (on parle par exemple, dans certains cas, de *"position-dependent weights"*) :

3.2.5.2.1. Modèle à coûts ou contributions constants

Subsomption du modèle non valué

Le modèle non valué peut être vu comme un sous-modèle du modèle valué à coûts/contributions fixes : par exemple, pour la distance de Levenshtein avec substitutions il suffit d'attribuer un coût de 1 à chacun des 3 types d'édition Substitution, Insertion, Délétion (« couts unitaires »).

Utilisation musicale

Dans le cadre de la conception d'un système de suivi de partition (*score following system*), Dannenberg et ses collègues ont mis en œuvre un modèle d'édition à contributions constantes [Dannenberg 1984 ; Bloch & Dannenberg 1985]. Dannenberg un modèle d'édition standard, les contributions étant égales respectivement à 1 (identités) et 0 (subsitutions, indels). Le modèle a été repris par Rowe (1993) et implémenté au sein du système musical interactif Cypher.

3.2.5.2.2. Modèle à coûts ou contributions variables

Le modèle d'édition à coûts/contributions constants est couramment utilisé. Dans un certain nombre de contextes en revanche, il est plus adéquat d'utiliser des fonctions de coût/contribution qui, pour un type d'édition/appariement donné, dépendent des caractéristiques du ou des élément(s) impliqués dans chaque instance.

Prenons un exemple simple sur les chaînes de caractères. Il est concevable de baser la contribution de la substitution d'un caractère par un autre sur leur proximité dans l'alphabet. On peut par exemple fixer :

contrib(car1,car2) = 4 - 0.7×(nombre de caractères séparant car1 et car2 dans l'alphabet).

Pour éviter d’obtenir des valeurs trop négatives, nous prendrons en réalité le maximum entre la valeur ci-dessus et -3. Reprenons l'alignement de l'Exemple 3-3 (page 79). Nous avons :

$$contrib\begin{pmatrix} i \\ p \end{pmatrix} = \max(4-(0.7\times 7), -3) = -0.9$$

Les fonctions de contribution restant les mêmes (constantes) que dans l'Exemple 3-4, nous obtenons pour l'alignement une valeur de 21.1.

Utilisation musicale

Stammen & Pennycook (1993) et Mongeau & Sankoff (1990) ont mis en œuvre un modèle d'édition valué à coûts/contributions variables sur des séquences mélodiques. Pour alléger le texte, nous abrégerons dans cette section “Stammen & Pennycook”en « S&P » et « Mongeau & Sankoff » en « M&S ».

Dans les travaux des premiers, la comparaison de séquences intervient au sein d'un algorithme d'extraction en-ligne de patterns mélodiques en jazz tonal [Pennycook, Stammen & Reynolds 1993], cf. 3.4.1.8. La problématique visée par les travaux de M&S est, pour sa part, celle de la comparaison quantitative de monodies tonales.

Remarque : la première et principale partie de l'article de M&S présente un modèle de *dissimilitude*. Une seconde partie présente pour sa part un modèle de similitude dérivé directement du précédent. De ce fait, nous nous contenterons de décrire ici leur modèle de dissimilitude, en particulier les fonctions de coût mises en œuvre. Le modèle de dissimilitude sera vu au travers de l’exemple présenté au 4.5.2.3.

Du point de vue du modèle d'édition, M&S emploient le même formalisme que celui employé dans ce mémoire. Par contre, S&P font appel au formalisme du *dynamic timewarping* (3.2.4.2.9).

L’ensemble des types d’appariements autorisés par M&S est {Insertion, Délétion, Remplacement, Consolidation, Fragmentation} (voir plus haut pour la définition de ces différents types d’appariements). S&P permettent, en dehors des remplacements, les compressions et expansions de longueur 2 au maximum.

Dans les deux cas, deux dimensions sont prises en compte dans le calcul des fonctions de coût :

- Pour S&P, il s'agit pour chaque note :
    - de l'intervalle chromatique la séparant de la note précédente, exprimé en un nombre algébrique de demi-tons;
    - du *ratio de durée*, défini comme le quotient entre la durée de la note et la durée de celle qui la précède (voir aussi 4.7.7.3.2)
- Pour M&S, ce sont, pour chaque note :
    - La durée, exprimée en nombre de double-croches.
    - Le degré (diatonique) dans la tonalité globale du morceau auquel appartient la note.Voyons à présent comment sont définies les **fonctions de coût** :
- En désignant par ICP la description « intervalle chromatique entre chaque note et la précédente » et par RDP la description « ratio de durée entre chaque note et la précédente », S&P définissent le coût d’un remplacement $\begin{pmatrix} n \\ n' \end{pmatrix}$ par la distance euclidienne :

$$\text{coût} = (n.ICP - n'.ICP)^2 + (n.RDP - n'.RDP)^2$$

(Comme il s’agit du formalisme du *dynamic timewarp*, cette fonction de coût suffit à définir celle d’une compression ou expansion).

- Pour M&S, précisons par exemple ce qu’il en est des fonctions de coût associées au type Délétion et au type Remplacement. Soient ai et bj des notes respectives de la 1^e et de la 2^e séquence.

- Le coût du remplacement de ai par bj est la somme de deux termes : le premier se rapporte à la différence entre les degrés respectifs de ai et bj dans la tonalité de leurs morceaux respectifs ; le second se rapporte à la différence de durée entre les deux notes :

$$\text{coût}\begin{pmatrix} ai \\ bj \end{pmatrix} = \text{coûtDeg}\begin{pmatrix} ai \\ bj \end{pmatrix} + \text{k1}\times\text{coûtDur}\begin{pmatrix} ai \\ bj \end{pmatrix}$$

où k1 est un coefficient contrôlant l’emphase relative placée sur le deuxième terme par rapport au premier.

En désignant par Dur la description « durée », $\text{coûtDeg}\begin{pmatrix} ai \\ bj \end{pmatrix}$ se décompose encore en :

$$\text{coûtDeg}\begin{pmatrix} ai \\ bj \end{pmatrix}=\text{min(ai.Dur, bj.Dur)}\times\text{w'i}\begin{pmatrix} ai \\ bj \end{pmatrix},$$

où $\text{w'i}\begin{pmatrix} ai \\ bj \end{pmatrix}$ est un terme censé traduire spécifiquement l’écart tonal entre ai et bj. Son calcul diffère suivant que ai appartiennent ou non aux gammes respectives associées aux tonalités globales des deux morceaux comparés. Si cette appartenance est avérée à la fois pour ai et bj, la valeur de $\text{w'i}\begin{pmatrix} ai \\ bj \end{pmatrix}$ est donnée, en fonction de la différence Δ de degrés entre ai et bj, par le tableau suivant :

D	0	1	2	3	4	5	6
deg(D)	0	0.9	0.2	0.5	0.1	0.35	0.8

Lorsque soit ai, soit bj, soit les deux, n’appartiennent pas aux gammes respectives, $\text{w'i}\begin{pmatrix} ai \\ bj \end{pmatrix}$ fait intervenir des calculs plus compliqués qui ne seront pas détaillés ici.

Pour sa part, coûtDur(ai, bj) est défini tout simplement par :

$$\text{coûtDur}\begin{pmatrix} ai \\ bj \end{pmatrix} = |\text{ai.Dur} - \text{bj.Dur}|$$

- Le coût de la délétion de ai est traité comme un cas particulier de remplacement :

$$\text{coût}\begin{pmatrix} ai \\ - \end{pmatrix} = \text{k1} \times |\text{ai.Dur}|$$

3.2.6. Algorithmes de comparaison

Un modèle de comparaison de séquences ayant été fixé, il faut un algorithme pour calculer effectivement la valeur de la similitude entre deux séquences X et Y, c’est-à-dire calculer l'image de (X,Y) par la fonction de similitude. Pour certains modèles ce calcul est trivial, c'est notamment le cas des modèles à structure de correspondance fixe (section 3.2.4.1) : distance de Hamming par exemple. Pour d'autres modèles en revanche, notamment les modèles d'édition, une optimisation est requise, laquelle nécessite une forme de recherche dans un espace.

Par suite, au plan de la complexité algorithmique le calcul de similitude est plus ou moins coûteux suivant le modèle de comparaison de séquences utilisé. Pour un modèle à structure de correspondance fixe, cette complexité est linéaire, c’est-à-dire proportionnelle à la longueur (égale) des deux séquences comparées. Pour un modèle d'édition, la nécessaire optimisation sur l'espace de tous les alignements possibles rend la complexité, dans le cas général, au mieux quadratique, même en se limitant au type d’appariement (ou d'édition) standard {Insertion, Délétion, Remplacement}. Autrement dit, cette complexité est proportionnelle au produit des longueurs des deux séquences comparées.

Il est trivial de concevoir un algorithme de calcul de similitude lorsque la structure de correspondance est fixe, puisqu'il s'agit d'un simple parcours — par exemple "de gauche à droite" pour Hamming. En fait, parmi les modèles de comparaison présentés plus haut, seul le modèle d'édition sera pris en considération dans la présente section. Le calcul, dans le cadre de ce modèle, de la [dis]similitude entre deux séquences, et en particulier la détermination d'une distance d'édition, ont fait l'objet de très nombreux travaux. Nous distinguons trois situations en fonction du caractère global ou local de l'alignement (ou des alignements) recherché(s).

- La première situation (3.2.6.1) et la troisième situation (3.2.6.3) correspondent à un alignement local et global, respectivement. Ces deux catégories d'alignement ont été distinguées au 3.2.4.2.7.

- La seconde correspond au cas en quelque sorte intermédiaire où l'on recherche des occurrences d'une séquence, appelée *pattern*, dans une autre séquence dite *source*. L'alignement recherché est alors global du point de vue du pattern et local du point de vue de la source.

Pour ne pas allonger démesurément cette section 3.2, les descriptions "techniques" de chacun des trois types d'algorithmes correspondants seront donnés ultérieurement ; les trois sections suivantes s'attacheront à en indiquer des caractéristiques générales. Soulignons que tous ces algorithmes sont basés sur des techniques, analogues entre elles, s'inscrivant dans le cadre de la programmation dynamique [Bellman 1957]. Une discussion sera proposée au 5.6.3 concernant les apports liés à l'utilisation de ces techniques en termes de combinatoire

3.2.6.1. Version globale

Dans le cadre du modèle d'édition, la [dis]similitude — ou la distance — globale entre deux séquences X et X' est calculée comme la valeur optimale de tout alignement (global) entre X et X'. Par exemple la distance de Levenshtein avec substitutions entre les chaînes de caractères "isomorphe" et "polymorphe" est égale à 4, l'un des alignements optimaux étant par exemple le suivant :

$$\begin{bmatrix} i & s & o & - & - & m & o & r & p & h & e \\ p & - & o & l & y & m & o & r & p & h & e \end{bmatrix}$$

Cette famille d'algorithmes réalisant l'alignement global de 2 séquences, nous la désignerons par AG2S. Pour désigner la sous-famille constituée des algorithmes basés sur la programmation dynamique, nous utiliserons l'acronyme AG2SPD. A vrai dire, l'association conceptuelle qui est faite entre modèle d'édition et algorithmes de recherche d'alignements à base de programmation dynamique est si forte qu'abusivement des termes comme « alignement par programmation dynamique » sont par exemple employés là où la désignation exacte serait « comparaison basée sur la distance d'édition ».

Le principe sous-tendant les algorithmes AG2SPD [Needleman & Wunsch 1970 ; Sankoff 1972 ; Sellers 1974 ; Wagner & Fisher 1974 ; etc.] a fait l'objet d'une singulière "redondance" scientifique, ayant été découvert et publié de manière indépendante dans plusieurs pays et au sein de plusieurs disciplines [Kruskal 1983, p. 23]. Il est présenté brièvement ici, et sera détaillé dans le cadre de la présentation de FlExPat (voir 5.3), qui met en œuvre ce principe.

Soient X et X' deux séquences à comparer, de longueurs respectives n et n'. Supposons par exemple que ce que l'on cherche à calculer est une distance entre X et X', c'est-à-dire leur **distance d'édition**. La distance recherchée, notée DISTn,n' sera calculée à partir des distances entre certains **préfixes** de X et de X', distances calculées auparavant (voir 5.3.4.2 et 5.3.5). Chacune de ces distances, soit DISTn-k,n'-k' avec (k,k')>(0,0), aura elle-même été calculée à partir de distances, calculées encore auparavant, entre préfixes de la forme DISTn-k-l,n'-k'-l' avec (l,l')>(0,0). L'ensemble précis de préfixes pris en compte dépend (principalement) de l'ensemble des types d'éditions ou d'appariements autorisées.

Au total, l'algorithme de calcul de [dis]similitude ou de distance entre peut se traduire par le remplissage progressif et complet d'une matrice COMPAR_G de dimension (n+1)×(n'+1). (Voir Figure 4-10, page 137, pour un exemple d'une telle matrice ; il s'agit en l'occurrence de comparaison **locale**, comme décrit ci-après). COMPAR_Gi,j est la [dis]similitude ou la distance entre X[[1,i]] et X'[[1,j]], le nombre recherché étant COMPAR_Gn,n'. Le remplissage débute par le coin "supérieur gauche" (0,0) de la matrice et s'achève par le coin "inférieur droit" (n,n').

La complexité en temps de cet algorithme est **O(n,n')**, si toutefois l'ensemble des signatures des appariements autorisés est à cardinal constant (cf. 5.3.5), Ce cas de figure est notamment rencontré pour le modèle d'édition standard où ce cardinal est égal à 3 (ensemble des signatures : {(0,1), (1,0), (1,1)}). La complexité en espace est O(min(n,m)) [Myers & Miller 1988 ; Sagot 1996].

Indiquons que la formalisation est légèrement différente dans le cadre du Dynamic TimeWarping (cf. 3.2.4.2.9).

Un certain nombre de travaux ont porté sur la réduction de la complexité algorithmique. Les moyens mis en œuvre sont soit de limiter la nombre d'éléments de la matrice COMPAR_G effectivement calculés [Kruskal & Sankoff 1983 ; Spouge 1989 ; Ukkonen 1985], soit de faire des hypothèses particulières sur la forme des fonctions de coût/contribution [Masek & Paterson 1983].

3.2.6.2. Version « loco –globale »

La définition standard de ce problème de comparaison ("fitting one sequence into another" [Waterman 1995]) est la suivante :

Définition 3-4

Etant donnés une séquence source X sur un alphabet Σ et un pattern Pat qui est un mot également sur Σ :

La comparaison loco-globale *de Pat et X consiste à trouver le(s) facteur(s) F de X maximisant Simil(Pat,F), où Simil(Pat,F) est la valeur de la similitude entre Pat et F au sens de l'alignement* ***global*** *entre ces deux mots.*

La Figure 3-7 illustre cette définition.

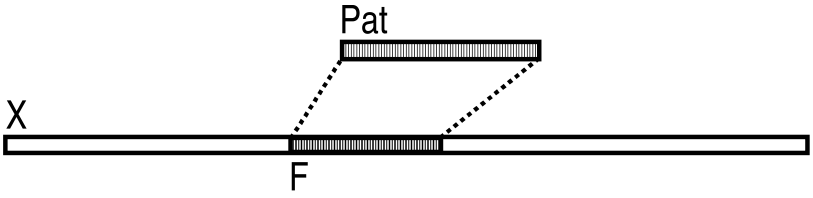


Figure 3-7. Comparaison loco-globale entre un mot Pat et une séquence X

Il s'agit d'une forme particulière de recherche d'occurrence de pattern (cf. 3.3). Ce qui la caractérise est que le pattern est un **mot** sur le même alphabet que la/les séquence(s) source(s), et que l'on recherche des occurrences approximatives. Les algorithmes [Sellers 1980 ; Erickson & Sellers 1983 ; Waterman 1995, pp. 201-202 ; Crochemore & Rytter 1994, pp. 244-247] seront détaillés dans la section correspondante (3.3.2). La complexité temporelle obtenue est proportionnelle au **produit des longueurs** de Pat et X, dans le cas où $\mathrm{ETAA}=\mathrm{ETAA}_{\mathrm{standard}}$.

3.2.6.3. Version locale

Définition 3-5

Etant données deux séquences X et X', la comparaison locale *de X et X' consiste à trouver le(s) couple(s) de facteur(s) (F,F')* ∈ $\underline{\underline{facts}}(X)\times\underline{\underline{facts}}(X')$ *maximisant Simil(F,F'). Ici encore, Simil(F,F') est la valeur de la similitude entre F et F' au sens de l'alignement* ***global*** *entre ces deux facteurs.*

La Figure 3-8 illustre cette définition.

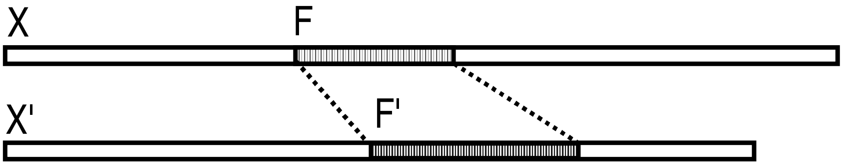


Figure 3-8. Comparaison locale entre deux séquences X et X'

Nous désignerons par AL2SPD la famille d'algorithmes réalisant l'alignement local de 2 séquences par programmation dynamique. Selon Waterman (1995, p. 203), il s'agit là des algorithmes de programmation dynamique "les plus utiles aujourd'hui en biologie moléculaire". Les algorithmes [Smith & Waterman 1981 ; Kruskal & Sankoff 1983] seront détaillés au 4.5.2. Contentons-nous d'en préciser ici la complexité temporelle : celle-ci est ici encore proportionnelle au **produit des longueurs** des deux séquences comparées, dans le cas où $\text{ETAA}=\text{ETAA}_{\text{standard}}$.

3.3. Algorithmes de localisation

Parmi les critères possibles de classement, nous choisissons de classer les algorithmes de localisation selon les critères d'existence d'occurrences, et en particulier le modèle explicite ou implicite de comparaison de séquences, qui sous-tend chacun d'entre eux. Aussi opérons-nous une distinction majeure entre localisation d'occurrences exactes et localisation d'occurrences appoximatives (cf. Définition 2-5). Schématiquement :

- Un grand nombre d'algorithmes ont été développés pour le cas exact, atteignant de performances très bonnes ou même optimales en termes :
  - de complexité en temps (par exemple [Boyer & Moore 1977 ; Knuth, Morris & Pratt 1977]), voire :
  - de complexités en temps et en espace (par exemple [Crochemore & Perrin 1991 ; Galil & Seiferas 1983]).
- Pour une bonne partie des algorithmes développés pour le cas approximatif, les performances en termes de complexité sont nettement moins bonnes que pour le cas exact, en raison d'une combinatoire accrue.

Cela ne veut pas dire que l'intersection entre ces deux familles d'algorithmes est vide. Il existe notamment des algorithmes initialement conçus pour la localisation d'occurrences exactes qui ont été adaptés pour pouvoir traiter la localisation d'occurrences approximatives.

Sauf lorsque nous le signalerons explicitement, les patterns dont on recherche une ou plusieurs occurrences sont des mots (sur le même alphabet que les séquences sources). C'est le cas pour la grande majorité des algorithmes présentés ci-après.

3.3.1. Localisation d'occurrences exactes

Dans la présentation que nous faisons ci-dessous des différents algorithmes, en l'absence de précisions particulières de notre part le problème à résoudre est supposément celui du test d'occurrence défini au 2.2.2. Le passage au problème de la localisation proprement dite (Définition 2-1) est souvent immédiat. De même, sauf spécification particulière il est supposé que la localisation s'effectue dans une seule séquence et non dans plusieurs, une remarque analogue à la précédente pouvant là aussi être faite.

Naturellement, comme il s'agit d'occurrences exactes tous les algorithmes de la présente section recherchent des occurrences de patterns qui sont eux-mêmes des mots. Toutefois certains de ces algorithmes peuvent permettre — moyennant d'éventuelles adaptations — la recherche d'occurrences de patterns d'une autre nature. Par exemple, les algorithmes à base d'automates permettent de rechercher des occurrences de patterns qui sont [définis par] des expressions régulières.

L'algorithme élémentaire que nous présentons en premier lieu a une complexité en temps *quadratique* dans le pire des cas. Les "bons" algorithmes présentés dans les sections ultérieures ont une complexité au pire *linéaire*, la taille du problème étant typiquement définie comme la longueur cumulée du pattern et de la séquence.

Pour des raisons pratiques, nous supposons que :

- Le pattern et la séquence sont stockés dans des variables, respectivement Pat et X, considérées comme globales par rapport aux différentes fonctions et procédures que nous écrirons.
- Leurs longueurs respectives sont elles aussi stockées dans des variables globales m et n, respectivement.

3.3.1.1. Algorithme élémentaire

Un algorithme tout à fait immédiat peut être donné, qui consiste à parcourir la X de gauche à droite jusqu'à ce que :

- soit une occurrence de Pat soit trouvée,
- soit on ait dépassé le dernier rang possible pour une occurrence de Pat, c'est-à-dire n - m + 1.

```
Fonction testOccurrence {"algorithme PatMatchElém"}
    Type renvoyé : booléen
    Variables locales : i et k, deux entiers
    Début
        i ← 1
        TantQue i ≤ n - m + 1 Faire Début
            {"Vérifier s'il y a une occurrence au rang i "}
            k ← 1
            TantQue k≤m et Pat[k]=X[i+k-1] Faire k ← k+1
            Si k=m+1 alors Renvoyer VRAI
            i←i+1
        Fin
        Renvoyer FAUX
    Fin
```

La complexité en temps, mesurée en nombre de comparaison de symboles ('Pat[k]=X[i+k-1]'), est quadratique au pire [Crochemore & Rytter 1994].

3.3.1.2. Algorithmes de Knuth-Morris-Pratt, de Boyer-Moore et variantes

L'algorithme de Knuth, Morris & Pratt et ses variantes [Morris & Pratt 1970 ; Knuth, Morris & Pratt 1977] peut être vu comme une amélioration de l'algorithme élémentaire du 3.3.1.1. Donnons le principe de cet algorithme, abrégé ici en KMP.

Dans PatMatchElém, à chaque nouvel échec il y a décalage d'1 position vers la droite ('i←i+1'). Dans KMP on effectue au contraire un "saut" vers la droite de longueur variable. Par rapport à l'algorithme élémentaire, cela permet d'éviter un nombre potentiellement très grand de comparaisons de symboles inutiles. Cela résulte en une réduction importante de la complexité au pire, qui devient linéaire.

Voyons comment est déterminé la taille du saut au moment où un échec se produit. Un échec est caractérisé par :

$$Pat[[1,j-1]]=X[[i,j-1]]$$

et :

$$Pat[j] \neq X[i+j-1]$$

Un saut de longueur s est dit *acceptable* si et seulement si :

- On est certain qu'il n'y a d'occurrence de Pat à aucun des rangs i, i+1, … i+s-1
- Il peut y avoir une occurrence de Pat au rang i+s

Sans détailler comment on détermine une telle valeur s dans KMP, indiquons simplement simplement que cela fait intervenir une fonction auxiliaire B définie comme suit : B[j] est la longueur de la *bordure* de Pat[1..j], préfixe de longueur j de Pat (cf. 2.2.1.2.5).

L'algorithme de Boyer & Moore (1977), abrégé en BM, peut lui-aussi être vu comme une amélioration d'un algorithme élémentaire, appelé PatMatchElémBis, légèrement différent de PatMatchElém (3.3.1.1). La différence tient au fait suivant : Dans PatMatchElém, le test d'égalité entre Pat et X[[i,m] s'effectue par un parcours de gauche à droite : 'k←1' puis 'k ← k+1'. On privilégie ainsi la prise en compte des **préfixes** de Pat. Dans PatMatchElémBis au contraire, ce test d'égalité s'effectue de droite à gauche : 'k←m' puis 'k ← k-1', privilégiant la prise en compte des **suffixes** de Pat.

Dans BM comme dans KMP, l'amélioration sensible de l'efficacité tient aux sauts qui sont effectués et qui réduisent le nombre de comparaison de symboles. Dans sa version de base, BM effectue au plus 4n comparaisons de symboles, indépendamment de la taille de l'alphabet. La fonction B (bordures) y intervient également pour le calcul de la taille des différents sauts. L'algorithme initial de Boyer & Moore a fait l'objet d'un certain nombre d'améliorations publiées dans la littérature, par exemple par Galil (1979) ou plus récemment par Crochemore et al. (1992).

3.3.1.3. Automates

3.3.1.3.1. Définitions de base

Un automate fini déterministe (cf. par exemple [Aho, Hopcroft & Ullman 1974]) est un 5-uplet $A=\{\Sigma, Q, q_0, \delta, T\}$ où :

- Σ est l'alphabet sur lequel on se place

- Q est un ensemble (fini) d'*états*
- $q_0 \in Q$ est l'*état initial*.
- δ est la *fonction de transition* : si $q \in Q$ et $\sigma \in \Sigma$, $\delta(q,\sigma)$ est l'état atteint depuis l'état q par la transition étiquetée par σ ("*lecture*" du symbole σ). En étendant δ à l'ensemble des mots sur Σ, on obtient :

$$\delta : Q\times\Sigma^* \to Q$$
$$(q,X) \mapsto q'$$

  où q' est l'état atteint depuis q après lecture de X.

- T est l'ensemble des états terminaux de l'automate.

δ peut être une fonction partielle. L'automate A *accepte le langage* L(A) défini par :

$$L(A)=\{X\in \Sigma^*, \delta(q_0,X) \text{ est défini et est dans } T\}$$

3.3.1.3.2. Utilisation pour la localisation de pattern

Le concept d'automate peut être appliqué au problème de test d'occurrence exacte pour un pattern qui est un mot et, au-delà, au problème de la localisation d'un tel pattern.

Etant donné un pattern $Pat \in \Sigma^*$, pour effectuer le test d'occurrence on construit un automate fini déterministe noté A(Pat) qui accepte le langage formé des mots admettant Pat comme suffixe :

$$L(A(Pat)) = A^*Pat$$

Notons $m = |Pat|$. Un algorithme permet de construire un tel automate A(Pat) en un temps **$O(|\Sigma|.m)$**. De plus cet algorithme permet de minimiser le nombre d'états de A(Pat) : m+1 états (et $|\Sigma|.(m+1)$ transitions).

Une fois A(Pat) construit, un algorithme **en temps-réel** permet de localiser toutes les occurrences de Pat dans une séquence X sur Σ. X est parcouru élément par élément par ordre croissant d'indices, et à chaque indice i l'algorithme renvoie un booléen qui vaut VRAI s'il y a une occurrence de Pat de rang terminal i dans X, et FAUX sinon. A fortiori, la complexité du test d'occurrence est linéaire par rapport à la taille de X.

3.3.1.3.3. Qualités spécifiques

Par rapport aux approches présentées précédemment, deux avantages spécifiques des approches à base d'automates sont de permettre la recherche d'occurrences :

- de plusieurs patterns simultanément (cf. par exemple, [Aho & Corasick 1975 ; Crochemore et al. 1993]).
- de patterns qui peuvent être des expressions régulières et non plus simplement des mots.

3.3.1.4. Arbres et automates des suffixes

Les *arbres des suffixes* [McCreight 1976 ; Ukkonen 1992] et les *automates des suffixes* ou *Directed Acyclic Word Graphs* (DAWGs) [Blumer et al. 1983 ; Crochemore 1985 ; 1986] sont des structures de données représentant de manière adéquate tous les facteurs d'une séquence donnée. Outre la recherche d'occurrences de patterns, les applications de ces structures de données incluent la recherche de facteurs répétés, cf. 3.4.1.2.

Arbres de suffixe (ADSs) et DAWGs présentant de nombreux points communs, nous parlerons ici essentiellement des premiers, en considérant (uniquement) le cas d'une séquence unique X. Un ADS pour X est un arbre orienté possédant exactement L=|X| feuilles numérotées de 1 à L. Chaque nœud interne autre que la racine a au moins deux fils, et chaque branche est étiquetée par un mot non vide de X. Les étiquettes de deux branches descendant d'un même nœud diffèrent nécessairement au niveau de leur premier symbole.

La propriété essentielle d'un ADS est la suivante. Pour tout i, la concaténation des étiquettes des branches composant le chemin allant de la racine à la feuille N° i est égale au suffixe de longueur (L-i+1) de X, c’est-à-dire X[i .. L].

Un ADS (ou un DAWG) pour une séquence de longueur L peut être construit en un temps linéaire par rapport à L et occupe un espace également linéaire. Moyennement un prétraitement de complexité linéaire, l'arbre des suffixes d'une séquence X permet d'effectuer le test d'occurrence d'un mot Pat sur Σ en un temps **O(|Pat|)**. Il permet également, avec la même complexité, de déterminer par exemple le nombre d'occurrences de Pat dans X [Crochemore & Rytter 1994].

Les meilleurs algorithmes présentés ci-dessus (en particulier KMP et BM) permettaient, après un prétraitement sur Pat, d'effectuer le test d'occurrence de Pat dans

X en un temps linéaire par rapport à |X|. Le passage, permis par un ADS ou un DAWG, à une complexité linéaire par rapport à |Pat| représente une amélioration très significative par rapport aux algorithmes précédents dans tout contexte où X est fixe.

3.3.1.5. Autres

En dehors des catégories principales présentées ci-dessus, un certain nombre d'autres approches ont été proposées. Peuvent par exemple être citées (cf. [Crochemore & Rytter 1994]) :

- les approches par duels [Vishkin 1985], qui présentent des liens avec KMP, ou encore :
- les approches par échantillonage [Vishkin 1991].

3.3.2. Localisation d'occurrences approximatives

Nous nous concentrons ici sur les algorithmes où le modèle de comparaison est le modèle d'édition (valué). Nous supposerons, sans perte de généralité, qu'il s'agit d'un modèle de *similitude*. Pour toute paire de séquences (X,X'), Simil(X,X') est donc la valeur maximale de tous les alignements (globaux) entre X et X' (cf. 3.2.4.2).

Les algorithmes proposés dans la littérature [Sellers 1980 ; Erickson & Sellers 1983] servent à résoudre un problème légèrement différent de celui de la localisation tel que nous l'avons défini. Cet autre problème correspond à la Définition 3-4 (page 93) : recherche du/des facteur(s) F de X maximisant Simil(Pat,F).

3.3.2.1. Approche naïve

Supposons que, pour le résoudre, on énumère tous les facteurs[12] F={i',m'} de X en comparant chacun à Pat l'aide d'un algorithme classique de comparaison globale de deux séquences par programmation dynamique (AG2SPD, cf. 3.2.6.1). Plaçons-nous dans le cas standard ETAA=ETAA$_{\text{standard}}$, et posons n=|X|. Nous avons vu que la complexité en temps d'AG2SPD est alors égale au produit des longueurs des deux séquences comparées soit ici m.m'. On obtient au total une complexité en temps égale à :

$$1+\sum_{i'=1}^{n}\sum_{m'=1}^{n+1-i'} m.m',$$

---

[12] Y compris le facteur vide.

grandeur qui est en O($m.n^3$). (Le terme "1" avant la double sommation entière correspond au facteur vide). En vue de réduire cette complexité, cubique en considérant (classiquement) la taille du problème comme étant n, on met en oeuvre une autre approche, dérivée de celle d'AG2SPD.

3.3.2.2. Approche basée sur la programmation dynamique

Cette approche [Sellers 1980 ; Erickson & Sellers 1983 ; Waterman 1995, pp. 201-202 ; Crochemore & Rytter 1994 pp. 244-247] permet de ramener la complexité en temps à **O(m.n)**. Elle passe par la construction, explicite ou implicite, d'une matrice de similitude "loco-globale" que nous notons SIMIL_LG. Cette matrice possède m+1 lignes et n+1 colonnes, numérotées respectivement de 0 à m et de 0 à n.

On définit SIMIL_LGi,j comme étant la valeur maximale de la similitude entre le préfixe de longueur i de Pat et tout facteur de X de rang terminal j.

Autrement dit :

$$\text{SIMIL\_LGi,j} = \max\{\text{Simil}(\text{Pat}[1..i],X[k..j]),\ 1 \leq k \leq j\}$$

On démontre alors la propriété suivante :

Propriété 3-1

La valeur maximale de Sim(Pat,F) où F décrit facts(X) est donnée par :

$$\max\{\text{SIMIL\_LGm,j},\ 1 \leq j \leq n\}$$

Autrement dit, pour obtenir la valeur recherchée il suffit de trouver le(s) plus grand(s) élément(s) de la dernière ligne de SIMIL_LG.

Précisons à présent comment calculer SIMIL_LG. On l'initialise comme suit (rappelons que par hypothèse ici ETAA={Insertion, Délétion, Remplacement}) :

- Initialisation de la ligne 0 :

$$\{\forall j \in \{0,\ldots,n\}\}\ \{\text{SIMIL\_LG0,j} = 0\}$$

- Initialisation de la colonne 0 :

{∀j ∈ {0,…,n}} $\{ SIMIL\_LGi,0 = \sum_{k=1}^{i} Contrib\begin{pmatrix} Pat[k] \\ - \end{pmatrix} \}$

Puis on fait appel à la propriété de récurrence suivante :

Propriété 3-2

Pour tout (i,j) ∈ **N**m×**N**n on a :

$$SIMIL\_LGi,j = \max\begin{cases} contrib\begin{pmatrix} Pat[i] \\ X[j] \end{pmatrix} & + \quad SIMIL\_LGi-1,j-1 \\ contrib\begin{pmatrix} Pat[i] \\ - \end{pmatrix} & + \quad SIMIL\_LGi-1,j \\ contrib\begin{pmatrix} - \\ X[j] \end{pmatrix} & + \quad SIMIL\_LGi,j-1 \end{cases}$$

Equation 3-3

De cette façon SIMIL_LG peut être remplie progressivement en partant du coin "supérieur gauche" 0,0 et en arrivant au coin "inférieur droit" m,n.

Nous venons d'indiquer comment obtenir la valeur maximale de Sim(Pat,F) où F décrit facts(X). Cependant la méthode ne fournit pas directement le(s) facteur(s) accomplissant la maximisation. Pour obtenir ce(s) facteur(s), une méthode consiste à faire usage d'une matrice annexe calculés en même temps que SIMIL_LG et dont les éléments (pointeurs) mémorisent le numéro de la ou des ligne(s) ayant réalisé la maximisation dans l'Equation 3-3. Cette méthode sera détaillée au 5.5 dans le cadre de la comparaison (alignement) locale, sa mise en œuvre variant très peu qu'il s'agisse de comparaison locale, globale ou loco-globale .

3.4. Algorithmes d'extraction

Un certain nombre d'approches et d'algorithmes pour l'extraction de patterns ont été proposés dans la littérature. Nous qualifions un certain nombre de ces algorithmes de *généraux*, au sens où — tant au plan de leurs caractéristiques que de la formalisation qui en est donnée — ils ne sont pas a priori restreints à un domaine d'application particulier. D'autres en revanche s'inscrivent dans un contexte d'application précis ; parmi eux, les quelques algorithmes proposés dans un contexte musical seront présentés dans une section unique (3.4.1.8).

En ce qui concerne les algorithmes généraux, pour une bonne partie d'entre eux les applications qui en ont été faites se situent en biologie. Aussi la section les regroupant est-elle intitulée "Algorithmes généraux ou biologiques".

Remarque : un autre critère général pourrait être une distinction sur la base de la nature des patterns extraits. En particulier, les approches (les plus nombreuses) où les représentants de patterns sont des facteurs des séquences du corpus se distinguent de celles où ces représentants peuvent être des mots n'appartenant pas au corpus voire d'autres structures plus abstraites [Wang et al. 1994 ; Sagot 1996 ; Favory 1995].

3.4.1. Algorithmes généraux ou biologiques

De façon analogue au 3.3, nous choisissons d'ordonner la présentation des différentes approches en fonction du modèle de comparaison d'entités séquentielles qui les sous-tendent. Ainsi les approches exigeant une structure de correspondance fixe pour la comparaison seront présentées d'abord, celles autorisant une structure de correspondance variable ensuite.

Parmi les algorithmes présentés, celui de Karp, Miller & Rosenberg (1972) fera l'objet d'une description plus détaillée au 3.4.1.1 pour deux raisons :

- Premièrement, cet algorithme est — directement ou indirectement — à la base de plusieurs autres algorithmes d'extraction de patterns qui seront présentés dans la suite.

- Deuxièmement, nous l'avons appliqué au domaine musical en l'implémentant dans notre système Imprology.

Il faut souligner que nous nous centrons ici sur les approches combinatoires exactes, par opposition aux approches heuristiques. Un certain nombre de ces dernières ont été proposées, que l'on peut répartir en deux groupes :

- Celles basées sur une optimisation stochastique (par exemple [Lawrence et al. 1993 ; Lawrence & Reilly 1990])

- Celles basées sur une optimisation de fonction de similitude (par exemple [Schuler, Altschul & Lipman 1991])

Un passage en revue détaillé de ces approches, qui sortirait du cadre du présent mémoire, est proposé dans [Sagot 1996].

3.4.1.1. Approches par étiquetage

Plaçons-nous dans l'hypothèse d'un corpus unitaire, confondu avec son unique séquence S. Karp, Miller & Rosenberg (1972) ont proposé un algorithme, que nous nommerons KMR, permettant — entre autres — de trouver tous les facteurs répétés dans S. Cet algorithme est basé une technique d'étiquetage de tous les facteurs de longueur k de S, pour un ensemble de valeurs de k qui sera explicité dans la suite.

La première idée centrale de KMR est la suivante : les blocs de facteurs répétés de longueur k peuvent être obtenus simplement à partir des blocs de longueur k' où k' est un entier quelconque de [k/2 ; k-1]. Cela se justifie par le lemme suivant, illustré par la Figure 3-9. Etant donnés deux facteurs F et F' de longueur k, ce lemme donne une condition nécessaire et suffisante d'égalité de F et F' portant sur des préfixes et suffixes de longueur k' de F et F' :

Lemme 3-1

Soient :

- *X une séquence,*
- *F et F' deux facteurs de même longueur k et de rangs respectifs i et i'*
- *k' un entier de [k/2 ; k-1].*

Alors F = F' si et seulement si

- *leurs préfixes respectifs de longueur k' sont égaux, et :*
- *leurs suffixes respectifs de longueur k' sont égaux.*

En d'autres termes, en posant d = k-k' :

$$F=F'$$

$$\Leftrightarrow$$

$$\{X[[i,k']]=X[[i',k']]\} \text{ ET } \{X[[i+d,k']]=X[[i'+d,k']]\}$$

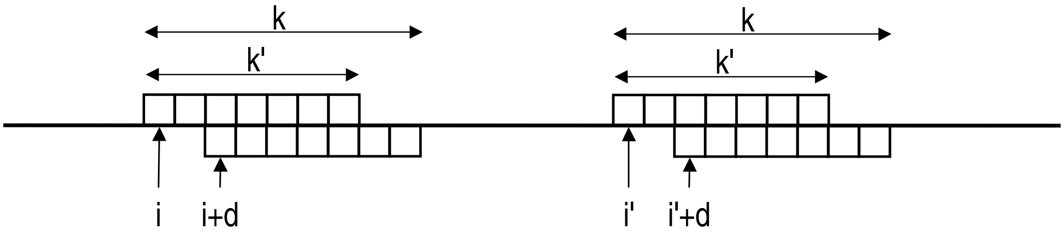


Figure 3-9. Illustration du Lemme 3-1 (KMR) avec k=9 et k'=7.

Un cas particulier, important parce qu'il représente une situation "optimale", est celui où $k \in 2\mathbf{Z}$ et k'=k/2. On a alors p=k' et on peut écrire :

$$F=F'$$

$$\Leftrightarrow$$

$$\{X[[i,k']]=X[[i',k']]\} \text{ ET } \{X[[i+k',k']]=X[[i'+k',k']]\}$$

La seconde idée centrale de KMR est, pour une longueur donnée k, de grouper les facteurs égaux (mais de coordonnées différentes) en classes d'équivalence. Celles-ci sont numérotées — chaque numéro étant vu comme *l'étiquette* de la classe — et pour tout k de {1,…,L} on peut alors "réécrire" la liste des (L-k+1) facteurs de longueur k comme la liste de leurs étiquettes.

Concrètement, soit $\{F_1,\dots F_{f(k)}\}$ l'ensemble des facteurs **distincts** de longueur k de X avec $f(k) \le (L-k+1)$. On associe à chacun une étiquette distincte Etiq(Fj) qui pour des raisons pratiques est un entier de {1,…,f(k)}.

On peut alors associer à chaque indice $i \in \{1,\dots L-k+1\}$ une étiquette notée $NUM_k(i)$, qui est celle du facteur de rang i et de longueur k. $NUM_k(i)$ est donc égal à Etiq(X[[i,k]]). Sur l'ensemble des indices i, on obtient donc un tableau NUM_k de dimension L-k+1.

Exemple 3-5

Soient X=ababab et k=4. On a alors f(4)=2 et l'ensemble des facteurs distincts de X est {F1,F2} avec F_1=abab, F_2=baba. En associant les étiquettes $Etiq(F_1)=1$ et $Etiq(F_2)=2$, on obtient $NUM_4=\{1,2,1\}$. †

Les blocs de facteurs répétés de longueur k s'obtiennent trivialement à partir de NUM_k : chaque bloc est formé de facteurs dont les indices de début (rangs) ont même étiquette. Il suffit donc de chercher dans NUM_k les ensembles d'éléments égaux. Ainsi,

le problème de la recherche de blocs de facteurs répétés de longueur k se ramène à celui de la construction de NUM_k suivi de la recherche des éléments répétés dans ce tableau.

Exemple

Reprenons les données de l'Exemple 3-5. Compte tenu de la valeur de NUM4, nous pouvons dire qu'il n'existe qu'un seul bloc de facteurs répétés de longueur 4, traduit par la présence multiple de l'étiquette 1 dans NUM4. Les indices où elle apparaît nous fournissent les rangs des différents facteurs du blocs, en l'occurrence 1 et 3. †

Il nous faut maintenant préciser le fonctionnement effectif de KMR. Celui-ci repose sur le fait qu'en vertu du Lemme 3-1, un quelconque NUM_k peut se calculer à partir d'un $NUM_{k'}$ où $k' \in [k/2 ; k-1]$. En effet ce lemme permet d'écrire, pour tout couple d'indices (i,i') :

$$\{NUM_k(i) = NUM_k(i')\}$$
$$\Leftrightarrow$$
$$\{NUM_{k'}(i) = NUM_{k'}(i')\} \text{ ET } \{NUM_{k'}(i+d) = NUM_{k'}(i'+d)\}$$

Plaçons-nous dans le cas où $k \in 2\mathbf{Z}$ et $k'=k/2$. On a alors :

$$\{NUM_k(i) = NUM_k(i')\}$$
$$\Leftrightarrow$$
$$\{NUM_{k'}(i) = NUM_{k'}(i')\} \text{ ET } \{NUM_{k'}(i+k') = NUM_{k'}(i'+k')\}$$

Equation 3-4

Supposons que NUMk' ait déjà été calculé et construisons le tableau de couples d'entiers $Yk=\{(NUM_{k'}(i), NUM_{k'}(i+k'))\}_{i\in\{1,\ldots,L-k+1\}}$, ce qui requiert un temps O(L-k+1). Compte tenu de l'Equation 3-4, les couples d'entiers Yk[i] et Yk[i'] sont égaux si et seulement si $NUM_k(i) = NUM_k(i')$ — ce qui signifie que les facteurs X[[i,k]] et X[[i',k]] sont égaux.

Cette construction fournit donc un cheminement algorithmique pour construire NUM_k à partir de $Num_{k/2}$. Nous ne le détaillons pas, mais précisons simplement que cela fait intervenir le tri lexicographique d'un tableau à L-k+1 éléments, lequel peut être effectué en un temps O(L-k+1) [Aho, Hopcroft & Ullman 1983]. Plus généralement, il est possible via un cheminement analogue de construire en un temps O(L-k+1) le tableau NUM_k à partir d'un $NUM_{k'}$ où :

- k n'est pas nécessairement pair
- k' est un entier quelconque de [k/2 ; k-1]

Problèmes

Les principes de KMR ayant été présentés, voyons maintenant différents problèmes que cet algorithme permet de résoudre.

➊ Le premier consiste à rechercher le(s) bloc(s) de facteurs répétés d'une longueur K fixée au départ, s'il(s) existe(nt). Le schéma algorithmique est le suivant :

1. Calculer NUM_1, puis NUM_2 (à partir de NUM_1), puis NUM_4 (à partir de NUM_1), …puis NUM_j, où j est la plus grande puissance de 2 inférieure ou égale à K.
2. Si K=j, c'est terminé. Sinon, calculer NUM_K à partir de NUM_j.

Remarque : en réalité, à l'instant où NUM_{2p} — avec 2p<j — vient d'être calculé, si ce tableau ne contient aucun élément répété, l'algorithme peut être stoppé à ce stade. En effet cela permet de conclure qu'il n'existe aucun bloc de facteurs répétés de longueur 2p, en particulier il n'en existe aucun de longueur Kmax.

Le temps total requis est en O(L.ln Kmax) au pire.

➋ Un second problème consiste à rechercher le(s) bloc(s) des ***plus longs*** facteurs répétés. Pour cela, le schéma algorithmique est le suivant :

3. Calculer NUM_1, NUM_2, NUM_4, … NUM_j où j est la plus grande puissance de 2 pour laquelle il existe au moins un facteur répété de longueur j.
4. Rechercher par dichotomie les plus longs facteurs répétés. Pour cela, calculer tout d'abord $NUM_{3/2j}$ à partir de NUMj :
    - Si $NUM_{3/2j}$ met en évidence au moins un facteur répété de longueur (3/2)j , alors calculer $NUM_{7/4j}$ à partir de $NUM_{3/2j}$, etc.
    - Si au contraire il n'existe aucun facteur répété de longueur (3/2)j , alors calculer $NUM_{5/4j}$ à partir de $NUM_{j,}$ etc.

❸ Un troisième problème, non standard celui-là, consiste à rechercher, lorsqu'il en existe, le(s) bloc(s) de facteurs répétés de longueur inférieure ou égale à une longueur K fixée au départ. On calcule pour cela dans l'ordre NUM_1, NUM_2, NUM_3, … , NUM_K en s'arrêtant éventuellement en cours de route si pour une certaine longueur k aucun bloc n'existe (cf. remarque ci-dessus). La complexité en temps est dans le pire des cas de l'ordre de L + L-1 + … + L-K+1 soit $O(K\times(L - K/2))$. En particulier, pour la valeur maximale K=L-1 on obtient $O(L^2)$.

Extension au cas d'un corpus multiple

Landraud, Avril & Chretienne (1989) ont généralisé KMR au cas d'un corpus multiple C avec contrainte de quorum. Dans un premier temps, l'algorithme concatène C, puis il recherche les mots répétés dans la séquence globale S obtenue. La contrainte dans cette recherche est que les mots qui sont à cheval sur deux séquences différentes ne soient pas pris en compte. A cette fin, une procédure de concaténation particulière est mise en œuvre : un symbole spécial, ne faisant pas partie de l'alphabet, est intercalé entre chaque séquence et la suivante. La complexité totale de l'algorithme est :

- O(L.ln k) en temps, où k est la longueur des facteurs communs recherchés.
- O(L) en espace.

3.4.1.2. Arbres et automates des suffixes

Ces structures de données ont été introduites au 3.3.1.4.

Soit ADS un arbre des suffixes, correspondant à une séquence X. Rappelons qu'un tel arbre peut être construit en un temps O(|X|). Les mots répétés dans X peuvent être obtenus par la concaténation des étiquettes des chemins dans ADS allant de la racine à un nœud interne. Pour localiser les blocs de facteurs répétés dans X, il suffit donc de parcourir ADS en profondeur. Par ailleurs la cardinalité d'un bloc est donnée par le nombre de feuilles descendant du nœud interne correspondant.

Dans le cas d'un corpus C={S1,…, SN} de longueur cumulée L, un ADS peut être obtenu de manière analogue au cas d'une séquence unique. Pour obtenir cet ADS "généralisé" [Wang et al. 1994], on concatène les N séquences en ajoutant à la fin de chacune d'entre elles, soit Su, un symbole distinctif $£_u$ n'appartenant pas à l'alphabet. Il est possible de trouver le plus long bloc de facteurs répétés satisfaisant une contrainte de quorum q ($q \in \{2,…N\}$) en un temps linéaire par rapport à N [Hui 1992]. Wang et al.

(cf. 3.4.1.7) utilisent un tel ADS au sein de la première phase d'un algorithme d'extraction de patterns de forme particulière.

L'extraction de blocs de facteurs identiques s'effectue dans des conditions similaires avec des automates des suffixes (DAWGs) [Clift et al. 1986].

3.4.1.3. Autres approches basées sur l'égalité entre séquences

Quelques autres approches ont été proposées traitant le problème de l'extraction de blocs de facteurs répétés exactement. Celles proposées dans ([Leung et al. 1991 ; Karlin et al. 1988 ; Martinez 1983 ; 1988 ; Santibanez & Rohde 1987]) ont en commun de se fonder sur une indexation des mots par des entiers. Ces index sont, suivant les cas, manipulés dans des listes chaînées ou dans des tableaux, et l'extraction des blocs dans un corpus de longueur cumulée L prend un temps O(L.ln(L)). Indiquons à titre de remarque que la recherche de répétitions exactes n'est pas le seul problème pouvant être résolu par exemple par l'algorithme de Leung [Leung et al. 1991 ; Karlin et al. 1988] lequel peut, mais sous certaines conditions uniquement, prendre en compte des substitutions et des indels.

3.4.1.4. Passage de l'égalité entre symboles à une équivalence

Dans KMR et dans l'algorithme de Landraud, Avril & Chretienne l'égalité des symboles comparés deux à deux peut être remplacée par une *relation d'équivalence* ≈ sur Σ. Dans ce cas au lieu de blocs de facteurs identiques, ce sont des blocs de facteurs *équivalents* qui sont recherchés. Pour tout couple de facteurs (F,F'), F est équivalent à F', ce que nous notons F Equiv F', si et seulement si :

$$\begin{cases} |F'| = |F| \\ \{\forall i \in \{1,\ldots|F|\}\}\{F[i] \approx F'[i]\} \end{cases}$$

Equation 3-5

En revanche, lorsque la relation sur Σ n'est pas une relation d'équivalence, KMR et ses extensions nécessitent des modifications significatives, voir section suivante.

3.4.1.5. Relation de similarité non-transitive sur l'alphabet

Soldano, Viari & Champesme (1995) ont proposé une extension de l'algorithme KMR dans laquelle on n'exige pas l'égalité des symboles comparés deux à deux. Contrairement à ce que nous décrivions plus haut, il ne s'agit plus simplement de remplacer cette égalité par une relation d'équivalence : ici, la relation r sur Σ peut être non-transitive — ce qui lui interdit d'être une relation d'équivalence. En conséquence, on n'a plus affaire à des classes d'équivalence de facteurs identiques ou équivalents, mais à des *cliques* de facteurs équipollents (cf. Définition 2-3). Notons Equip la relation induite par r sur les facteurs (de la même manière que Equiv est induite par $\approx$, cf. Equation 3-5). Dans chaque clique de Equip, tous les facteurs sont en relation deux à deux par Equip. (Remarque : Nous réutiliserons la notion de clique, cette fois-ci dans le cadre de la théorie des graphes, au Chapitre 5).

Le cheminement algorithmique est similaire à celui qui a été décrit dans la section sur KMR (3.4.1.1). Le Lemme 3-1 reste valide, si l'on remplace simplement "égaux" par "équipollents" (au sens de Equip). La recherche des blocs de longueur K s'effectue par doublements successifs de la longueur des facteurs. Pour un corpus de longueur totale L, cela requiert un temps $O(L.g^K.\ln k)$ et un espace $O(L.g^K)$, où g est un entier dépendant de r que l'on appelle son *degré de non-transitivité*.

On peut vouloir rechercher des cliques possédant une propriété supplémentaire, à savoir la *maximalité*. Les cliques obtenues par l'algorithme évoqué ci-dessus ne la possédant pas forcément, un algorithme de filtrage est ensuite mis en œuvre. L'ensemble des deux algorithmes est appelé *KMR-Cliques*.

3.4.1.6. Modèle de Hamming

3.4.1.6.1. Distance de Hamming standard

Avant de décrire l'algorithme HMoivre [Sagot et al. 1995a] qui permet d'extraire des patterns basés sur des répétitions Hamming-approximatives, il nous faut dire quelque mots sur la famille d'algorithme dont il fait partie [Sagot 1996]. Celle-ci se décline en fonction :

- de la manière dont les patterns sont définis. Par ordre de "richesse" croissante :
    - il peut s'agir soit de **mots** sur Σ, soit de **produits de sous-ensembles de** Σ.
    - dans ce second cas, soit la collection de sous-ensembles de Σ que l'on choisit forme une couverture de Σ, soit on fait appel à la notion plus complexe de *couverture combinatoire pondérée* [Sagot & Viari 1996]
- du modèle de comparaison sous-jacent entre modèles et facteurs : identité, distance de Hamming ou distance de Levenshtein (cf. 3.4.1.7). Suivant la définition adoptée pour les patterns, cela requiert éventuellement de généraliser les modèles de comparaison.

L'une des originalités de l'approche, inspirée à l'origine par celle mise en œuvre dans KMR, est la suivante. Contrairement au cas de KMR ou de ses prolongements, où les facteurs sont comparés deux à deux, ici un pattern — appelé *modèle* — peut être "extérieur" au corpus, et va être confronté aux facteurs du corpus (recherche d'occurrences du modèle). Dans notre formalisme, cela revient à dire que deux facteurs sont déclarés équipollents s'ils sont tous deux occurrences d'un même modèle.

Les deux problèmes que l'on cherche alors à résoudre, inspirés de ceux définis au 3.4.1.1 dans le cadre de KMR, sont les suivants :

Etant donnés un corpus C={S1 .. SN} et une contrainte de quorum $q \in \mathbf{N}q$,

- Problème 1 : un entier K étant fixé, trouver tous les modèles, et leurs ensembles d'occurrences, de longueur K actifs dans au moins q séquences différentes du corpus
- Problème 2 : trouver la longueur maximale Kmax pour laquelle il existe au moins un modèle de longueur Kmax actif[13] dans au moins q séquences différentes du corpus, et résoudre le Problème 1 pour K=Kmax.

La combinatoire ne porte plus seulement sur l'exploration de l'espace des facteurs du corpus, mais aussi sur l'exploration de l'espace des modèles. Ceux-ci sont construits dans l'ordre des longueurs croissantes par récurrence, se basant sur un lemme similaire au Lemme 3-1. Les ensembles d'occurrence des modèles sont construits en même temps que ceux-ci, ce qui permet un contrôle sur l'exploration de l'espace (arbre) des modèles. En particulier, si l'ensemble des occurrences d'un modèle M ne vérifient pas la contrainte de quorum (M est alors qualifié de *non valide*), alors tout modèle contenant M ne la vérifiera pas non plus. Ainsi dans l'arbre des modèles valides (qui seuls sont recherchés) il est inutile d'explorer le sous-arbre descendant de M, lequel est donc élagué.

Un récapitulatif des différents algorithmes proposés par Sagot et al. peut être trouvé dans [Sagot 1996], page 256. Dans le cadre du présent mémoire, nous nous concentrons sur ceux le plus directement liés à notre problématique.

---

[13] C’est à dire (cf. supra) ayant une ou plusieurs occurrences.

L'algorithme HMoivre

HMoivre est celui des algorithmes proposés par Sagot et al. pour lequel les patterns (modèles) sont des mots sur Σ et le modèle de comparaison est la distance de Hamming "classique" [Sagot et al. 1995a]. Notons K la longueur maximale des modèles recherchés (ou celle du plus long modèle trouvé) et e le nombre maximal d'erreurs autorisées, c'est-à-dire la valeur maximale permise de la distance de Hamming entre un pattern et une occurrence. La complexité de HMoivre est :

- En temps : $O(L.K^{e+1}.|\Sigma|^{e+1})$
- En espace : O(L.K)

Remarque : le cas où les modèles sont des produits de sous-ensembles de Σ est pris en charge par l'algorithme HPoivre [Sagot, Viari & Soldano 1995]

3.4.1.6.2. Distance de Hamming valuée

Le concept de distance de Hamming valuée a été introduit au 3.2.4.1.2. Un algorithme implémenté dans le programme MACAW [Schuler, Altschul & Lipman 1991] utilise ce modèle de comparaison pour rechercher des blocs de facteurs similaires appartenant à un ensemble de séquences. La cardinalité des blocs peut descendre à 2, et l'algorithme utilise des techniques de programmation dynamique semblables à celles évoquées au 3.2.6. Il s'agit toutefois de techniques "dégénérées" du fait que l'on a affaire ici à un modèle d'édition dégénéré (les substitutions sont le seul type d'éditions autorisées). Pour un corpus de longueur cumulée L, l'algorithme s'exécute en un temps $O(L^2)$. L'heuristique principale permettant d'atteindre cette complexité consiste à imposer une contrainte forte sur les blocs de facteurs recherchés. Dans un bloc, toute paire de facteurs doit avoir une similitude supérieure à un seuil fixé au départ. (Nous verrons néanmoins qu'en pratique des contraintes bien moins fortes (5.4.2) peuvent donner de bons résultats, du moins sur des séquences musicales).

3.4.1.7. Modèle d'édition standard

Rappel : est dit *standard* tout modèle d'édition dans lequel l'ensemble des types d'appariements autorisés est $ETAA_{standard}$ = {Insertion, Délétion, Remplacement}. En particulier, la distance de Levenshtein avec substitutions (cf. 3.2.5.1) s'inscrit dans le cadre d'un tel modèle, contrairement à la distance de Hamming valuée.

3.4.1.7.1. Distance de Levenshtein avec substitutions

LeMoivre est celui des algorithmes proposés par Sagot et al. pour lequel les patterns (modèles) sont des mots sur Σ et le modèle de comparaison est la distance d'édition standard [Sagot et al. 1995a]. S'inscrivant dans le prolongement de HMoivre, LeMoivre se trouve nettement complexifié du fait d'un surcroît de combinatoire. Celui-ci est induit par le passage d'un modèle de comparaison de séquences à structure de correspondance fixe, à un modèle à structure de correspondance variable.

Notons K la longueur maximale des modèles recherchés (ou celle du plus long modèle trouvé) et e le nombre maximal d’erreurs autorisées (susbstitutions ou indels) entre un pattern et une occurrence. La complexité de LeMoivre est [Sagot et al. 1995a] :

- En temps : $O(L.K^{2e+1}.|\Sigma|^{e+1})$
- En espace : $O(L.K^{2e+1})$ ou $O(L.K^{2e}.|\Sigma|^{e})$ suivant la version de l'algorithme utilisée

Remarque : le cas où les modèles sont des produits de sous-ensembles de Σ est pris en charge par l'algorithme LePoivre [Sagot, Viari & Soldano 1995]. Celui où une couverture combinatoire pondérée est utilisée fait l'objet de LeCombi [Sagot 1996 ; Sagot & Viari 1996]

3.4.1.7.2. Distance de Levenshtein avec substitutions, en présence de jokers

L'approche présentée dans [Wang et al. 1994] vise des objectifs de découverte de patterns et également de classification basée sur les patterns, les applications étant présentées dans le domaine biologique.

Les patterns que l'on cherche à découvrir dans le corpus sont des expressions régulières de la forme *X1*X2*… *Xf où :

- X1, … Xf, … sont des facteurs du corpus (f : entier quelconque)
- chaque * représente un joker de longueur variable (*variable length don't care,* VLDCs), un symbole spécial pouvant être apparié avec n'importe quel facteur avec un coût nul du point de vue de la distance d'édition.

Le modèle de comparaison employé est une extension de la distance de Levenshtein avec substitutions. La valeur de l'alignement entre X et Y est le nombre minimal d'indels et de substitutions nécessaires pour transformer X en Y, après substitution optimale des VLDCs.

La stratégie générale proposée pour la découverte de patterns est en deux phases :

- 1. Rechercher des k-uplets de facteurs candidats dans un échantillon du corpus. Un arbre des suffixes "généralisé" (en version multi-séquences, cf. 3.4.1.2) est employé à cet effet. Puis :
- 2. Combiner les facteurs pour former des patterns candidats, et trouver les occurrences de ceux-ci dans l'ensemble du corpus.

Les caractéristiques des ensembles d'occurrences obtenus déterminent quels candidats sont sélectionnés. Les contraintes imposées aux patterns prennent la forme (entre autres) :

- de la longueur minimale des patterns recherchés, c'est-à-dire |X1| + … + |Xf|
- de la distance maximale pouvant exister entre un pattern et une occurrence
- du nombre minimal de séquences différentes dans lequel un pattern doit posséder des occurrences (contrainte de quorum)

3.4.1.8. Discussion

En guise de (brève) conclusion, soulignons que la complexité des algorithmes d'extraction de patterns dans les séquences dépend majoritairement du modèle de comparaison qui y intervient. La tendance est que plus ce modèle et riche, plus la complexité (en temps, particulièrement) augmente, cette complexité étant exprimée en fonction de la taille cumulée du corpus ainsi que, le cas échéant, de la longueur maximale des patterns extraits et de la taille de l'alphabet.

Une discussion plus approfondie sera proposée au 5.6 dans le cadre de la comparaison de ces différents algorithmes avec l'algorithme FlExPat que nous avons conçu et développé.

3.4.2. Algorithmes musicaux

Parmi les classifications possibles, nous choisissons ici de distinguer :

- D'un côté, les approches statiques (ou en différé — *offline*), dont le contexte typique est l'analyse musicale au sens standard du terme. Pour ces approches, l'extraction est susceptible de se faire simultanément sur l'ensemble de la/des séquences, lesquelles sont connues et accessibles en totalité au moment de l'extraction.

- De l'autre, des approches en-ligne (ou en direct — *online*)[14], dont le contexte est le jeu (accompagnement, improvisation) en direct. Dans ce cas, les séquences (séquence des notes ou des accords joués par un instrumentiste) ne sont connues que partiellement, et "dévoilées" progressivement, au cours du processus.

Remarque : Hormis la musique "live", d'autres exemples typiques de contextes d'extraction en-ligne sont les jeux (séquence des coups de l'adversaire) et les interfaces "intelligentes" (séquence des actions de l'utilisateur). Pour le jeu "Pierre Ciseaux Papier", Minasi (1991) a par exemple mis en œuvre un algorithme ad hoc de recherche en-ligne de séquences de coups répétées (exactement). A chaque coup, le système exploite les résultats de cette recherche pour prédire le prochain choix de son adversaire et faire son propre choix en conséquence.

3.4.2.1. Extraction statique

Les trois algorithmes présentés dans cette section ne font pas partie de ceux décrits au 3.4.1. Ce sont en effet des algorithmes ad hoc, dont un seul fait l'objet d'une description détaillée dans la littérature [Cope 1991a, 1992, 1990]. En ce qui concerne les deux autres [Stech 1981; Smaill, Wiggins & Harris 1993], notre présentation sera basée sur ce que les articles laissent supposer sur les algorithmes effectivement mis en oeuvre.

Seul le système de Cope est baptisé par son auteur dans la littérature citée : EMI (pour Experiments in Musical Intelligence) ; nous désignons respectivement les deux autres par "S81" et "SWH93". Le contexte scientifique et applicatif dans lequel se situent les trois systèmes a déjà été mentionné au 2.1.3.4 (page 43).

Les modèles précis de comparaison de facteurs intervenant dans les algorithmes sont tous paramétrés. Par suite, selon les valeurs de paramètres choisies, le type de répétitions recherchées est soit exact, soit — cas le plus fréquent — approximatif.

Pour présenter le fonctionnement de ces algorithmes statiques, nous décrivons tout d'abord celui d'EMI qui est le plus représentatif.

---

[14] Nous évitons l'adjectif "dynamique" ici. En effet, une partie des algorithmes d'extraction étant basés sur la programmation dynamique[Bellman 1957], une confusion pourrait être engendrée entre "algorithmes d'extraction dynamiques" et "algorithmes d'extraction basés sur la programmation dynamique".

3.4.2.1.1. Extraction de patterns dans EMI

EMI est un système de composition en temps différé visant à "répliquer" le style de différents compositeurs, par exemple Mozart ou Debussy. L'entrée est un corpus constitué d'au moins deux compositions dans un style similaire, dans lesquelles EMI tâche d'extraire des patterns appelés "signatures" stylistiques. Celles-ci sont des séquences (facteurs) de hauteurs et/ou de durées ayant été déterminées comme "significativement" récurrentes. Lors de la phase de recomposition, les signatures sont insérées dans un substrat compositionnel spécifique (par exemple un modèle de sonate). La partition 'interstitielle' est engendrée par un ATN (*Augmented Transition Network*) à partir de *règles compositionnelles* également extraites dans le corpus.

Le processus d'extraction est en deux volets :

- En premier lieu, des patterns récurrents sont recherchés dans chaque composition (séquence) du corpus. Il s'agit donc de répétitions *internes* uniquement. A cette fin, toutes les paires possibles de segments de même longueur (quelle que soit cette longueur) font l'objet d'un test d'équipollence. Celui-ci peut concerner les points de vues (jeux de descriptions) 'hauteurs uniquement', 'durées seulement' ou 'hauteur + durée'. Les hauteurs — *midipitches* — sont redécrites en intervalles chromatiques ce qui rend l'équipollence invariante à la transposition chromatique (voir 3.2.4.1.1). Avec le jeu de descriptions 'hauteur + durée' chaque note, sauf la première de chaque séquence source, se trouve alors décrite par le couple dont :

    q Le premier élément (un entier relatif) est l'intervalle chromatique — nombre de demi-tons — formé avec la note qui précède

    q Le second élément (un rationnel) est la durée de la note.

  Les patterns correspondent aux différents ensembles de facteurs équipollents trouvés. Suivant le paramétrage choisi, ces ensembles sont soit des classes d'équivalence, soit des cliques (cf. 3.4.1.5), la relation d'équipollence n'étant plus transitive. Rien n'est dit sur la manière dont on effectue l'indispensable opération de choix d'un représentant (prototype) dans chaque ensemble.

- En second lieu, les patterns "internes" trouvés dans les différentes compositions sont comparés. (Il faut supposer que ce sont en fait les représentants qui sont comparés). Les patterns identiques ou équipollents trouvés dans des compositions différentes acquièrent le statut de signature.

3.4.2.1.2. Schéma algorithmique

L'exemple d'EMI illustre chacun des éléments du schéma algorithmique proposé en Figure 3-10. Ce schéma général unifie celui des trois systèmes : EMI, S81 et SWH93. Les structures de données sont représentées par des rectangles, les traitements par des ellipses. L'une des ellipses est en pointillé pour traduire le caractère optionnel du traitement correspondant.

- La *sélection de segments* consiste à choisir et énumérer l'ensemble des paires de segments de la ou des séquence(s) source(s) qui seront comparées. Dans les trois systèmes, toutes les paires possibles de segments sont sélectionnées. Ceci diffère sensiblement de ce qui se passe pour les systèmes d'extraction en-ligne, comme on le verra plus bas.

- Le caractère récurrent d'un segment est testé en le comparant avec d'autres segments (ellipse "comparaison de segments par paires"). Il faut qu'au moins un autre segment lui soit équipollent. Plusieurs paradigmes différents de comparaison de séquences, présentés au 3.2 (page 59), sont utilisés.

- Les *patterns candidats* sont issus des résultats des comparaisons de segments. Même si cela n'est pas explicite, les patterns candidats sont définis par les classes d'équivalence (ou les cliques) de segments.

- Enfin, la *sélection de patterns* consiste à faire un choix parmi les candidats sur un critère de "proéminence" correspondant au nombre et/ou à la nature des segments dans chaque classe/clique. Par exemple, dans EMI un pattern candidat est sélectionné si et seulement si la classe d'équivalence ou la clique associée contient au moins deux segments provenant de séquences sources *différentes*. Ce critère de quorum (q=2) traduit la définition que donne Cope d'une signature : un pattern apparaissant dans plusieurs pièces d'un compositeur donné. Dans S81 et SWH93, il n'y a pas d'opération de sélection de patterns, les patterns candidats étant systématiquement sélectionnés.

  Il faut aussi choisir un représentant de chaque pattern. SWH93 est le seul des trois systèmes pour lequel ce choix est explicité : il s'agit de la première occurrence, appelée *motif* — les autres occurrences étant appelées *derivations*.

Le processus d'extraction n'intervient qu'une fois (pour l'ensemble du corpus). De ce point de vue là, SWH constitue en quelque sorte une exception, puisqu'il fait intervenir deux passes dans lesquelles le paradigme de comparaison de segments diffère.

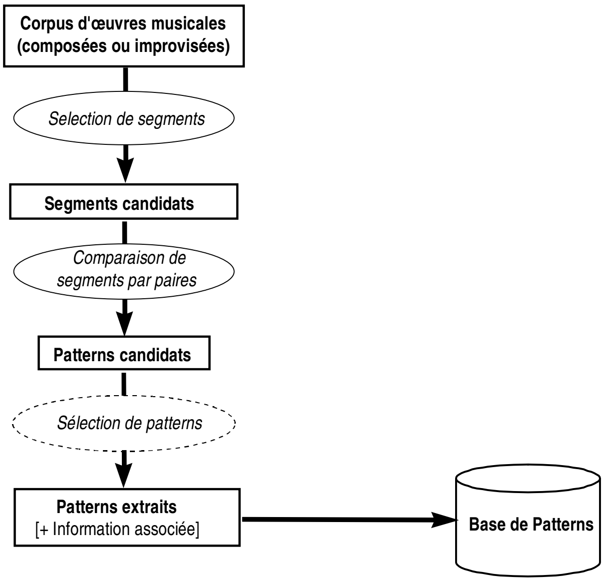


Figure 3-10. Schéma algorithmique unifiant les systèmes existants d'extraction **statique** de patterns musicaux

3.4.2.2. Extraction en-ligne

Nous décrivons dans cette section les approches mises en œuvre sur des séquences musicales au sein des systèmes de Pennycook, Stammen & Reynolds (1993) et de Rowe (1993). Dans les deux cas il s'agit de recherche de répétitions approximatives.

Le premier système, que nous baptisons "PSR93", comme le second, Cypher, sont des systèmes musicaux interactifs temps-réel. Si PSR93 est spécialisé dans la production d'improvisations en jazz tonal, Cypher en revanche implémente divers algorithmes de génération musicale non rattachés à un genre en particulier. S'appuyant sur des principes introduits par Minsky (1986), les deux systèmes sont basés sur la coopération de plusieurs composants appelés *agents*. Parmi ces derniers, les plus importants sont le *Listener* et le *Player* ; lors d'une session (*performance*), le Listener cherche à découvrir dans le flux musical temps-réel des patterns destinés à être réutilisés par le Player. La réutilisation peut être précédée d'une phase d'adaptation ou de transformation. Les mécanismes d'extraction de patterns respectifs de Cypher et de PSR93 peuvent être décrits à partir du schéma algorithmique proposé en Figure 3-11.

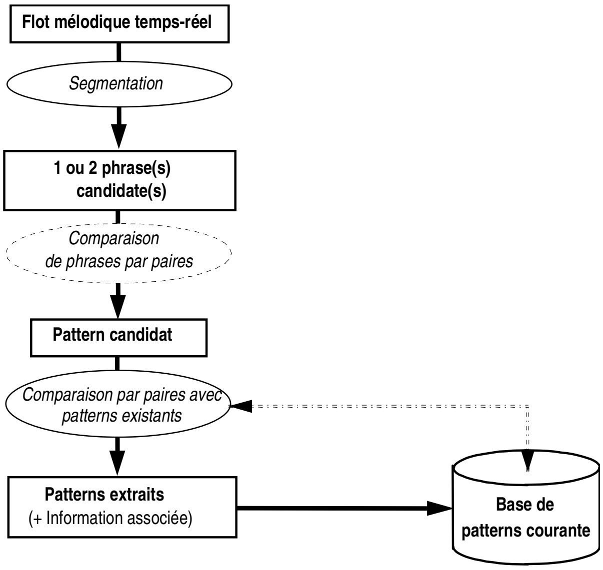


Figure 3-11. Schéma algorithmique unifiant les systèmes existants d'extraction **en-ligne** de patterns musicaux

- Le flux entrant est formé de notes de type MIDI (c'est-à-dire décrites par leur midipitch, leur durée et leur amplitude).

- Des agents spécialisés se chargent de segmenter le flux temps-réel en phrases monodiques —*"melodic groups"* [Stammen & Pennycook 1994]. Nous parlons ici de phrases plutôt que de segments. En effet les segments dont il était question à la section 3.4.1.8 correspondaient à l'énumération combinatoire de tout ou partie des facteurs d'un corpus. Les frontières de chaque segment étaient donc fixées par la stratégie d'énumération utilisée. Ici au contraire la mise en œuvre de mécanismes de segmentation vise à isoler des passages dont le choix des frontières résultent de considérations musicales multiples. Celles-ci s'appuyant notamment sur des théories et résultats expérimentaux en psychologie de la cognition musicale. Compte tenu de cela, les parties de séquence sur lesquelles opèrent Cypher et PSR93 correspondent plus à la définition de phrases mélodiques qu'à celle, générique, des segments mélodiques employée au 3.4.1.8.

- Une redescription (changement de représentation) intervient au niveau des phrases :

  ☐ Dans Cypher comme dans EMI (cf. 3.4.2.1.1) chaque phrase est redécrite sous la forme d'une séquence d'entiers relatifs représentant la taille de l'intervalle chromatique séparant chaque note de la précédente, c'est-à-dire le nombre algébrique de demi-tons correspondant. Nous avons déjà indiqué l'intérêt de cette redescription en termes d'invariance à la transposition chromatique.

- Dans PSR93, chaque phrase est redécrite en une séquence de couples {intervalle chromatique, *ratio de durée*} ∈ **Z**×**R**. Le ratio de durée est (rappel) défini comme le quotient entre la durée de chaque note et la durée de la note qui précède, soit : X[i].durée / X[i+1].durée (voir 4.7.7.3.2 pour plus de détails).

- Ensuite, dans Cypher, les deux dernières phrases segmentées font l'objet d'un calcul de similitude. Un modèle d'édition valué à contributions constantes est employé (cf. 3.2.5.2.1) ; la contribution de tout indel ou de toute substitution vaut 0, celle de toute identité vaut 1. La similitude calculée est comparée à un seuil constant fixé à 4, indépendamment de la taille des phrases comparées. Si on est au-dessus du seuil, cela donne lieu à un pattern candidat. Malheureusement, la description fournie [Rowe 1993, p. 246] ne permet pas de savoir si le pattern candidat est l'une des deux phrases ou s'il est au contraire *construit* (par exemple, généralisé) à partir des deux phrases. Tout au plus infère-t-on qu'il s'agit d'une séquence du même type que les deux phrases.

  Dans PSR93, les deux phrases ne sont pas comparées, le pattern candidat étant tout simplement la dernière phrase segmentée.

- Le pattern candidat est alors comparé à l'ensemble de la base de patterns extraits précédemment. Pour chaque paire {pattern candidat — pattern de la base}, un calcul de similitude est effectué. (PSR93 utilise des distances au lieu de similitudes, mais cela ne fait aucune différence dans le contexte où nous sommes, compte tenu de la dualité qui existe entre fonction de similitude et distances / fonction de dissimilitude — cf. plus haut). Cypher emploie la même fonction de similitude que pour la selection de patterns candidats (ci-dessus). PSR fait appel à un modèle d'édition valué à coûts variables (section 3.2.5.2.2). Pour les deux systèmes, un test d'équipollence est positif (*"match"*) lorsque la valeur de la similitude entre le pattern candidat et le pattern de la base est supérieure à un seuil. Dans Cypher, la valeur, constante, de ce seuil est documentée : 4 comme précédemment.

  - Si le test d'équipollence est positif avec un pattern précédemment extrait Pat, il n'y a pas d'ajout de pattern à la base. En revanche, l'*activation* ("*strength*") de Pat est augmentée — les patterns avec la plus forte activation étant ceux que sélectionne le plus souvent le Player de Cypher.

- Si le test d'équipollence est négatif, le pattern candidat est réputé "unique" et est ajouté à la base de patterns.

  Remarque : l'extraction d'un pattern exige que celui-ci soit unique par rapport aux patterns existants ; pourtant, à terme on obtient une base de patterns récurrents, ce qui peut d'une certaine manière sembler paradoxal…

Le processus d'extraction est exécuté à chaque nouvelle segmentation de phrase. A chaque exécution, un pattern au plus est extrait.

3.4.2.3. Synthèse

Un certain nombre de différences interviennent entre les deux types d'approches décrites plus haut. Cependant, nous proposons à la Figure 3-12 un schéma algorithmique qui unifie ceux proposés plus haut, respectivement, pour l'extraction de patterns statique et pour l'extraction de patterns en-ligne.

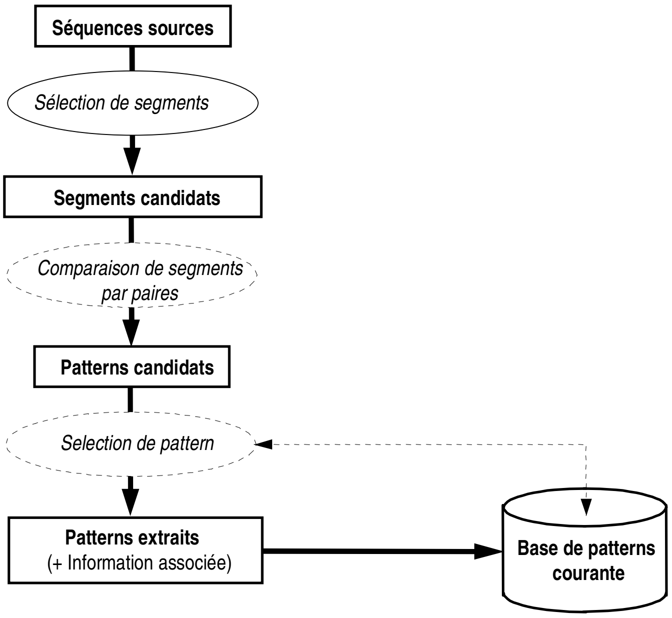


Figure 3-12. Extraction de patterns musicaux : schéma algorithmique unifié des systèmes existants (statiques ou en-ligne)

Les quatre différences clés entre les deux types d'approches peuvent maintenant être présentées à partir de ce schéma. Détaillées au Tableau 3-1, elles concernent respectivement :

- La nature de la/des séquence(s) source(s)
- Les principes de sélection de facteurs mis en oeuvre

- Le paradigme de sélection de patterns
- Le caractère périodique ou non de l'exécution du processus d'extraction de patterns.

Tableau 3-1. Différences clés entre extracteurs de patterns statiques et en-ligne

Type d'extracteur de patterns	**Séquences sources (corpus)**	**Paradigme de sélection des facteurs à comparer**	**Critère de selection des patterns**	**Périodicité des appels au processus d'extraction**
Statique (systèmes d'analyse et/ou de recomposition)	Ensemble d'œuvres composées ou improvisées	Exhaustif (→ "segments")	Mesure de proéminence	Unique (un appel par corpus)
En-ligne (systèmes musicaux interactifs)	Flux musical temps-réel	Segmentation (→"phrases")	Comparaison avec les patterns extraits précédemment	Cyclique (un appel à chaque nouvelle segmentation)

Chapitre 4. LOCALISATION ET COMPARAISON — LE SYSTÈME IMPROLOGY

4.1. Organisation du chapitre

Le plan de ce chapitre est le suivant. Dans un premier temps, nous discutons les caractéristiques et les avantages spécifiques présentés par les séquences mélodiques en général, et plus particulièrement les transcriptions de jazz improvisé appartenant au genre *Be-bop,* telles que celles étudiées par Owens (1974). Puis nous donnons une description générale du système Imprology que nous avons conçu et développé en vue de la découverte de régularités dans les séquences mélodiques (localisation et extraction de patterns). Le corpus sur lequel a porté l'étude d'Owens (1974) et le lexique de patterns extraits par ce dernier sont ensuite présentés. Nous proposons au 4.5 un algorithme de localisation de patterns (passages) mélodiques, implémenté dans Imprology, dont le critère d'équipollence utilise un modèle d'édition valué à fonctions de contribution non constantes — ou tout modèle plus simple[15]. Un ensemble de résultats expérimentaux, obtenus avec cet algorithme sur une partie du corpus et du lexique de patterns d'Owens, sont présentés (4.6). Ces résultats fournissent une validation empirique du *principe de la connaissance* évoqué précédemment, dans le contexte de la découverte de régularités dans les séquences mélodiques. Sur cette base, nous introduisons (4.7) un nouveau modèle général de comparaison entre entités séquentielles, le *modèle d'edition valué multi-descriptions* (MEVM).

---

[15] C'est-à-dire tout modèle subsumé par le modèle d'édition valué à fonctions de contribution non constantes.

4.2. Séquences mélodiques

La musique occupe une place privilégiée en tant que domaine d'application de nos travaux. En particulier, nous avons effectué un certain nombre d'expérimentations sur un gros sous-ensemble du corpus étudié par Owens (1974), corpus constitué de transcriptions de solos de jazz tonal (Be-bop) improvisés par le saxophoniste Charlie Parker. L'objet de cette section est de proposer différents arguments appuyant ces choix : choix du jazz improvisé, puis choix du genre Be-bop, puis choix de Charlie Parker. Cela nous permettra au passage de mettre en lumière les caractéristiques spécifiques des séquences dont il s'agit.

4.2.1. Musique improvisée, jazz

Les problèmes posés de découverte de motifs sont en général nettement plus intéressants lorsqu'on a affaire à de la musique improvisée telle que le jazz, que lorsqu'on a affaire à de la musique *composée* telle que la musique tonale occidentale (baroque, classique). Cela tient principalement au fait que, pour cette dernière, les variations affectant les occurrences de motifs sont la plupart du temps triviales (transposition globale ou changement de mode mineur / majeur par exemple). Par contraste, le jazzologue B. Kernfeld note à propos du jazz que « l'improvisation à base de motifs fait très rarement intervenir les répétitions et transpositions systématiques entendues souvent en musique classique » [Kernfeld 1989], ce qui fait de la simulation de l'analyse musicale orientée-patterns, un problème nettement plus intéressant (et difficile) sur le jazz improvisé que sur la musique tonale occidentale.

L'intérêt du jazz par rapport à d'autres genres musicaux (comme par exemple la musique dite « contemporaine » ou « nouvelle ») est d'être une musique qui, fortement enracinée culturellement, a fait l'objet de nombreux travaux de théorisation et/ou de formalisation (cf par exemple Baudoin 1992a; b; Coker 1964; Gridley 1985; Mehegan 1962). Aux travaux qui viennent d'être cités s'ajoutent une partie des très nombreux travaux théoriques, au sens large, effectués dans le domaine de la musique tonale en général (par exemple, [Schenker 1954 ; Meyer 1956 ; Lerdahl & Jackendoff 1983 ; Narmour 1989 ; 1992]). Vu sous l'angle de l'intelligence artificielle, cela permet de formaliser, représenter et utiliser différents types de connaissances en vue de doter le système que l'on construit d'une plus grande « intelligence musicale » — pour reprendre une expression due à Cope (1991a).

4.2.2. Jazz Be-bop

Au sein du jazz, les avantages liés au choix du genre Be-bop (ou « Bop ») résulte de plusieurs éléments. Tout d'abord, contrairement à d'autres périodes du jazz comme le *swing* ou *mainstream*, le Be-bop est caractérisé par une recherche musicale globalement plus axée sur la ligne mélodique, le rythme, le « swing » etc. que sur le *timbre*, c'est-à-dire la sonorité des instruments. Ainsi, à étudier un corpus de transcriptions de solos (partitions) dans le style Be-bop, le musicologue (et par suite l'ordinateur) perd moins par rapport à l'enregistrement sonore que s'il s'agissait de jazz appartenant à d'autres styles. D'autre part on s'accorde à dire que, aux tempi très rapides qui caractérisent de nombreux morceaux Be-bop (par exemple « Koko » par C.Parker, 300 à la noire), aucun être humain ne serait capable d'engendrer des improvisations inventées de toutes pièces en temps-réel, et qui soient tout à la fois agréables à entendre, cohérentes et recherchées. Cela entraîne que la plupart des improvisateurs du Be-bop possèdent plus ou moins consciemment un lexique, ou catalogue, de patterns mélodiques travaillés et retravaillés à l'avance. Différents spécialistes de l'enseignement du jazz, par exemple Coker (1964), conseillent d'ailleurs aux improvisateurs de se constituer au fil du temps un tel lexique personnel de patterns les plus personnels possibles, qu'ils doivent adapter pour les utiliser lors de l'improvisation.

4.2.3. Charlie Parker

Charlier Parker (dit « Bird » ou encore « Yardbird[16] ») est considéré par les spécialistes comme l'un des instigateurs du style Be-bop, avec d'autres comme Dizzy Gillespie ou Thelonious Monk, et par certains comme le musicien le plus innovateur et le plus talentueux de cette époque du jazz (cf. par exemple [Malson 1993; Owens 1974]). Pour sa part Kernfeld (1989) présente le saxophoniste comme le plus grand maître de ce que le jazzologue appelle « formulaic improvization », et que nous appellerions « improvisation basée sur les motifs » par analogie avec le terme « analyse musicologique basée sur les motifs » introduit précédemment. Par ailleurs, Parker improvisait toujours différemment sur un même morceau, au contraire d'autres improvisateurs célèbres tels qu'Armstrong, comme le précise Kernfeld[17]. Ces différentes caractéristiques de la technique d'improvisation de Parker fait de celle-ci un

---

[16] Les refus obstinés de Parker, à certaines périodes de sa tumultueuse existence, de tenir ses engagements (concerts, enregistrements) lui ont valu quelques autres *noms d'oiseaux…*

[17] « *Parker never repeated an entire solo, and successive performances based on the same tune are sometimes startlingly different (...). Armstrong, once having arrived at a succesful approach, might repeat the contour and many details of a solo in different performances.* » (Kernfeld 1989, p. 562).

objet d'étude idéal pour nous. Sans oublier bien entendu l'intérêt considérable que constitue l'existence et la disponibilité de l'étude d'Owens.

4.3. Le système Imprology

4.3.1. Caractéristiques générales

Bien que la visée de nos travaux ne se restreigne pas à des applications musicales, le système Imprology — déjà mentionné au 2.1.3.4— qui implémente les algorithmes décrits dans ce chapitre et dans le Chapitre 5 est, pour sa part, conçu pour la localisation et l'extraction de patterns musicaux.

Le nom choisi pour le système, au-delà de ce qu'évoquent les racines qu'il contient — "étude de musique improvisée" — est un clin d'œil au jazz Be-bop et à Charlie Parker. Faisant echo à une mode de l'époque consistant à donner des noms en "ology" aux thèmes, « Imprology » évoque « Ornithology », « Crazeology » et autres « Anthropology », autant de thèmes créés ou joués par les musiciens du Bop et en particulier par Bird…

Imprology est implémenté en Smalltalk-80 [Goldberg & Robson 1983], qui appartient à la catégorie des langages orienté-objets. Cette catégorie de langages, aujourd'hui très répandue, est particulièrement bien adapté à la réprésentation de concepts et d'entités musicales. Précisément, le système réutilise une partie de la plate-forme de représentation de connaissances en musique tonale MusES [Pachet 1994 ; Pachet, Ramalho & Carrive 1996]. Cet ensemble de classes et de méthodes permet de visualiser et manipuler efficacement la plupart des concepts fondamentaux de la musique tonale: notes et pitchclasses, durées, hauteurs, gammes, tonalités, accords, monodies, polyphonies, tempo, mesures, séquences d'accords etc. A travers le développement d'Imprology nous avons réutilisé une partie importante de MusES et par ailleurs enrichi ou ajouté d'autres parties. La plupart des ajouts se rapportent à la prise en compte de descriptions musicales supplémentaires. Parmi les autres ajouts, nous avons ajouté à MusES des fonctionnalités de conversion de données musicales depuis et vers le format commercial Standard MIDI File [SMF 1988].

L'implémentation de Smalltalk-80 employée est l'environnement VisualWorks de ParcPlace-Digitalk. Il peut fonctionner aussi bien sur PC sous Windows 32 bits ou 16 bits, sur Macintosh sous MacOS ou sur station sous Unix (SunOS par exemple), VisualWorks étant lui-même entièrement portable entre ces plate-formes.

4.3.2. Importation de données musicales

En analyse musicale, usuellement le matériau musical étudié est disponible sous forme de partitions. Pour pouvoir travailler sur des corpus en vraie grandeur, il nous était indispensable de disposer de fonctionnalités de conversion depuis (et, dans une moindre mesure, **vers**) un format standard de stockage informatique de données musicales, en l'occurrence le format commercial *Standard MIDI File* (SMF) ; il était hors de question de saisir à nouveau, au moyen de l'éditeur de partitions de MusES, des pièces déjà saisies au format SMF.

Nous avons conçu et intégré à MusES un convertisseur de format (appelé *SMF2MusESConverter)* permettant d'importer les données contenues dans des fichiers au format Standard MIDI File [SMF 1988] pour les convertir en objets MusES (typiquement, une instance pleinement spécifiée de la classe Monody ou de la classe Polyphony pour chaque fichier de séquence SMF). SMF2MusESConverter, écrit en Smalltalk, permet de construire automatiquement les *objets* Smalltalk manipulés par MusES qui correspondent au contenu du MIDI File : parties instrumentales, notes (hauteur, instant de début et de fin, amplitude, etc.), tempo, armure, etc.

4.4. Corpus et patterns de référence

Un certain nombre d'expérimentations ont été effectuées avec Imprology sur un corpus qui est un (gros) sous-ensemble de celui de l'étude d'Owens (1974), évoquée précédemment. La richesse et le caractère quantitatif de cette étude fournit un support de validation exceptionnel pour Imprology.

L'étude d'Owens constitue donc un terrain d'investigation d'une grande richesse pour nos travaux ; cela apparaîtra notamment au travers des expérimentations présentées, par exemple celles relatives au dénombrement et à la localisation d'occurrences de patterns découverts par Owens dans des parties du corpus (4.6). Pour éviter la redondance qui résulterait d'un appel de référence bibliographique complet à chaque fois que nous nous y référons, souvent nous indiquerons simplement le numéro de volume (I ou II), accompagné le cas échéant de numéros de page.

4.4.1. Corpus de référence

Comme nous l'avons vu, d'autres analyses comprenant une composante de découverte de patterns ont été effectuées [Kernfeld 1983; Baker 1980; Smith 1983]. Cela dit, celle d'Owens constitue à nos yeux un cas unique non seulement parce que la majeure partie du matériau d'étude — très conséquent — est lui-même mis à disposition du lecteur, mais aussi de par la quantification qui est effectuée (dénombrements d'occurrences, « statistiques »). Le corpus que nous avons constitué à partir de celui d'Owens regroupe 146 séquences (solos monodiques) parmi les 250 transcrits par le musicologue, représentant collectivement un peu plus de 10 000 mesures[18] et environ 60 000 notes. Nous le désignons par *corpus de référence*. Conformément à la structure de groupement adoptée par Owens, nous répartissons les différents solos en fonction de la tonalité du morceau auquel ils appartiennent. A titre d'exemple, la catégorie « do majeur » est constituée de 34 solos. Au sein d'une même tonalité, un second niveau de groupement concerne le type de grille d'accords : « Blues », « I got Rhythm » et « Autre ». Par exemple, il y a 17 solos « do majeur Blues », 7 morceaux « do majeur I Got Rhythm » et 10 morceaux « Autre ».

4.4.2. Lexique de patterns de référence

Owens a extrait dans le jeu improvisé de C. Parker un lexique d'environ 200 patterns (cf. Figure 2-1, page 44, et Annexe A). Indiquons que, dans sa thèse, Owens appelle ceux-ci "*motives*", mais a depuis remis en cause ce terme ici impropre — pour les raisons exposées au 2.1.3.2 — à la faveur des termes « formule », « figure » ou encore « pattern mélodique » [Owens 1996]

Les identificateurs de patterns proposés par Owens (1, 1A, 1Aa, etc.) induisent une **hiérarchie** au sein du lexique. Dans la Figure 4-1 et la Figure 4-2, nous donnons une représentation hiérarchique d'une partie de ce lexique. Les patterns (feuilles) apparaîssent en caractères gras, par opposition aux *groupes* abstraits (nœuds internes) qui sont en caractères maigres. Ces figures illustrent la diversité morphologique de cette hiérarchie (comparer par exemple le nœud 2B (feuille) avec son frère 2A qui, lui, possède des fils et des petits-fils, ou encore la taille de la descendance de 1A par rapport à celle de son frère 1C). En ce qui concerne les 3 premiers niveaux (en dessous du nœud origine), l'arbre compte 64 nœuds, internes ou terminaux, de niveau 1 (numérotés 1 à 64), 33 nœuds de niveau 2 (1A, 1B, etc.), et 78 nœuds de niveau 3 (1Aa, 1Ab, etc.). Une autre diversité existe au sein de ce lexique, qui est avantageuse dans le cadre de nos expérimentations : celle de la longueur (nombre de notes) des patterns, qui varie (pour la majorité) entre 4 et 25 notes.

---

[18] De métrique 4/4, à quelques exceptions près.

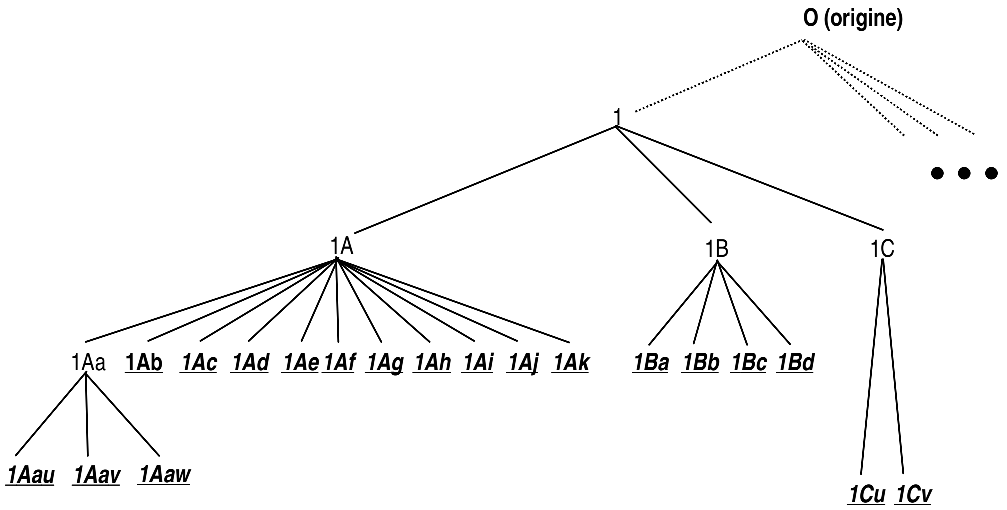


Figure 4-1. Représentation hiérarchique d'une partie du lexique d'Owens

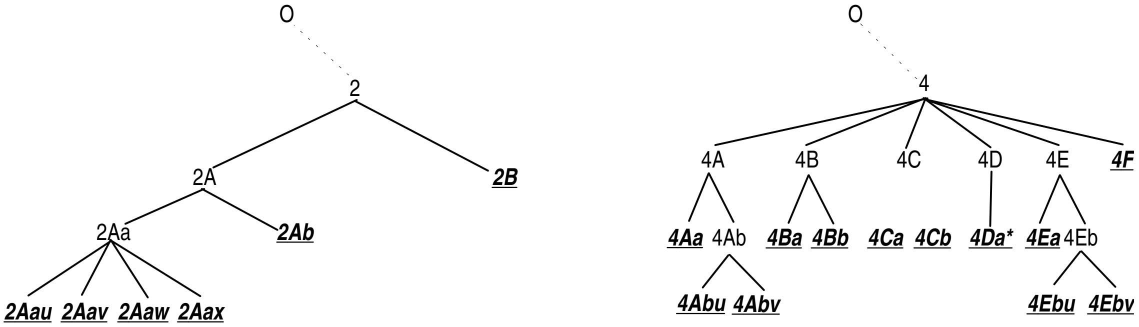


Figure 4-2. Suite de la figure précédente

4.4.3. Saisie

L'opération de saisie informatique de partitions, version musicale de la frappe dactylographique, portait ici un volume considérable — environ 450 pages de partitions denses pour l'ensemble des 146 transcriptions de solos, plus une dizaine de pages pour les patterns.

Il existe certes des systèmes de reconnaissance de partitions (*optical music recognition*) [cf. par exemple Fujinaga et al. 1989 ; Pennycook 1990 ; Coüasnon & Retif 1998], permettant de convertir une partition (scannée) en fichier MIDI. Cependant les tentatives que nous avons effectuées avec le logiciel commercial populaire MIDISCAN [Musitek 1995] sur les partitions d'Owens qui sont (très proprement) manuscrites se sont révélées totalement infructueuses (moins de 5% de notes reconnues !...). Il a donc fallu recourir à une saisie manuelle du corpus sur séquenceur / éditeur de partitions du commerce, offrant des interfaces de saisie relativement conviviales. Les séquences

MIDI obtenues (fichiers SMF) ont ensuite été importées dans Imprology par l'intermédiaire du SMF2MusESConverter mentionné plus haut.

4.5. Paradigme de localisation

4.5.1. Introduction

Notre problème, baptisé LOC, consiste à trouver toutes les occurrences approximatives d'un ensemble de patterns P1,…PM au sein d'un corpus C={S1,…SN}, comme illustré sur la Figure 4-3.

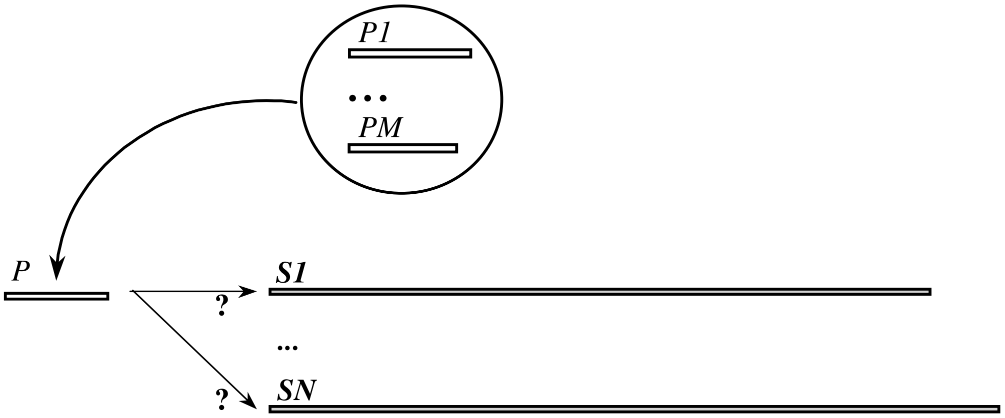


Figure 4-3. Localisation d'un ensemble de patterns dans un coprus

Nous avons présenté au 3.3.2 (page 100) un algorithme permettant de localiser **la** meilleure occurrence approximative (ou **les** meilleures en cas d'ex æquo). Cet algorithme ne permet donc pas directement de résoudre le problème LOC. De plus, suivant le domaine applicatif il n'est pas pas nécessairement adéquat que les alignements définissant les différentes occurrences soient des alignements loco-globaux, c'est-à-dire ayant comme séquence origine la **totalité** du pattern. En particulier, au travers de l'observation de la forme prise par les occurrences des patterns d'Owens tels qu'elles ont été localisées par lui, nous avons constaté que, très souvent, un occurrence ne concerne qu'une partie (importante) du pattern. (Voir à ce sujet la section 4.6.2.1 sur la méthodologie d'*annotation* des occurrences que nous avons proposée et mise en œuvre dans le cadre d'expérimentations sur la localisation d'occurrences approximatives avec Imprology).

Exemple

A titre d'exemple, la Figure 4-5 montre l'annotation traduisant une occurrence approximative du pattern 3Ac du lexique d'Owens (voir Figure 4-4) au sein du solo "Kim Take 2" (30/12/1952, Verve MGV 8005) [Vol. I, p. 31]. Cette annotation est tirée de l'une des 4 seules transcriptions annotées que fournit Owens — représentant en tout 4 pages de partition sur un total d'environ 500. Sur la figure, nous limitons du contexte de l'occurrence à une taille d'une mesure. ***N.B. :*** sur les partitions montrées les choix enharmoniques (sol dièse vs. la bémol ; do bémol) relèvent exclusivement du logiciel séquenceur à partir duquel les figures ont été obtenues.

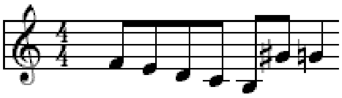


Figure 4-4. Le pattern 3Ac du lexique d'Owens

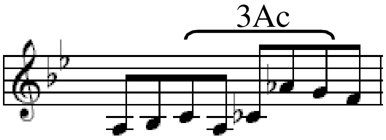


Figure 4-5. Une occurrence (approximative) du pattern 3Ac du lexique d'Owens dans "Kim Take 2"

Compte tenu du segment choisi par Owens pour signaler l'occurrence, il est clair que le passage devant être mis en correspondance au niveau du pattern est un segment ne l'englobant pas en totalité. Une possibilité pour un tel segment est proposée sur la Figure 4-6. †

Ce qui précède peut se traduire exactement par le fait de définir chaque occurrence par un alignement **local** (optimal) et non par un alignement loco-global. Dans le cadre de l'exemple proposé, un tel alignement entre les deux segments considérés pourrait être celui représenté par sa trace généralisée à la Figure 4-7.

Par suite, avant de présenter l'algorithme que nous avons conçu et implémenté au sein d'Imprology dans cette perspective, il nous faut présenter un certain nombre d'aspects algorithmiques relatifs à la comparaison locale (cf. 3.2.6.3, page 93).

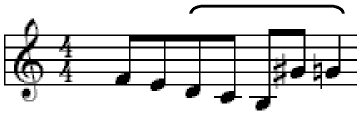


Figure 4-6. Segment du pattern 3Ac pouvant être mis en correspondance avec le segment de “Kim Take 2” annoté plus haut

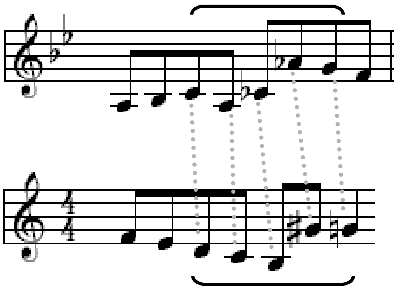


Figure 4-7. Représentation (trace généralisée) d’un alignement local possible entre les deux segments

4.5.2. Comparaison locale par programmation dynamique

Ce type de comparaison a été introduit, dans le cadre du modèle d'édition, au 3.2.6.3.

Soient X et X’ les deux séquences à comparer localement, de longueurs respectives n et n’. Plaçons-nous dans le cadre d’un modèle de similitude (l’alignement local dans le cadre d’une distance ne se traite pas aussi bien [Mongeau & Sankoff 1990 ; Waterman 1995]). L’objectif est de trouver un quadruplet (c,d,c’,d’), où $1 \leq c \leq d \leq n$ et $1 \leq c' \leq d' \leq n'$, tel que :

$$\mathrm{Simil}(X[c..d], X'[c'..d']) = \max\{\mathrm{Simil}(F,F'), (F,F') \in \underline{\mathrm{facts}}(X) \times \underline{\mathrm{facts}}(X')\}$$

Autrement dit, on recherche les coordonnées (c,d) et (c’,d’) de facteurs F et F’ maximisant Simil(F,F’). Nous notons le maximum Similoc(X,X’). Ici encore pour la clarté d'exposition nous supposons que $ETAA=ETAA_{standard}$={Insertion, Délétion, Remplacement}.

4.5.2.1. Algorithme naïf

Un tel algorithme consiste à énumérer toutes les paires possibles (F,F') et à effectuer pour chacune une comparaison globale (cf. 3.2.6.1). Le nombre de paires possibles (F,F') est :

$$C_{n+1}^{2} \times C_{n'+1}^{2} = \frac{n(n+1)}{2} \times \frac{n'(n'+1)}{2}$$

Pour chaque comparaison globale, nous avons vu que le temps requis est proportionnel au produit des longueurs des deux facteurs comparés. Au total, on obtient pour cet algorithme naïf une complexité en temps en O(n^6) dans le cas où n=n' [Waterman 1995, p. 203]…

4.5.2.2. Algorithme adapté

L'algorithme que nous allons présenter [Smith & Waterman 1981 ; Kruskal & Sankoff 1983] peut être vu, de même que pour la comparaison loco-globale (0), comme une variante de l'algorithme de comparaison globale. Il utilise des techniques de programmation dynamique, faisant intervenir une matrice explicite ou implicite SIMIL_L de dimension n×n' et dont le terme général est défini par :

$$\text{SIMIL\_Li, j} = \max\begin{cases} 0 \\ \max\{Simil(X[c..i], X'[c'..j]), 1 \le c \le i, 1 \le c' \le j\} \end{cases}$$

Cela signifie que SIMIL_Li,j est la valeur maximale de tous les alignements entre un suffixe de X[1..i] et un suffixe de X'[1..j].

On a alors la propriété suivante :

Propriété 4-1

La valeur recherchée Similoc(X,X') *est le plus grand élément de* SIMIL_L.

Précisons à présent comment calculer SIMIL_L.

- On l'initialise en mettant à zéro tous les éléments de la colonne 0 et de la ligne 0 :

$$\{\forall i \in \{0,\ldots,n\}\}\ \{\text{SIMIL\_Li,0} = 0\}, \text{ et :}$$
$$\{\forall j \in \{0,\ldots,n'\}\}\ \{\text{SIMIL\_L0,j} = 0\}$$

- Puis on fait appel à une propriété de récurrence :

Propriété 4-2

Pour tout (i,j) ∈ **N**n×**N**n' on a :

$$SIMIL\_L_{i,j} = \max \begin{cases} 0 \\ contrib\begin{pmatrix} X[i] \\ X'[j] \end{pmatrix} \quad + \quad SIMIL\_L_{i-1,j-1} \\ contrib\begin{pmatrix} X[i] \\ - \end{pmatrix} \quad + \quad SIMIL\_L_{i-1,j} \\ contrib\begin{pmatrix} - \\ X'[i'] \end{pmatrix} \quad + \quad SIMIL\_L_{i,j-1} \end{cases}$$

Equation 4-1

Ainsi SIMIL_L peut être remplie progressivement en partant du coin "supérieur gauche" 0,0 et en arrivant au coin "inférieur droit" n,n'.

De même que pour les autres modes de comparaison dans le cadre du modèle d'édition, l'obtention de la/des paire(s) de facteurs réalisant la maximisation nécessite des opérations supplémentaires (méthode des pointeurs par exemple), cf. 5.5.1. Les alignements correspondant aux paires de facteurs optimales peuvent être obtenus en même temps que celles-ci.

4.5.2.3. Exemple

Soient à comparer localement les deux passages mélodiques présentés respectivement à la Figure 4-8 et à la Figure 4-9,et désignées par Mél1 et Mél2. Rappelons que l'égalité de longueur des séquences, avérée ici, n'est nullement nécessaire pour pouvoir effectuer leur comparaison locale.

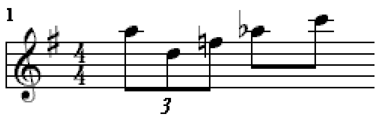


Figure 4-8. Passage mélodique Mél1

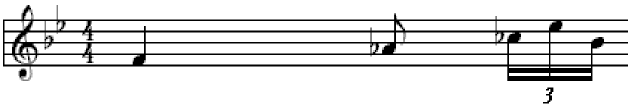


Figure 4-9. Passage mélodique Mél2

Nous nous plaçons dans le cadre d'un modèle d'édition standard, valué et à contributions variables. Les fonctions de contribution pour les trois types d'appariement sont celles proposées par Mongeau et Sankoff (1990) et dérivées directement de celles que nous avons détaillées au 3.2.5.2.2.

Les contributions des différents appariements qui interviennent sont présentées au Tableau 4-1, que nous désignons par CONTRIB et sur lequel Mél1 est « disposée » verticalement et Mél2 horizontalement. Les colonnes et les lignes sont, pour des raisons pratiques, numérotées **de -1 à 5** :

- La ligne 0, titrée « λ » (symbole qui — rappel — désigne la séquence vide), indique les contributions des différentes **insertions** : par exemple, la valeur de CONTRIB[0,3] est la contribution de l'insertion de la 3e note (Cb3) de mél2.
- La colonne 0, également titrée « λ », indique les contributions des différentes **délétions** : par exemple, la valeur de CONTRIB[2,0], soit -0.2, est la contribution de la délétion de la seconde note (D4) de mél1.

Les autres éléments de la matrice contiennent les contributions des différents remplacements. Ainsi le remplacement de la note Ab4 par Bb3 a une contribution de -4.2, inscrite à la ligne 4 et la colonne 5.

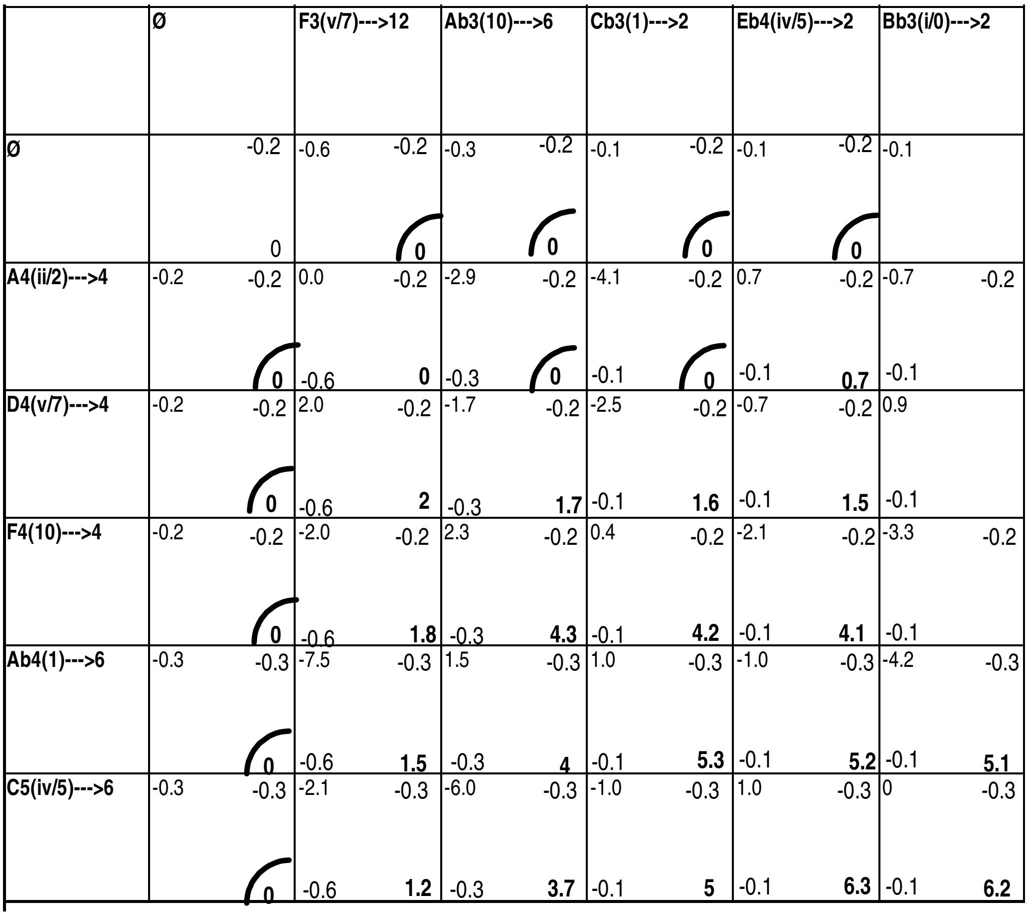
Tableau 4-1. Contributions respectives des différents appariements

		F3	Ab3	Cb3	Eb4	Bb3
	0	-0.6	-0.3	-0.1	-0.1	-0.1
A4	-0.2	0.0	-2.9	-4.1	0.7	-0.7
D4	-0.2	2.0	-1.7	-2.5	-0.7	0.9
F4	-0.2	-2.0	2.3	0.4	-2.1	-3.3
Ab4	-0.3	-7.5	1.5	1.0	-1.0	-4.2
C5	-0.3	-2.1	-6.0	-1.0	1.0	0.0

La matrice SIMIL_L, entièrement remplie, est illustrée à la Figure 4-10. Ses valeurs à proprement parler figurent **en gras** dans le coin **inférieur droit** de chaque case. Néanmoins, pour refléter le mécanisme de calcul de chacun des éléments de SIMIL_L (Equation 4-1), nous faisons également figurer les contributions des différents appariements :

- celles des remplacements sont portées dans le coin supérieur gauche de chaque case,
- celles des insertions, dans le coin supérieur droit de chaque case, enfin :
- celles des délétions, dans le coin inférieur gauche

N.B. : Les notes respectives des deux passages sont désignés par leur hauteur absolue en notation anglo-saxonne (par exemple : « F3 »). Les chiffres romains et arabes indiqués à côté de celle-ci, qui se rapportent respectivement à la hauteur diatonique (degré dans les tonalités globales respectives) et à la durée de chaque note, sont juste fournis à titre d'information supplémentaire.



	Ø	**F3(v/7)--->12**	**Ab3(10)--->6**	**Cb3(1)--->2**	**Eb4(iv/5)--->2**	**Bb3(i/0)--->2**
Ø	-0.2 0	-0.6 -0.2 **0**	-0.3 -0.2 **0**	-0.1 -0.2 **0**	-0.1 -0.2 **0**	-0.1
A4(ii/2)--->4	-0.2 -0.2 **0**	0.0 -0.2 -0.6 **0**	-2.9 -0.2 -0.3 **0**	-4.1 -0.2 -0.1 **0**	0.7 -0.2 -0.1 **0.7**	-0.7 -0.2 -0.1
D4(v/7)--->4	-0.2 -0.2 **0**	2.0 -0.2 -0.6 **2**	-1.7 -0.2 -0.3 **1.7**	-2.5 -0.2 -0.1 **1.6**	-0.7 -0.2 -0.1 **1.5**	0.9 -0.1
F4(10)--->4	-0.2 -0.2 **0**	-2.0 -0.2 -0.6 **1.8**	2.3 -0.2 -0.3 **4.3**	0.4 -0.2 -0.1 **4.2**	-2.1 -0.2 -0.1 **4.1**	-3.3 -0.2 -0.1
Ab4(1)--->6	-0.3 -0.3 **0**	-7.5 -0.3 -0.6 **1.5**	1.5 -0.3 -0.3 **4**	1.0 -0.3 -0.1 **5.3**	-1.0 -0.3 -0.1 **5.2**	-4.2 -0.3 -0.1 **5.1**
C5(iv/5)--->6	-0.3 -0.3 **0**	-2.1 -0.3 -0.6 **1.2**	-6.0 -0.3 -0.3 **3.7**	-1.0 -0.3 -0.1 **5**	1.0 -0.3 -0.1 **6.3**	0 -0.3 -0.1 **6.2**

Figure 4-10. Visualisation de la matrice SIMIL_L, “enrichie” avec les valeurs des contributions d’appariements propres (remplacements) et impropres (indels).

La **valeur** du **meilleur alignement local** entre Mél1 et Mél2 est donnée par le plus grand élément de SIMIL_L, soit **6.3**. La détermination, à proprement parler, de cet alignement optimal, et de la paire de facteurs correspondants, sera traitée comme suite du présent exemple au 5.5.1.2, page 248.

4.5.3. Algorithme de localisation mis en oeuvre

Replaçons-nous à présent dans le cadre du problème LOC défini au 4.5.1. Pour localiser **toutes** les occurrences approximatives d'une qualité supérieure à un Seuil fixé, il nous faut rechercher non le meilleur alignement local entre un pattern et une séquence mais **les** meilleurs alignements. Pour cela, nous mettons en œuvre une approche itérative prenant explicitement en compte l’évolution de la partie de chaque séquence dans laquelle les occurrences sont successivement recherchées. Cette évolution est un « amoindrissement » — schématiquement, et en considérant pour la clarté d’exposition que N=M=1 : à chaque fois qu’une occurrence du pattern P est trouvée dans la séquence X, le segment correspondant de X (facteur destination de l’alignement local optimal définissant l'occurrence) peut être « enlevé » de X au sens où l'on considère qu'aucune nouvelle occurrence de P ne pourra être trouvée dans ce segment. La méthode est formalisée ci-après.

Indiquons que d'autres approches itératives ont été proposées [Waterman & Eggert 1987 ; Kruskal & Sankoff 1983], qui consistent en des manipulations sur la matrice SIMIL_L (cf. 4.5.2.2) directement. Ces approches sont, dans certaines conditions, potentiellement plus efficaces que l'algorithme mis en oeuvre ici du strict point de vue de la complexité algorithmique, quoiqu'un recalcul d'une partie de SIMIL_L soit nécessaire à chaque itération (de l'ordre du quart du nombre d'éléments de la matrice en moyenne dans le cas où ETAA={Remplacement, MultiInsertion, MultiDélétion}) [Waterman 1995]. Cela dit les temps de calcul atteints en pratique (notamment dans le cadre des expérimentations présentées au 4.6) restent de toute manière modérés.

Le principe mis en œuvre est plus simple à décrire pour le cas d'un seul pattern P et d'une seule séquence S (M=N=1), la généralisation à des nombres quelconques de patterns et de séquences ne présentant aucune difficulté particulière. Cette généralisation consiste tout simplement à rechercher successivement et séparément chacun des M patterns dans chacune des N séquences. Cela peut conduire à trouver des occurrences de deux patterns différents qui se recoupent. Si cette caractéristique doit être évitée dans certains domaines applicatifs (dans lesquels aucun facteur ne peut raisonnablement appartenir simultanément aux occurrences de 2 patterns différents), elle est au contraire adéquate voire nécessaire dans d'autres. En particulier, sur la localisation de patterns mélodiques ce genre de recoupement est fréquent, comme en attestent par exemple les annotations d'Owens [Vol. I, p. 31].

Le principe de l'algorithme est illustré en Figure 4-11, où un pattern P est localisé dans la u-ième séquence Su d'un corpus multiple C. Plutôt que de rechercher des alignements loco-globaux (cf. 3.2.4.2.7) comme dans la comparaison loco-globale, nous recherchons des alignements **locaux** d'une valeur suffisante. En d'autres termes, un nombre Seuil ayant été fixé, un facteur F du corpus est une occurrence du pattern P si et seulement si la valeur du meilleur alignement local entre F est P est supérieure ou égale à Seuil. (Auquel cas, la valeur de cet alignement optimal soit Simil(F,P) fournit une évaluation de la "qualité" de l'occurrence trouvée). L'algorithme employé pour la comparaison locale est celui présenté au 4.5.2.

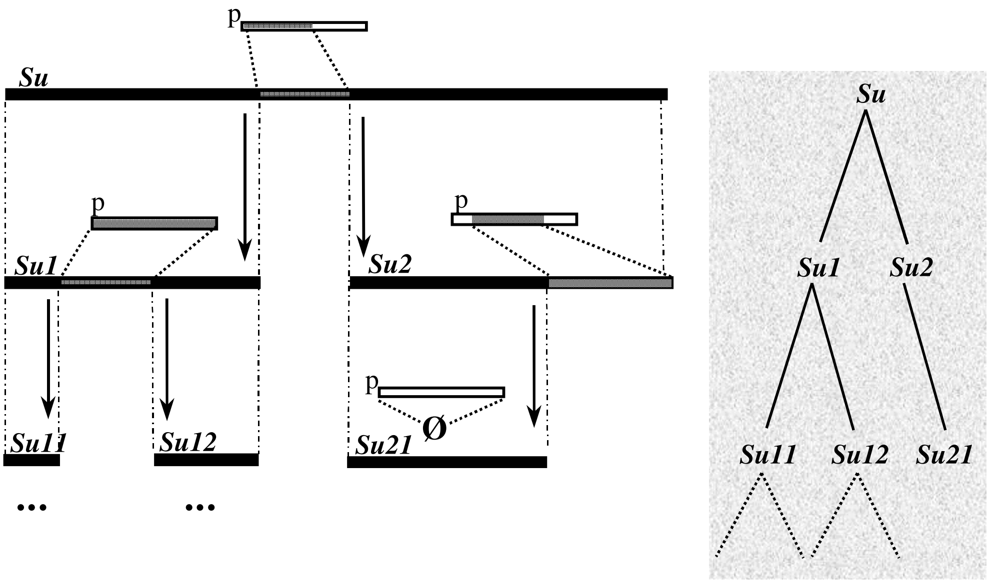


Figure 4-11. Représentation graphique de l'algorithme de recherche d'occurences de pattern mis en oeuvre.

Un processus itératif est mis en oeuvre. A chaque itération on recherche — via une comparaison locale — une occurrence de P dans l'une des séquences partielles restant à traiter. Notons ESP_k l'ensemble des séquences partielles restant à traiter au début de la k-ième itération; et SPk la séquence partielle que l'on choisit de traiter lors de cette itération. On a initialement ESP1={Su} et, bien entendu, SP1=Su. Quatre cas (identifiés par des caractères gras ci-dessous) peuvent se présenter pour le résultat de la recherche de P dans SPk, :

- **Cas 1** : aucune occurrence n'est trouvée (c'est le cas pour la séquence partielle Su21 sur la figure). SPk est alors tout simplement éliminée de ESPk.

- **Cas 2** : une occurrence est trouvée. Il y a alors 3 sous-cas concrets, discutés ci-après par ordre décroissant de fréquence :

  - **Cas 2.1** : l'occurrence n'est ni un préfixe, ni un suffixe de SPk, c'est-à-dire qu'elle se trouve quelque part "au milieu" de la séquence partielle, comme pour Su1 dans la Figure 4-11. Cela conduit à l'ajout, dans ESPk+1, des deux séquences partielles "gauche" et "droite" (Su11 et Su12), et à l'enlèvement de SPk.

  - **Cas 2.2** : l'occurrence est un suffixe (resp. un préfixe) de SPk — cf. l'exemple de Su2. Alors la séquence partielle initiale (resp. la séquence partielle terminale) est ajoutée dans ESPk+1, et SPk en est retirée.

- **Cas 2.3** : l'occurrence est SPk elle-même (cas non illustré en Figure 4-11). Dans ce cas très particulier, on se contente de retirer SPk de ESPk+1.

Il est naturel de représenter l'ensemble des séquences partielles successives par un arbre dont chaque nœud représente une séquence partielle ; Dans la partie droite de la Figure 4-11 est représenté l'arbre des séquences tronquées successives (fils gauche : préfixe, fils droit : suffixe) dans lesquelles la recherche d'occurrences approximatives du pattern p est faite. Pour désigner chaque nœud, on emploie tout simplement la désignation de la séquence partielle qu'il représente. Au début de l'itération k, ESPk est représenté par l'ensemble des feuilles de l'arbre qui n'ont pas encore été traitées. Les actions entreprises dans chacun des 4 cas (cf. ci-dessus) ont une traduction en terme de développement de l'arbre, comme présenté sur le Tableau 4-2. Pour chacun, une feuille (et une seule) est traitée, conduisant soit à la création de fils (1 ou 2), soit à l'étiquetage de la feuille comme ayant été traitée. L'algorithme s'arrête lorsque l'on obtient un arbre dont toutes les feuilles ont été [étiquetées comme déjà] traitées.

Tableau 4-2. Traduction des actions effectuées sur l'ensemble ESPk en opérations de développement de l'arbre des séquences partielles.

Numéro du cas	Actions	Traduction en termes de développement de l'arbre
1	SPk est retirée de ESPk+1	Etiquetage de la feuille SPk comme "déjà traitée"
2.1	Ajout dans ESPk+1 des deux séquences partielles "gauche" et "droite" Enlèvement de SPk.	Création de deux fils (feuilles) sous SPk
2.2	Ajout dans ESPk+1 de la séquence partielle terminale (resp. initiale) Enlèvement de SPk	Création d'un fils unique (feuille) sous SPk
2.3	SPk est retirée de ESPk+1	Etiquetage de la feuille SPk comme "déjà traitée"

4.5.3.1. Structure de contrôle

4.5.3.1.1. Choix de la prochaine séquence partielle à traiter

Nous avons défini plusieurs critères pouvant être employés alternativement ou conjointement pour décider, à chaque itération, quelle séquence partielle de SPk doit être la prochaine traitée. Citons en deux principaux:

- Critère de qualité d'occurrence: la prochaine séquence partielle traitée est celle (ou l'une de celles), parmi celles restant à traiter, qui résulte(nt) de la localisation de la meilleure occurrence parmi celles qui viennent d'être trouvées.

- Critère de longueur: la prochaine séquence partielle traitée est la plus longue restante.

Par 'emploi conjoint de deux critères' nous signifions l'emploi d'un critère "majeur" C1 et d'un critère "mineur" C2, le second permettant de départager deux séquences partielles lorsque celles-ci sont ex-aequo au regard de C1. En pratique, un choix judicieux est d'utiliser conjointement le critère de qualité d'occurrence (critère majeur) et celui de longueur (critère mineur).

4.5.3.1.2. Critères d'arrêt

Le critère d'arrêt est que, à l'issue de l'itération k, ESP_k soit vide. Des stratégies d'élagage de l'arbre des séquences partielles peuvent être proposés, qui suivant le paramétrage au sens large (c'est-à-dire l'ensemble des caractéristiques du modèle de comparaison : ensemble des types d'appariements autorisés et fonctions de contribution) sont on non strictement heuristiques[19]. Par exemple, on peut choisir d'éliminer que toute séquence peut être éliminée (et par suite, le sous-arbre qui en descend, élagué) dès que sa longueur devient inférieure à un seuil. Ce seuil peut avantageusement être défini en fonction du pattern P en cours de traitement : on peut tout simplement choisir comme seuil la longueur de P (ou sa durée dans le cas de séquences temporelle). Cela revient à considérer qu'il est inutile — ou inopportun[20], suivant les cas — de rechercher une occurrence de P dans une (portion de) séquence plus courte que lui.

4.6. Résultats expérimentaux

Le but des expérimentations présentées dans cette section était d'évaluer empiriquement l'influence d'un certain nombre de facteurs dans le cadre de la localisation de patterns connus à l'avance. Ainsi ce qui va être présenté quantifie l'influence, sur la qualité des résultats obtenus :

- Du jeu de descriptions employé, et de la pondération des différentes descriptions

---

[19] C'est-à-dire pouvant avoir une influence négative sur la qualité des résultats [Shapiro 1992].

[20] Notons que, comme les alignements recherchés ici sont des alignements **locaux**, il n'est pas impossible de trouver une occurrence de P dans une séquence plus courte que lui en nombre d'éléments ou en durée (séquences temporelles).

- Du mode de comparaison entre notes (ou groupes de notes), c'est-à-dire de la manière dont les descripteurs sont effectivement, quantitativement, pris en compte.

- De la structure de correspondance (fixe, variable) intervenant dans la comparaison de facteurs (passages mélodiques)

En ce qui concerne les deux premiers items, soulignons schématiquement que même si l'on sait, au plan qualitatif, que toutes les descriptions introduites ici (durée, intervalle chromatique formé entre chaque note et son successeur/prédécesseur, position métrique, extrema locaux de hauteur) peuvent jouer un rôle certain dans le jugement de similitude entre passages mélodiques, en revanche les modalités (quantitatives) de cette intervention sont largement inconnues. Les expérimentations telles que celles rapportées ici vont dans le sens d'une quantification empirique de ces modalités.

L'évaluation est permise par la comparaison des résultats d'Imprology avec ceux obtenus par Owens (1974) sur le même matériau expérimental, résultats qui sont pris comme référence.

Remarque : il est possible de discuter la parfaite validité des résultats pris ici comme référence, ceux-ci étant notamment, par nature, marqués d'une certaine subjectivité. Cela dit l'analyse musicale n'est pas, et n'a jamais eu la vocation d'être, une science exacte. Au-delà, exiger l'objectivité des résultats de référence signifierait l'abolition de toute expérimentation quantitative (et même peut-être qualitative) dans tout un pan de l'informatique musicale.

Nous allons présenter les résultats obtenus dans le cadre de la localisation d'une (grande) partie des patterns trouvés par Owens, dans un sous-ensemble des solos transcrits par celui-ci. Plus précisément (voir Figure 4-12) :

- La collection des patterns localisés est formée de 142 formules issues du lexique d'Owens, soit environ les trois-quarts, en nombre de patterns, de ce lexique.

- Le (sous-)corpus de monodies dans lesquelles est effectuée la localisation regroupe l'ensemble des 17 solos de la catégorie « do majeur Blues » transcrits par Owens[21] (Volume II). La tonalité de do majeur est celle pour laquelle Owens

---

[21] En réalité, cette catégorie en compte 18. Cependant le dix-huitième, « Funky Blues », fait partie des quelques transcriptions d'Owens qui — en raison notamment de la mauvaise qualité des enregistrements sonores dont elles sont issues (milieu du siècle), sont trop imprécises pour pouvoir être saisies comme

propose le plus grand nombre de transcriptions de solos improvisés sur des Blues. Correspondant au la majeur sur le saxophone alto (instrument en mi bémol), cette tonalité est l'une des favorites de Parker si l'on en croit les nombres respectifs de morceaux dans les différentes tonalités [Owens 1974] ; quand au type de grille d'accords, le Blues est de loin celui que Bird utilisait le plus souvent. Les 17 transcriptions totalisent 764 mesures et 4480 notes, le tempo moyen étant de 196 (pulsations par minute). Le détail est donné au Tableau 4-3.

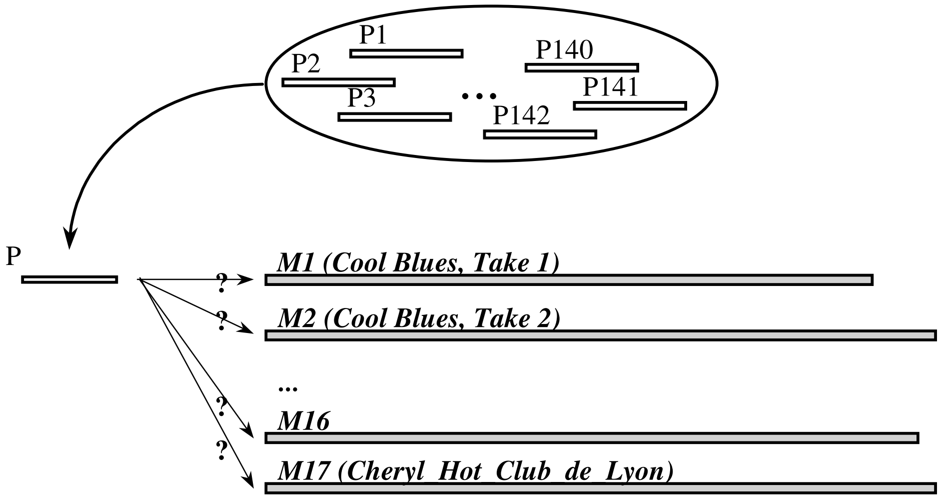


Figure 4-12. Localisation : schéma expérimental

Dans un premier temps, nous avons effectué un comptage des occurrences exactes des différents patterns au sein des différents solos (section 4.6.1). Puis nous avons enrichi ces résultats en effectuant cette fois-ci une localisation des occurrences approximatives de patterns, et en confrontant les lieux de ces occurrences avec ceux que nous avons pu extraire de l'analyse d'Owens (section 4.6.1.4).

---

séquences MIDI : en différents endroits, soit le rythme, soit les hauteurs sont indéfinis (soit encore les deux !).

Tableau 4-3. Caractéristiques quantitatives générales des solos de la catégorie “do majeur Blues” transcrits par Owens [1974, Vol II]

Titre du morceau	Sous-titre éventuel	Numéros de pages [Owens 1974, vol. II]	Tempo	Nombre de mesures	Nombre de notes
Cool blues	Take 1	342	260	38	203
Cool blues	Take 2	343	220	38	224
Cool blues	Take 3	344	165	40	208
Cool blues	Take 4	345	180	39	233
Relaxin' at Camarillo	Take 1	346	190	26	134
Relaxin' at Camarillo	Take 3	346	195	26	129
Relaxin' at Camarillo	Take 4	347	190	26	138
Relaxin' at Camarillo	Take 5	348	205	25	130
Cheryl	Take 2	349	185	37	192
Perhaps	Take 1	350	200	38	226
Perhaps	Take 3	351	190	38	247
Perhaps	Take 4	352	185	13	81
Perhaps	Take 5	353	195	38	233
Perhaps	Take 6	354	195	38	238
Cool blues 1949 Tape		355	205	157	905
Cheryl Hot Club Lyon		359	180	85	583
Cool Blues Sonet slp 27		362	185	62	376

4.6.1. Dénombrements d’occurrences exactes

Les conditions expérimentales étaient les suivantes.

- Au plan des jeux de descriptions, dans une première expérimentation (4.6.1.1) les notes étaient décrites uniquement par **l’intervalle chromatique** formé avec la note précédente (représenté par un nombre algébrique de demi-tons). Dans la seconde (4.6.1.2), on a adjoint, pour chaque note, sa **durée relative**, en nombre — rationnel — de noires. Enfin dans la troisième, est encore ajoutée la **position métrique** de chaque note dans la mesure (4.6.1.3). Cette position traduit l’instant de début de la note (en temps relatif) modulo k, où k est donné par la métrique du morceau : ici, k=4 (métrique 4/4). Soulignons que ces descriptions n'existent pas au départ, elles sont ajoutées au cours d'une étape de prétraitement des séquences mélodiques — solos aussi bien que patterns (étape d'enrichissement des données / représentations). En cela, nous anticipons ici sur la présentation de notre modèle d'édition valué multi-descriptions (section 4.7).

- Au plan du modèle de comparaison du critère d'équipollence et du paradigme de localisation : il s'agit de recherche d'occurrences exactes. Un passage F (de longueur m) est donc localisé comme occurrence d'un pattern P (passage de même longueur m) si et seulement si les deux passages sont égaux, autrement dit si :

$$\{\forall i \in \mathbf{N}m\} \; \{F[i]=P[i]\},$$

Equation 4-2

l'égalité entre deux éléments (notes) signifiant l'égalité sur chacun des descripteurs : intervalle chromatique, et aussi, le cas échéant, durée relative et/ou position métrique.

L'algorithme de localisation employé est celui présenté au 4.5.3. Les conditions particulières de localisation décrites ci-dessus s'obtiennent très facilement en réglant, dans Imprology, les paramètres (au sens large) de l'algorithme à des valeurs adéquates :

- ETAA = {Remplacement}
- Fonction de contribution associée au type d'appariement Remplacement : elle est de nature pseudo-booléenne (cf. 4.7.4.3), utilisant deux constantes, ConstId = 1 et ConstDiff = 0 (par exemple ; voir aussi note n°22 de bas de page), qui représentent respectivement :
    - la contribution de toute identité
    - la contribution de toute substitution
- Critère d'équipollence[22] (en utilisant les mêmes identificateurs qu'à l'Equation 4-2) :

$$Eq(F,P) \Leftrightarrow Simil(F,P)=m$$

---

[22] En fixant, alternativement, ConstId=0 et ConstDiff=-1, le critère d'équipollence devient : $\{Eq(F,P) \Leftrightarrow Simil(F,P)=0\}$.

Notons que, s'agissant de localisation d'occurrences exactes, nous aurions pu alternativement mettre en œuvre un algorithme spécialisé du type de ceux présentés au 3.3.1. Cependant, cela n'obéissait à aucune nécessité dans le cas de ces expérimentations.

4.6.1.1. Intervalle chromatique seul

Le premier paramétrage consiste à ne prendre en compte, comme description de chaque note, que l'intervalle chromatique formé avec la note précédente. Rappelons (cf. 3.4.2.1.1) qu'il s'agit là de l'un des modes de description utilisés dans le système EMI.

Improlog localise au total **2328** occurrences, le détail solo par solo étant donné dans le Tableau 4-7. En termes de moyennes, cela signifie 16.4 occurrences par pattern dans l'ensemble des morceaux, et 137 occurrences par morceau pour l'ensemble des patterns. Les écarts-types correspondants sont respectivement égaux à 52 et 100, des valeurs élevées. Cela traduit d'un côté les disparités entre patterns en matière de « rareté » (laquelle est généralement proportionnelle à leur longueur), de l'autre les disparités de longueur entre les différentes transcriptions (par exemple, 157 mesures pour Cool blues 1949 Tape contre seulement 13 pour Perhaps Take 4 ; cf. Tableau 4-3).

Il s'agit à présent de comparer les résultats obtenus par Improlog avec ceux indiqués par Owens (Volume I). Cela serait fort simple si nous disposions d'un tableau indiquant, pour chaque pattern du lexique, son nombre d'occurrences dans chacun des solos. Or la réalité diffère à deux titres :

- D'une part, la manière dont le musicologue présente les nombres d'occurrences est fondée sur la structure hiérarchique du lexique. Ainsi, par exemple, les nombres d'occurrences, dans les solos de la catégorie « do majeur Blues », des différents patterns (feuilles) descendant du nœud interne étiqueté « 1 » (voir Figure 4-1) ne sont pas détaillés. Au contraire, Owens indique collectivement :
    - 105 occurrences pour les différents patterns descendants du nœud 1A (soit 13 patterns différents)
    - 37 occurrences pour les descendants de 1B (4 patterns différents)
    - 3 occurrences pour les descendants de 1C (2 patterns différents)
- D'autre part, tous les patterns (feuilles) ne sont pas couverts. Par exemple, Owens n'indique pas le nombre d'occurrences du pattern 2B (voir Figure 4-2). Celui-ci

est une feuille, mais ce genre d'information manquante se retrouve aussi pour des nœuds internes.

Remarque : si cela n'était pas totalement évident à l'issue ce qui vient d'être dit, précisons que — à l'exception de 4 parties de morceaux (représentant à peine 4 pages de partition en tout [Vol. I, pp. 31-34]) — aucune des transcriptions d'Owens n'est fournie *annotée*. Cet adjectif désigne le fait que les différentes occurrences de patterns soient signalées sur la partition, par exemple de la manière que nous avons adoptée (voir Figure 4-13).

Ainsi, pour comparer les nombres d'occurrences trouvés respectivement par Imprology et par Owens, nous avons mis en œuvre le mécanisme suivant :

- Parcourir l'ensemble des nœuds internes pris en considération dans les décomptes d'Owens
- Pour chacun de ces nœuds internes :
    - Parcourir l'ensemble des descendants terminaux (patterns) et sommer les nombres d'occurrences de tous ces patterns
    - Comparer cette somme avec le dénombrement indiqué par Owens.

Bien entendu, compte tenu de ce qui a été dit plus haut, cela conduit à des nombres totaux d'occurrences par solo différents de ceux du Tableau 4-7. Ces nombres sont, respectivement, 1785 contre 1005 pour Owens. Cela traduit globalement le fait que, avec ce paramétrage, Imprology est *trop sensible* sur la localisation, du moins peut-on l'affirmer sur l'ensemble de données dont il est question (une discussion là-dessus sera proposée au 4.6.1.4, à la lumière des autres résultats expérimentaux qui seront présentés dans les deux sections qui suivent). Le Tableau 4-4 donne le détail du nombre de nœuds internes pour lesquels les résultats d'Imprology sont corrects ou au contraire entachés, respectivement, d'un excès de sensibilité ou d'un manque de sensibilité. Pour 29 nœuds internes, le nombre d'occurrences trouvés par Imprology est conforme aux chiffres d'Owens. Cela dit il faut bien distinguer, parmi ces 29 nœuds internes, ceux pour lesquels le nombre d'occurrences est nul. Or c'est le cas pour 28 d'entre eux, le dernier représente à peine deux occurrences. Pour 13 nœuds internes, le nombre d'occurrences localisées par Imprology dépasse strictement celui d'Owens. Le total des écarts est de 1011 occurrences. A l'opposé, sur 24 nœuds internes la localisation faite par Imprology est — du point de vue du nombre d'occurrences — moins sensible que celle faite par Owens. Cela étant le cumul des écarts n'est que de 231 occurrences.

Tableau 4-4. Résultats obtenus en termes de sensibilité avec "intervalles chromatiques seulement"

	Correct		"Ultrasensible"	"Infrasensible"
	Zéro occurrences	*1 occurrence au moins*		
Nombre de nœuds internes	28	1	13	24
Différence entre les nombres d'occurrences (Imprology moins Owens)	(0)	(0)	1011	231

4.6.1.2. Intervalle chromatique et durée

Cette fois-ci, chaque note est décrite par l'intervalle chromatique formé avec la note précédente et par sa durée, exprimée en temps relatif. Là aussi, nous avons vu au 3.4.2.1.1 qu'il s'agit d'un des jeux de descriptions employés dans EMI. Même si l'on sait qu'il existe des corrélations entre différentes dimensions musicales — en particulier les deux qui interviennent ici — il fallait s'attendre à une baisse de la sensibilité d'Imprology.

Le nombre total d'occurrences localisées par Imprology est égal à **469**, le détail solo par solo étant donné dans le Tableau 4-7.

En nous plaçant à présent dans le contexte d'une comparaison avec les dénombrements d'Owens, on obtient le Tableau 4-5. Le mécanisme décrit en section précédente est à nouveau utilisé ; le tableau indique que les seuls nœuds internes de la hiérarchie du lexique pour lesquels le nombre d'occurrences trouvé est conforme aux dénombrements d'Owens (il y en a 30) concernent des patterns sans occurrences. Cette fois-ci, le nombre total d'occurrences trouvées par Imprology qui entrent en ligne de compte est de 459, à comparer toujours avec 1005 (Owens). Cela traduit globalement une *sensibilité insuffisante*. Les grandeurs respectives des nombres d'occurrences intitulés « ultrasensibles » et « infrasensibles » sont à peu près inversées par rapport à celles que l'on avait pour la description « intervalle chromatique » seule.

Tableau 4-5. Résultats obtenus en termes de sensibilité avec "intervalles chromatiques et durées"

	Correct		**“Ultrasensible”**	**“Infrasensible”**
	Zéro occurrences	***1 occurrence au moins***		
Nombre de nœuds internes	30	0	6	30
Différence entre les nombres d'occurrences (Imprology moins Owens)	(0)	(0)	85	603

4.6.1.3. Intervalle chromatique, durée et position métrique

Cette fois-ci, Imprology localise au total 95 occurrences, réparties comme le montre le Tableau 4-7. Le Tableau 4-6 résume la comparaison faite avec les dénombrements d’Owens. « L’infrasensibilité » est encore accentuée par rapport à ce qu’elle était avec le jeu de descriptions « intervalle chromatique et durée ».

Tableau 4-6. Résultats obtenus en termes de sensibilité avec "intervalles chromatiques, durées et métrique"

	Correct		**“Ultrasensible”**	**“Infrasensible”**
	Zéro occurrences	***1 occurrence au moins***		
Nombre de nœuds internes	31	0	1	34
Différence entre les nombres d'occurrences (Imprology moins Owens)	(0)	(0)	1	909

4.6.1.4. Discussion

Le Tableau 4-7 synthétise les résultats obtenus avec les trois jeux de descriptions (nombres totaux d’occurrences, tous patterns confondus).

Tableau 4-7. Comparaison des résultats obtenus avec "intervalles chromatiques seulement", avec "intervalles chromatiques et durées" et avec "intervalles chromatiques, durées et métrique"

	Cheryl_Hot_Club_Lyon	Cheryl_Take_2	Cool_blues_1949_Tape	Cool_Blues_Sonet_slp_27	Cool_Blues_Take1	Cool_Blues_Take2	Cool_Blues_Take3	Cool_Blues_Take4	Perhaps_Take1	Perhaps_Take3	Perhaps_Take4	Perhaps_Take5	Perhaps_Take6	Relaxin_at_Camarillo_Take1	Relaxin_at_Camarillo_Take3	Relaxin_at_Camarillo_Take4	Relaxin_at_Camarillo_Take5	**Total**
Intervalles seulement	211	87	358	113	91	81	46	64	95	108	40	97	121	62	48	71	59	**2328**
Intervalles + durées	36	19	94	34	26	26	11	13	38	24	8	27	35	10	13	17	16	**469**
Intervalles + durées + positions métriques	10	0	20	6	5	8	0	2	7	5	4	7	7	3	2	5	4	**95**

Nous avons plusieurs fois souligné que le modèle de comparaison d’entités séquentielles intervenant, explicitement ou implicitement, au sein de tout processus de découverte de régularités, avait un impact déterminant sur la qualité des résultats obtenus. Concernant le modèle employé ici, les résultats expérimentaux obtenus au travers de la localisation de patterns d’Owens dans le corpus permettent de tirer des enseignements s’articulant suivant quatre axes interdépendants :

- Celui du choix des (jeux de) représentations : les résultats obtenus quantifient clairement que le jeu de description réduit à « intervalles chromatiques » conduit à une fonction d’équipollence trop « lâche », trop tolérante. L’ajout de la description « durée » corrige ce biais, et l’ajout supplémentaire de la description « position métrique » va encore au-delà dans ce sens. Toutefois dans les deux cas la situation est **inversée** du point de vue de la fonction d’équipollence, celle-ci devenant au contraire trop stricte. On ne peut tirer des conclusions générales telle que *‘la description « position métrique » ne doit pas être prise en compte dans un modèle de comparaison mélodique puisqu’elle dégrade la validité des résultats par rapport à ceux obtenus avec les descriptions « intervalles chromatiques » et « durées »’* ; en effet comme nous l’avons dit plus haut, aussi bien les trois descriptions dont il est question ici interviennent potentiellement dans le jugement de similitude entre passages mélodiques. Pour interpréter correctement les résultats obtenus, il faut tenir compte d’autres facteurs, qui font l’objet des trois points qui suivent. En clair, le fait que la fonction d'équipollence soit trop stricte ne doit pas être imputé uniquement aux deux jeux de descriptions considérés, mais à d'autres paramètres, au sens large, de l'algorithme et du modèle de comparaison employés.

- Celui du mode de comparaison entre notes : les conditions de localisation dans lesquelles on se place ici peuvent être traduites par des fonctions de contribution binaires. Il apparaît plus adéquat d'utiliser des fonctions **graduelles**, permettant par exemple de considérer quantitativement plus similaires deux notes dont les durées sont presque égales que deux notes de durées très différentes.

- Celui de la pondération des différentes descriptions lorsqu'il y en a plus d'une. Ici, tout se passe comme si on pondérait de manière égale des trois descriptions. Or il est envisageable de considérer par exemple que, dans l'absolu ou en moyenne, la durée joue un rôle plus important que la position métrique du point de vue de la comparaison (évaluation de la similitude). Sans aller jusque là, on peut simplement observer que, suivant le contexte où l'on se place, l'une des trois descriptions (ou toute combinaison de deux) peut jouer un rôle prédominant par rapport aux autres. En particulier, la pondération des différentes descriptions permettant d'obtenir des nombres d'occurrences conformes peut différer d'un pattern à l'autre ; cela traduisant, comme nous l'avons dit précédemment, des points de vue différents sur les critères de similitude / d'équipollence.

- Celui enfin de la structure de correspondance. Le modèle de comparaison sous-jacent ici est un modèle de similitude à structure de correspondance fixe, verticale, (cf. 3.2). « L'infrasensibilité » du système par rapport aux résultats de référence (Owens) est imputable, pour partie, au fait que nombre d'occurrences repérées par Owens ne résultent pas d'une mise en correspondance verticale de deux passages de longueur égale. La nécessité, évoquée plus haut, de tenir compte de changements locaux tels qu'omissions ou ajouts de notes, est mise en évidence expérimentalement ici.

Ces différentes considérations, entre autres, ont influencé notre travail sur la conception d'un modèle de comparaison entre entités séquentielles (4.7). C'est le cas également des considérations issues d'expérimentations sur la localisation d'occurrences approximatives, qui font l'objet de la prochaine section.

4.6.2. Localisation d'occurrences approximatives

Les expérimentations décrites au 4.6.1 ont permis de tirer un certain nombre de conclusions, basées sur la comparaison quantitative de nombres d'occurrences localisées sur le même [sous-]corpus respectivement par Imprology et par Owens. Toutefois les lieux précis des occurrences ne pouvaient entrer en ligne de compte. Les expérimentations présentées dans la présente section visent précisément à prendre en considération ces aspects topologiques, se situant davantage à un plan qualitatif.

Ici encore, il s'agit de comparer les localisations d'occurrences du système avec ceux obtenus par Owens. Les choses sont encore moins simple que pour les dénombrements purs. En effet, comme nous l'avons vu les transcriptions données par Owens ne sont pas annotées, et en dehors des décomptes qu'il fournit — et qui ne nous sont ici d'aucune utilité — le musicologue ne fait que donner (Volume I) des indications imprécises telles que :

« The six-bar segment begining in bar 9 is filled with an extensive development of M24 » (où M24 désigne bien sûr le pattern portant cet identificateur dans le lexique), ou encore :

« M3A appears at bar 10 in all but 2 of the 12 choruses », les 12 choruses en question correspondant à l'ensemble des transcriptions de Cool Blues, *Takes* 1 à 4 (voir Tableau 4-3).

Nous avons donc dû successivement :

- Effectuer un travail manuel d'annotation des transcriptions sur la base des indications imprécises fournies par Owens , puis :
- Proposer un formalisme de représentation des informations de localisation (4.6.2.1), rendant compte notamment de l'incertitude qui caractérise intrinsèquement ces informations et enfin :
- Proposer et implémenter, sur la base de ce formalisme, un schéma de localisation adéquat (4.6.2.3).

4.6.2.1. Annotation des transcriptions

Les deux exemples donnés ci-dessus font apparaître différents types d'incertitude au niveau des indications d'Owens. Premièrement, souvent les informations données se rapporte collectivement à un **groupe** de patterns (nœud interne dans la hiérarchie du lexique) — or nous nous sommes attachés à associer à chaque annotation d'occurrence l'identificateur du **pattern** correspondant (i.e. une **feuille** dans la hiérarchie). Deuxièmement, les nombres d'annotations à porter sur la partition sont souvent elliptiques (« *extensive development* ») ; et, même lorsqu'ils ne le sont pas, des ambigüités subsistent (« all but 2 of the 12 choruses » : encore faut-il trouver les 10 choruses qui nous intéressent !). Troisièmement, aucune localisation précise n'est donnée, ni en ce qui concerne l'instant de début des occurrences que leur durée ou leur instant de fin. Tout au plus sait-on à quelle mesure les occurrences doivent être trouvées. Ajoutons que les indications d'Owens fournissent au mieux un *minorant* des nombres d'occurrences.

Malgré tout cela, nous avons pu annoter les occurrences d'un certain nombre de patterns, nous appuyant à cet effet sur nos propres compétences musicales… La Figure 4-13 montre une vue partielle de transcription avec trois annotations (il s'agit de « laps minimaux », concept qui sera exposé plus bas).

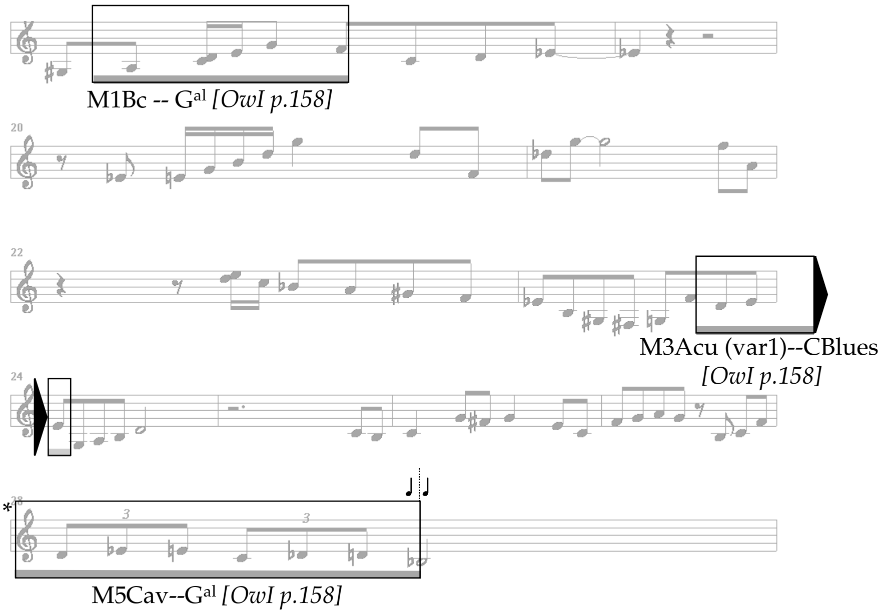



Figure 4-13. Annotation (laps minimaux) d'un passage de "Cool Blues 1949 Tape".

Une annotation correspond à un passage spécifique, qui à son tour peut être caractérisé par un laps de temps relatif (que nous appellons le *support temporel* [*relatif*] de l'annotation ou de l'occurrence). Dans quelques cas, nous avons pu effectuer une annotation sans aucune ambiguïté, au sens où l'occurrence évoquée par Owens pouvait être délimitée de manière univoque (occurrence « exacte », typiquement). Dans de nombreux autres cas en revanche, une telle délimitation unique posait problème. Dans un tel cas, décider de manière unique ce qu'est précisément l'occurrence (c'est-à-dire le laps de temps relatif sur lequel elle se situe) aurait en fait exigé d'adopter un point de vue particulier et subjectif. En d'autres termes, cela nous aurait forcé à poser à l'avance des contraintes fortes sur la manière dont la similitude doit être quantifiée par le système (modèle de comparaison utilisé).

Pour éviter le biais qui résulterait d'une telle méthodologie, nous avons tiré parti du fait que, s'il est souvent impossible de délimiter objectivement **le** laps de temps L(P) qui caractérise une occurrence d'un pattern donné P, en revanche il est presque toujours possible, en pratique, d'encadrer[23] ce laps entre un *laps minimal* Lmin(P) et un *laps maximal* Lmax(P) :

- Lmin(P) correspond à la plus grande partie (connexe) de l'occurrence qui en fasse indéniablement partie. C'est, en termes imagés, « l'intérieur » — au sens de la topologie mathématique — de l'occurrence, ***que*** doit contenir en totalité (ou presque) tout passage candidat au statut d'occurrence de P. Les laps signalés sur la Figure 4-13 sont les laps minimaux associés à chaque occurrence.
- Lmax(P) correspond au contraire au plus petit passage qui indéniablement contient entièrement l'occurrence. C'est, en quelque sorte, « l'adhérence » — au sens de la topologie mathématique — de l'occurrence, ***qui*** doit contenir en totalité (ou presque) tout passage candidat au statut d'occurrence de P.

Ces différentes notions sont illustrées et formalisées à la section suivante.

4.6.2.2. Représentation des informations de localisation

La Figure 4-14 illustre la prise en compte du laps minimal pour une occurrence Occ d'un pattern P. Les deux laps en noir correspondent aux deux passages issus de la comparaison locale par programmation dynamique (meilleur alignement local — cf. 4.5.2).

---

[23] $\mathrm{Lmin}(P) \subset L(P) \subset \mathrm{Lmax}(P)$.

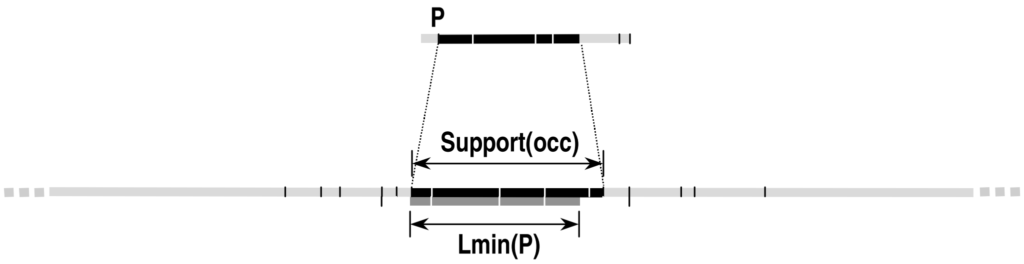


Figure 4-14. Annotation : laps minimal

Pour déterminer si Occ est valide selon le critère « laps minimal », nous pourrions nous contenter d'exiger que le support de Occ contienne entièrement Lmin(P). Il s'avère, en pratique, pertinent de relacher légèrement cette contrainte en introduisant un coefficient réel $\alpha \in [0;1]$ fixant la taille relative **minimum** du recoupement entre Support(Occ) et Lmin(P) pour que Occ soit valide ; lorsque $\alpha = 1$, cela revient à exiger que Support(Occ) contienne entièrement Lmin(P). On pose alors la condition nécessaire et suffisante suivante, portant sur la durée de l'intersection, notée Support(Occ) $\cap$ Lmin(P), entre les laps de temps Support(Occ) et Lmin(P). :

$$\{\text{Occ est une occurrence valide}\} \Leftrightarrow \{\text{Durée}(\text{Support}(\text{Occ}) \cap \text{Lmin}(P)) \geq \alpha \times \text{Durée}(\text{Lmin}(P))\}$$

Equation 4-3

L'existence, dans le système MusES intégré à Imprology (cf. 4.2), de fonctionnalités relatives à la représentation et à l'algèbre des laps de temps (classe Smalltalk Lapse) permet de mettre en oeuvre très aisément le genre de calculs intervenant dans l'Equation 4-3.

La Figure 4-15 illustre la prise en compte du laps minimal pour une occurrence Occ d'un pattern P. Le laps minimal (cf. figure précédente) est rappelé en hachuré. Cette fois-ci encore, on peut assouplir l'exigeance que le support de Occ soit entièrement contenu dans Lmax(P) pour que Occ soit considérée valide selon le critère « laps maximal ». Pour cela, on introduit un coefficient réel $\beta \in [0;1]$ fixant la taille relative **maximum** du recoupement entre Support(Occ) et Lmax(P) pour que Occ soit valide ; lorsque $\beta = 1$, cela revient à exiger que Support(Occ) soit entièrement contenu dasn Lmax(P). On pose alors la condition nécessaire et suffisante suivante, portant sur la durée de l'intersection entre les laps de temps Support(Occ) et Lmax(P). :

$$\{\text{Occ est une occurrence valide}\} \Leftrightarrow \{\text{Durée}(\text{Support}(\text{Occ}) \cap \text{Lmax}(\text{P})) \leq (1/\beta)\times\text{Durée}(\text{Lmax}(\text{P}))\}$$

Equation 4-4

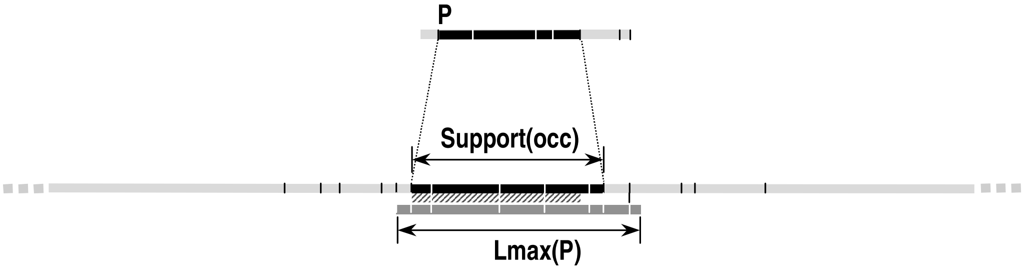


Figure 4-15. Annotation : laps maximal. (Le laps minimal correspondant — figure précédente — apparaît en hachuré)

Ajoutons que les critères laps minimaux et laps maximaux peuvent avantageusement être combinés, fixant des bornes inférieure et supérieure (en termes de laps de temps) pour la validation du support (laps) de toute occurrence.

Enfin, la solution mise en œuvre quant à la représentation effective des annotations (laps minimaux et maximaux) est la suivante. Une annotation est représentée par le nom (ou le numéro d'ordre dans le [sous-]corpus) de la mélodie, son support temporel et d'éventuelles informations supplémentaires. Nous avons notamment souhaité garder une trace de la provenance de chaque annotation en indiquant par exemple dans ce troisième champ les numéros de page du texte d'Owens à l'origine de l'annotation. La Figure 4-16 illustre la forme de la représentation des trois annotations (laps minimaux) de la Figure 4-13. Sur cette dernière :

- Les laps sont décrits en temps relatif de la manière suivante :
    - La première et la 3e donnée sont respectivement l’instant de début et l’instant de fin, sous la forme :

      <numéro de mesure> : <temps relatif dans la mesure (en noires)>

      Remarque : l’écriture de la valeur 1.499 au lieu de 1.5 traduit une simple considération pratique d’affichage qui ne sera pas détaillée ici.
    - La 2e donnée est la durée du laps, en nombre de noires.

- Les numéros de page entre crochets font référence au tome I de l'étude d'Owens.

Cool_blues_1949_Tape	{'16:1.5'--4.0--'17:1.499'}	(source: Owl p158)
Cool_blues_1949_Tape	{'65:1.25'--3.0--'65:4.249'}	(source: Owl p158)
Cool_blues_1949_Tape	{'88:3.0'--3.25--'89:2.249'}	(source: Owl p158)

Figure 4-16. Représentation des annotations

4.6.2.3. Schéma de localisation

L'algorithme présenté au 4.5.3 est ici aussi mis en œuvre. Le critère d'équipollence fait intervenir un seuillage sur la similitude de la paire de passages mélodiques isolée à chaque étape (valeur du meilleur alignement local trouvé à cette étape). Pour chaque occurrence trouvée, les calculs relatifs à la comparaison avec les annotations (laps minimaux / maximaux) sont effectués.

4.6.2.4. Conditions expérimentales

Cette fois-ci, le modèle de similitude employé est un modèle d'édition à proprement parler, avec $ETAA=ETAA_{standard}$. Dans le même état d'esprit que pour les dénombrements d'occurrences présentés plus haut, les jeux de descriptions employés sont progressivement enrichis au fil des expérimentations, jusqu'à atteindre une prise en compte simultanée de 4 descriptions différentes. Parmi celles-ci, une seule (la durée des notes en temps relatif) existe dans la représentation initiale des séquences mélodiques (solos et patterns), toutes les autres étant ajoutées lors de l'étape préliminaire d'enrichissement des données. Ici encore, nous anticipons sur la présentation de notre modèle d'édition valué multi-descriptions (section 4.7). En particulier, la description « extrema local » n'avait à notre connaissance jamais été implémentée auparavant dans un modèle de comparaison quantitative entre entités séquentielles musicales.

Les jeux de descriptions successivement employés sont :

- J1 = {Degré des notes dans la tonalité (globale) du morceau auxquelles elles appartiennent + durée, en temps relatif.}. Ce jeu de description (et les fonctions de contribution associées) est celui proposé par Mongeau et Sankoff (1990), voir présentation au 3.2.5.2.2. Le paramétrage, k1=0.348 et k2=0.05, est exactement celui préconisé par les auteurs[24].

- J2 = {Intervalles chromatiques (« avant », c'est-à-dire formé avec la note suivante, et « arrière », c'est-à-dire formé avec la note précédente) + durée, en temps relatif} (section 4.6.2.6).

- J3 = J2 $\cup$ {extrema locaux de hauteur}. La notion d'extremum local sera explicitée et illustrée dans la section correspondante (4.6.2.7). Il faut ajouter que, par rapport aux deux jeux de descriptions précédents, nous prenons explicitement en compte les silences (ceux-ci seront discutés au 4.7.7.3.3).

La validité de chaque occurrence trouvée est définie par un test de type « laps minimal » avec seuil α=0.75. En revanche, compte tenu des objectifs poursuivis au travers de la présentation des présents résultats expérimentaux, nous faisons figurer sur les graphiques d'occurrence (que nous appelons *profils*) **toutes** les occurrences trouvées qui présentent un recoupement, si mince soit-il, avec les laps minimaux. (Conceptuellement, cela revient à un test « laps minimal » avec α : infiniment petit positif).

N.B. : nous présentons ci-après les résultats de manière plus ou moins « brute », les conclusions étant pour leur part groupées au 4.6.2.8.

4.6.2.5. Jeu de descriptions J1

La partie des résultats présentée ici concerne les patterns 1Bc (Figure 4-17) et 5Cau (Figure 4-18) du lexique d'Owens. Le premier possède 5 notes, ce qui le situe du côté des patterns les plus courts du lexique, et le second 13 notes, ce qui le situe du côté des plus longs.

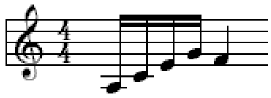


Figure 4-17. Pattern 1Bb du lexique d'Owens

---

[24] Une remarque importante doit être faite : d'un côté, nous avons apporté une amélioration au modèle initial de Mongeau et Sankoff en enrichissant la représentation des durées. Celles-ci y étaient mesurées en doubles croches, rendant l'approche inutilisable dès lors que, comme ici, on a affaire (entre autres) à des durées ternaires (triolets). En revanche, nous avons appauvri l'*ensemble des types d'appariements autorisés* en en retirant les types Consolidation et Fragmentation.

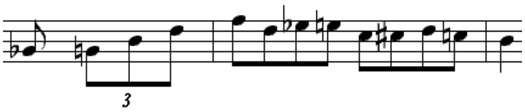



Figure 4-18. Pattern 5Cau du lexique d'Owens

Les résultats correspondent à la localisation d'occurrences au sein de *Cool blues 1949 Tape*, la plus longue des transcriptions « do majeur Blues » (157 mesures, environ 1000 notes). Les annotations de la transcription se répartissent, pour ce qui est des patterns 1Bc et 5Cau, comme suit :

- 1Bc : 12 occurrences, dont :
    - 5 occurrences (que nous désignerons par Occ1 à Occ5) issues du texte d'Owens (Vol. I)
    - 6 occurrences (que nous désignerons par Occ1' à Occ6') localisées par nos soins, manifestement moins bonnes que les 5 précédentes en termes de « qualité » d'occurrence.
    - 1 occurrence (que nous désignerons par Occ'') « découverte » par le système, que nous n'avions pas repérée nous-même.
- 5Cau : 3 occurrences issues du texte d'Owens.

Le profil des occurrences (toutes) du pattern 1Bc obtenues avec le jeu de descriptions J1 est montré sur la Figure 4-19, où l'on a en ordonnée la valeur rattachée à chaque occurrence (valeur d'alignement local). Les 5 occurrences Occ1..Occ5 (traduite chacune par un **carré noir**) sont effectivement localisées, mais — si les 2 premières occupent le premier rang, en revanche les trois suivantes occupent le 34e rang ex-aequo (occurrences 4 à 67). La signification de ces rangs en termes de sensibilité et de sélectivité du système sera explicitée au 4.6.2.6. Pour ce qui concerne les occurrences Occ1' à Occ6' (**losange gris**), une seule est localisée (34e rang ex-aequo). L'occurrence Occ'' n'est pas localisée. Pour le reste, les cercles traduisent des « fausses » occurrences, localisées à tort.

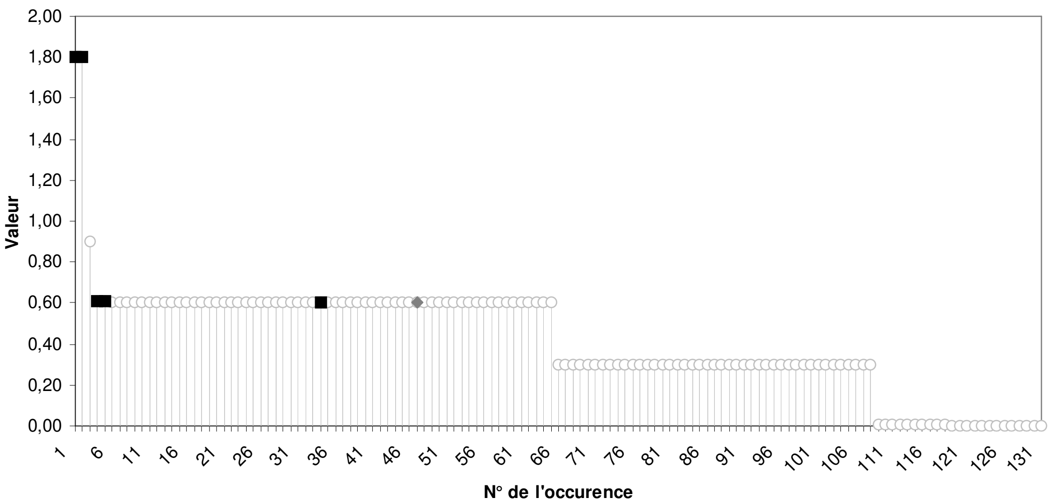


Figure 4-19. Profil des occurrences du pattern 1Bc obtenues avec le jeu de descriptions J1

Le profil des occurrences du pattern 5Cau obtenues avec le jeu de descriptions J1 est montré sur la Figure 4-20. Chaque occurrence est traduite par un **carré noir**. La première occurrence localisée occupe le premier rang, en revanche les deux autres occupent les rangs respectifs 347 et 348, à la droite extrême du graphique.

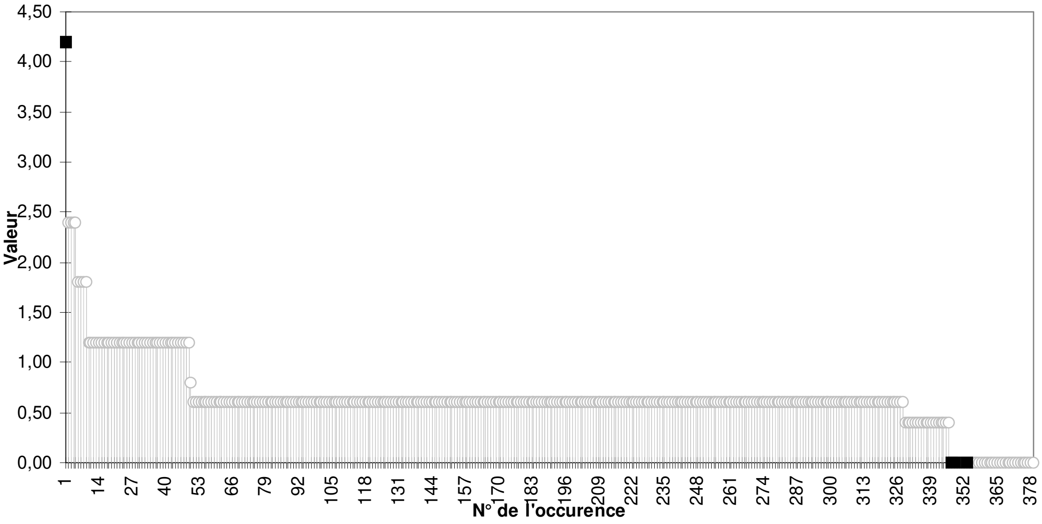


Figure 4-20. Profil des occurrences du pattern 5Cau obtenues avec le jeu de descriptions J1

4.6.2.6. Jeu de descriptions J2

Plutôt que de prendre en compte simplement l'intervalle formé entre chaque note et son prédecesseur (comme au 4.6.1 ou dans les autres travaux cités), nous prenons également en compte ici l'intervalle formé entre chaque note et son **successeur**. La manière dont ces deux descriptions sont combinées au niveau des fonctions de contribution sera détaillée plus loin (5.5.2.2.3).

Le profil des occurrences (toutes) du pattern 1Bc obtenues avec le jeu de descriptions J2 est montré sur la Figure 4-21. (Remarquons qu'en l'absence de normalisation sur les valeurs renvoyées par les fonctions de contribution, la différence observée, par rapport aux profils précédents, au plan de l'échelle des ordonnées n'est pas anormale). L'observation principale pouvant être faite concerne la morphologie du profil obtenu, par comparaison avec celui présenté en Figure 4-20. Le nouveau profil présente un pic initial très « dense », au sens où :

- les 8 premières valeurs sont des occurrences valides —schématiquement, Occ1 à Occ4, puis Occ'', puis Occ1' à Occ3'
- **Toutes** les 12 occurrences sont effectivement localisées (y compris Occ'', traduite par un **triangle gris**), et cela, parmi les 43 premiers rangs.

Par contraste, le pic initial du profil précédent (rangs 1 à 3) ne contenait que deux occurrences valides, sans oublier que la moitié des occurrences n'étaient pas localisées du tout.

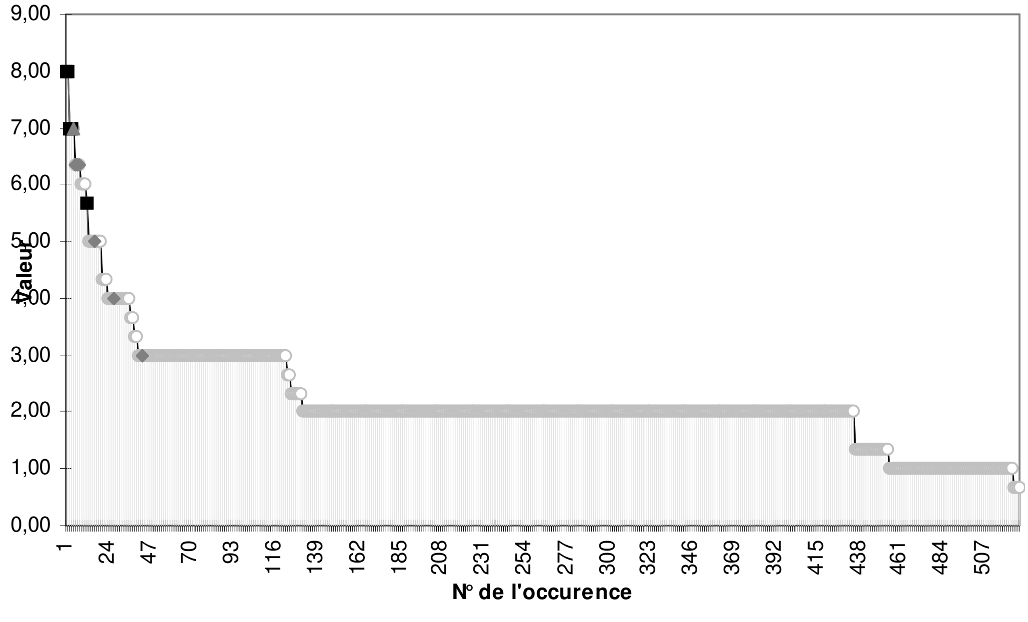


Figure 4-21. Profil des occurrences du pattern 1Bc obtenues avec le jeu de descriptions J2

En ce qui concerne le pattern 5Cau, le profil des occurrences obtenu avec le jeu de descriptions J2 est montré sur la Figure 4-22. La tendance qui vient d'être observée avec

1Bc est encore plus nette ici, avec un pic initial contenant, à son sommet, la totalité des (trois) occurrences (parmi 5 ex-aequo). Quant au contraste avec le profil qui avait été obtenu précédement (Figure 4-20), le contraste n'en est que plus fort.

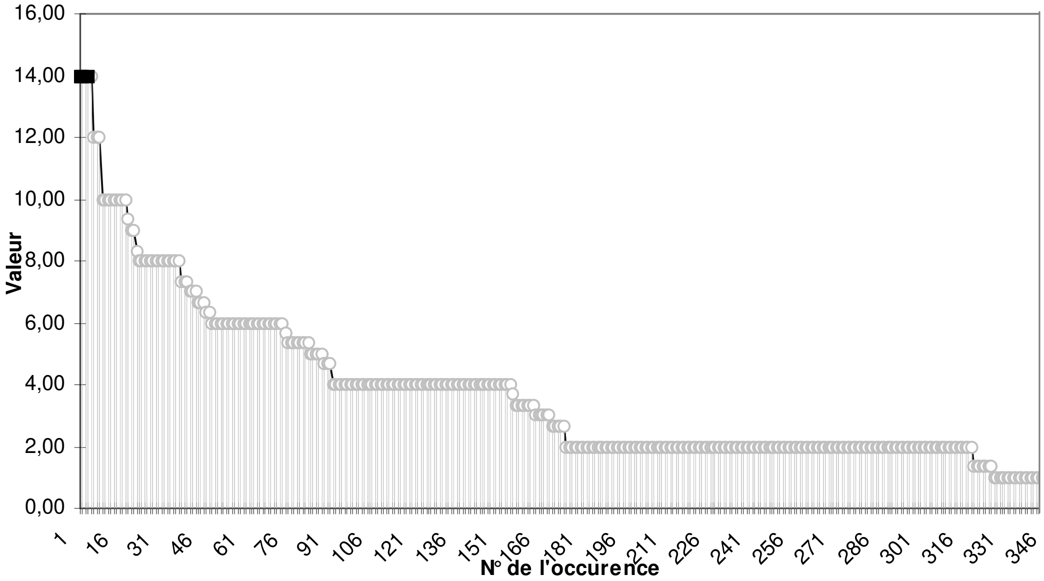


Figure 4-22. Profil des occurrences du pattern 5Cau obtenues avec le jeu de descriptions J2

L'analyse des profils d'occurrence peut être vue en termes des notions (liées) de *sensibilité* et de *sélectivité*. En effet, en pratique on souhaite pouvoir filtrer les occurrences valides des non-valides ; ou, vu autrement, on veut pouvoir arrêter la recherche d'occurrences à partir d'un certain point, en sachant qu'on ne perdra pas, au passage, d'occurrence (ou du moins que cela a une probabilité « suffisamment forte » de se produire). Nous appelons naturellement sensibilité d'un système / algorithme de localisation sa capacité à détecter **toutes** les occurrences valides. Sa sélectivité est au contraire vue comme sa capacité à détecter **uniquement** les occurrences valides — les occurrences non valides éventuellement détectées étant désignées collectivement, dans ce contexte, par *bruit*.

Voyons à présent ce qu'il en est de la sensibilité et de la sélectivité atteintes par Imprology avec les jeux de descriptions respectifs J1 et J2. Pour ce qui est de la sensibilité, la comparaison criante est celle entre les profils obtenus pour le pattern 1Bc : sur l'ensemble des occurrences localisées, la sensibilité atteinte avec J1 n'est que de 50% (6 occurrences sur 12) alors qu'elle est de 100% avec J2. Concernant la sélectivité, comparons les profils obtenus avec le pattern 5Cau. Une première manière d'évaluer les sélectivités respectives est la suivante : on cherche à fixer un seuil de valeur de manière à filtrer la proportion maximale d'occurrences valides, sans toutefois rien sacrifier à la sensibilité. On voit alors que :

- Avec J1, ce seuil doit être strictement inférieur à 0.01 (valeur des deux dernières occurrences), ce qui entraîne l'inclusion, à tort, de 351 occurrences non valides — soit un bruit catastrophique.
- Avec J2 en revanche, un seuil égal à 13 convient (et est optimal), n'entraînant l'inclusion que de 2 occurrences non valides —soit un bruit nettement plus modéré.

Au plan de la sélectivité, les résultats obtenus avec 1Bc sont moins criants, quoiqu'ils permettent clairement de conclure dans la même direction.

Indiquons qu'un second procédé d'évaluation de la sélectivité, proche du premier, consiste à fixer non pas un seuil de valeur mais un **nombre d'occurrences** : seules les k meilleures occurrences sont retenues. Les écarts de sélectivité entre J1, J2 et J3 apparaissent de façon tout aussi nette en employant ce second procédé d'évaluation.

4.6.2.7. Jeu de descriptions J3

L'évidence empirique indique que les extrema locaux de hauteur jouent un rôle dans la cognition musicale au sens large, et qu'à ce titre leur prise en compte dans un modèle de comparaison mélodique est potentiellement pertinente. Un exemple d'évidence empirique est l'application systématique, par Charlie Parker, d'accents (pics d'amplitude) à chaque extremum local de pitch — voir thèse d'Owens (Volume 1). La prise en compte des extrema locaux fera l'objet d'une discussion générale au 4.7. Pour l'heure, contentons-nous d'illustrer graphiquement, à titre d'exemple, l'enrichissement effectué sur la représentation du pattern 5Cau : voir Figure 4-23.

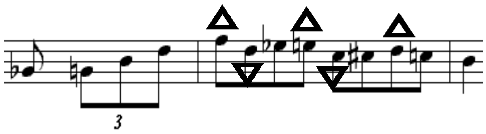


Figure 4-23. Extrema locaux de hauteur au niveau du pattern 5Cau (notation musicale standard)

Venons-en maintenant aux résultats expérimentaux. En ce qui concerne le pattern 1Bc, ceux-ci n'apportent pas d'enseignement supplémentaire. En revanche, le profil des occurrences du pattern 5Cau obtenues avec le jeu de descriptions J3 est montré sur la

Figure 4-24. (La même remarque que ci-dessus peut être faite quant à la différence observée par rapport aux profils précédents, au plan de l'échelle des ordonnées)

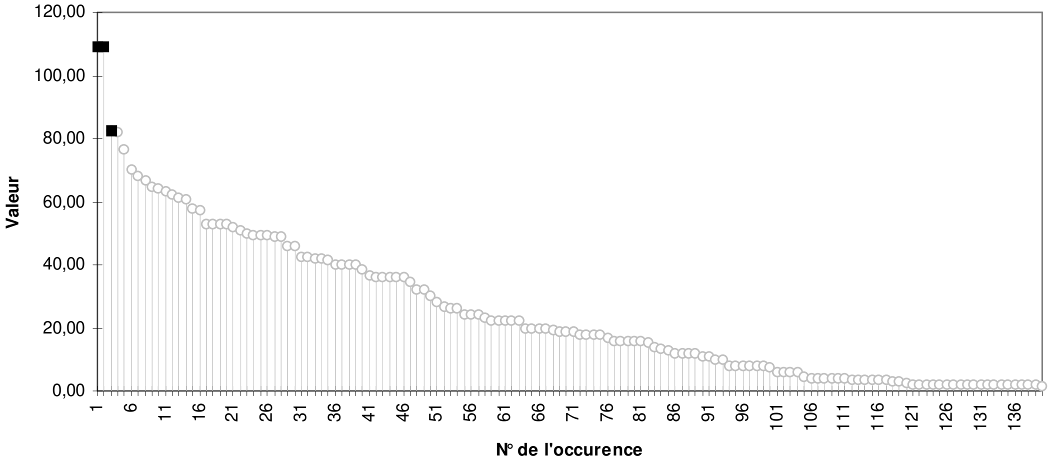


Figure 4-24. Profil des occurrences du pattern 5Cau obtenues avec le jeu de descriptions J3

Ce profil semble à première vue comparable avec celui de la Figure 4-22. Pour pouvoir en apprécier la différence, effectuons un zoom sur les 10 premières occurrences trouvées, ce qui conduit aux profils apparaissant en Figure 4-25 (jeu de descriptions J2) et en Figure 4-26 (jeu J3). La comparaison des valeurs relatives et des rangs des trois occurrences met en évidence un gain de qualité pour J3 au-delà de ce qui était obtenu avec J2. Sur l'ensemble des occurrences, la sensibilité reste égale à 100% ; la sélectivité, pour sa part, est à nouveau améliorée, puisque (idéalement) un seuillage à la valeur 82.5 sélectionne la totalité des occurrences avec un bruit nul.

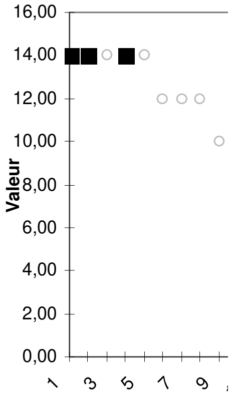


Figure 4-25. Profil des 10 meilleures occurrences du pattern 5Cau obtenues avec le jeu de descriptions J2

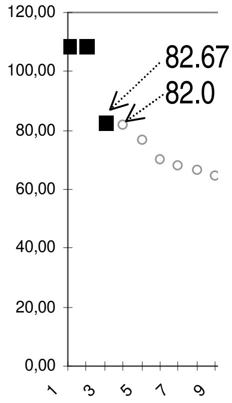


Figure 4-26. Profil des 10 meilleures occurrences du pattern 5Cau obtenues avec le jeu de descriptions J3

4.6.2.8. Discussion

Les résultats expérimentaux obtenus au travers de la localisation d'occurrences approximatives vont tout à fait dans le sens des conclusions tirées au 4.6.1.4 : un enrichissement pertinent de la représentation conduit à une amélioration qualitative et quantitative des résultats obtenus, et cela sur différents plans. Indiquons également que les résultats relatifs aux laps min/max[25], dont il ne nous a pas paru souhaitable d'allonger davantage la présente section expérimentale, mettent aussi quantitativement en évidence les apports des enrichissements successifs de la représentation.

Comme nous l'avons dit, nous avons à plusieurs égards anticipé sur la section 4.7, ayant mis en œuvre des modèles qui étaient en fait des instances — relativement simples — du modèle riche que nous y décrirons. Les résultats expérimentaux du genre de ceux que nous avons présentés dans cette section et la précédente sont entre autres à l'origine de notre proposition d'un modèle de comparaison permettant la prise en compte simultanée et pondérée d'un nombre arbitraire de descriptions. Ce *modèle d'édition valué multi-descriptions* va maintenant être décrit en détail.

4.7. Modèle d'édition valué multi-descriptions

Voici le plan de cette section : après une introduction motivant et présentant les caractéristiques générales du MEVM, nous appréhenderons (4.7.2) l'articulation entre le

---

[25] Il s'agit d'évaluer la qualité de chaque occurrence, non plus par sa valeur (c'est-à-dire la valeur de l'alignement local associé) ou par son rang dans le profil, mais par la position relative de son support par rapport aux segments minimal Lmin et maximal Mmax (cf. 4.6.2.1).

modèle de comparaison et l'étape d'enrichissement (ou de changement) de la représentation. Puis nous nous concentrerons sur la forme prise par les fonctions de contribution associées aux types d'appariements (4.7.4). Nous introduirons ensuite les trois niveaux de description — global, local et individuel. La [sous-]section 4.7.6 traitera de la formalisation des liens de dépendance entre descriptions sous forme d'un graphe. Une typologie des différentes catégories de descriptions pouvant être représentées dans le cadre du MEVM sera ensuite proposée. Les *propriétés d'invariance* de différentes catégories de descriptions à différentes catégories de *transformations* pouvant être appliquées aux séquences comparées, feront l'objet du 4.7.7.3.3. Après des considérations importantes concernant l'impact d'un enrichissement de la représentation sur la taille de l'alphabet effectif associé à une séquence (4.7.9), la section s'achèvera sur la proposition de critères d'évaluation pour les alignements issus du MEVM.

Remarques :

- Les résultats obtenus avec le MEVM seront présentés essentiellement au travers des expérimentations que nous avons conduites en matière d'extraction de patterns (section 5.5 : algorithme FlExPat, qui intègre le MEVM). Toutefois, en matière de localisation les expérimentations présentées plus haut (section 4.6) ont donné un premier aperçu des performances du MEVM, comme nous l'avons signalé dans la section correspondante.
- Lorsqu'il est plus naturel de faire référence à des *dimensions* qu'à des descriptions, nous parlons plutôt de *modèle d'édition valué multidimensionnel.*

4.7.1. Introduction

4.7.1.1. Motivation

La nécessité de la prise en compte simultanée et pondérée de plusieurs descriptions pour la localisation et l'extraction de patterns, et plus généralement pour la comparaison d'entités séquentielles, a été soulignée à plusieurs reprises dans ce qui précède.

Dans le domaine musical, cette nécessité est apparue indéniable et les résultats expérimentaux qui ont été présentés viennent à l'appui de cette considération. En biologie moléculaire, pour dépasser la représentation de la structure primaire (chaîne de symboles dont chacun désigne un monomère — cf. 2.1.1.1), les monomères peuvent être réprésentés, en tant qu'objets dans l'espace, en employant **deux descriptions**. Celles-ci sont les angles *angles dihédraux* Φ et Ψ, représentés sur une carte planaire appelée carte de Ramachandran [Ramachandran, Ramakrishnan & Sesisakharan 1963], ce qui permet de représenter la conformation du squelette de la macromolécule — protéine en l'occurrence. (Cette représentation peut être ramenée à une chaîne de caractères en employant un alphabet lié aux paires d'angles, comme nous l'évoquerons au 4.7.9).

Le modèle général de comparaison d'entités séquentielles que nous proposons ici est destiné à répondre à cette nécessité. Par rapport à la classification que nous avons utilisée au 3.2, ce modèle, baptisé *modèle d'édition valué multi-descriptions* (MEVM) s'inscrit, comme tout modèle d'édition, dans la catégorie des modèles à structure de correspondance variable. Le MEVM est conforme au modèle général de comparaison proposé en psychologie par Tversky (1977). Comme ce modèle appelé *modèle des contrastes*, le MEVM prend explicitement en compte à la fois ce qui est **identique** entre les deux entités comparées ("traits communs" [Rifqi 1996]), et ce qui **diffère** entre elles ("traits distincts"). Cela est matérialisé, respectivement, par les **contributions positives** d'appariements (identités ou substitutions « de bonne qualité », typiquement) et par les **contributions négatives** (substitutions « de mauvaise qualité » ou indels, typiquement).

Le MEVM est inspiré des modèles proposés par Mongeau & Sankoff (1990) et par Stammen & Pennycook (1993) [Rolland & Ganascia 1998a ; 1996b]. Dans ces modèles (cf. 3.2.5.2.2), deux descriptions sont explicitement prises en compte simultanément dans le calcul de la contribution associée à chaque appariement Le MEVM, pour sa part, fait intervenir un nombre arbitraire de descriptions simultanées.

A partir de maintenant, nous présentons le MEVM en tant que **modèle de similitude** plutôt que modèle de dissimilitude — ce qui n'est en rien restrictif comme nous l'avons expliqué précédemment.

Toute instance du MEVM, en tant qu'instance du modèle d'édition, peut faire intervenir un ensemble quelconque de types d'appariements (ETAA ; cf. 3.2.4.2). En particulier, au-delà des types standard [Mono]Insertion, [Mono]Délétion et [Mono]Remplacement, les types suivants peuvent avantageusement être employés suivant le contexte applicatif :

- InsertionMultiple
- DélétionMultiple
- Consolidation
- Fragmentation

- Permutation
- SubstitutionMultiple, dont la Figure 4-27 montre un exemple dans le domaine musical (jazz Be-bop improvisé en tempo rapide). Le groupe formé des 4 premières notes du second passage est substitué à celui formé par les deux premières notes du premier passage (les lignes reliant les 3 dernières paires de notes indiquent des identités). Le second groupe de notes peut être vu comme une version ornementée du premier groupe.

En ce qui concerne par exemple ce dernier type, l'exemple montré à la Figure 4-27 suggère que, moyennant une définition adéquate de fonction de contribution, les substitutions multiples peuvent rendre compte, d'une façon tout à fait adéquate, des phénomènes d'ornementation ou de broderie divers, fréquents en musique. A notre connaissance ni les permutations ni les substitutions multiples n'ont été mis en œuvre auparavant sur des séquences musicales.

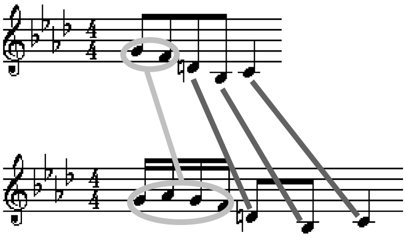


Figure 4-27. Exemple de *substitution généralisée* dans le domaine musical.

4.7.2. Enrichissement ou changement de représentation

4.7.2.1. Généralités

Nous avons évoqué précédemment la problématique générale de l'enrichissement de la représentation et celle, connexe, du changement de représentation. Nous avons notamment évoqué les modalités d'intervention d'un tel enrichissement (ou changement) dans le cas où la découverte automatique de régularités dans les séquences simule (ou assiste) un processus normalement pris en charge par un humain. Enfin nous avons proposé un schéma général amélioré pour la découverte de régularités dans les séquences (Figure 1-6, page 25) qui intercale une étape supplémentaire d'enrichissement ou de changement de la représentation, entre l'étape d'encodage d'un côté, et les étapes de localisation ou d'extraction, de l'autre. Précisons à nouveau que, si nous parlerons essentiellement d'**enrichissement** de la représentation, néanmoins les idées, concepts et techniques proposés restent valides dans le cadre d'un **changement** « destructif » de représentation, c'est à dire se traduisant par un « oubli » de la représentation initiale. (Ce genre de changement intervient par exemple dans le système de Widmer ou le système ImPact [Ramalho & Ganascia 1994 ; Ramalho & Pachet 1994] où la représentation oubliée est celle qui concerne le niveau des notes individuelles).

La Figure 4-28 illustre cette étape supplémentaire. Les descriptions sont symbolisées par des [sous-]arbres dont les feuilles (losanges) traduisent les valeurs terminales. (Par exemple, sur des séquences faisant intervenir des cercles de couleur et de coordonnées variés, la description de chaque cercle serait traduite par un arbre dont la racine porterait l'étiquette « forme » ; le premier fils serait une feuille correspondant à la **couleur** (codée par exemple sur un entier) ; le second fils serait un nœud interne étiqueté « caractéristiques géométriques », ayant deux fils traduisant respectivement le rayon (feuille) et les coordonnées du centre (nœud interne)).

Nous faisons figurer :

- L'enrichissement relatif aux descriptions *locales* et *globales.* La description de la séquence, dans sa globalité, étant traduite par l'arbre figurant en haut[26] du rectangle à angles arrondis, l'enrichissement de la représentation est matérialisé par l'ajout, sous la racine de cet arbre, de sous-arbres supplémentaires.
- L'enrichissement relatif aux descriptions *individuelles*, c'est-à-dire relatives aux éléments eux-mêmes (cf. 4.7.5) : cet enrichissement est traduit par des sous-arbres supplémentaires à l'intérieur de la case (rectangle à angles droits) matérialisant un élément quelconque de la séquence.

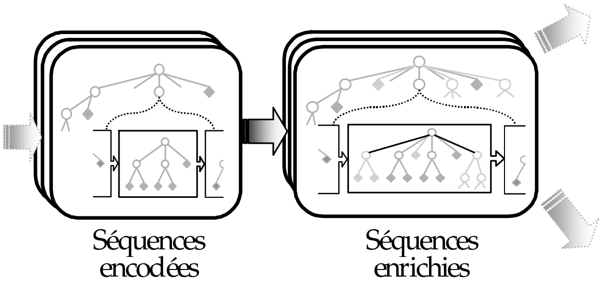


Figure 4-28. Enrichissement (ou changement) de la représentation

Rappelons que l'enrichissement ou le changement de la représentation n'a pas besoin, en pratique, d'être effectué en totalité lors d'une étape située chronologiquement **avant le début** de la localisation ou l'extraction de patterns. Il peut être préférable,

---

[26] Et dont l'un des fils pointe sur les éléments de la séquence à proprement parler (en bas du rectangle à angles arrondis).

notamment pour des raisons d'encombrement mémoire, que l'enrichissement se fasse dynamiquement en fonction des besoins liés au déroulement de l'algorithme de localisation ou d'extraction. Lorsque, comme FlExPat (Chapitre 5), cet algorithme utilise un modèle de comparaison d'entités séquentielles qui est une instance du modèle d'édition, les *descripteurs* — qui sont (rappel) les valeurs des différentes descriptions — d'un élément de séquence quelconque x sont uniquement utilisés lors du **calcul des contributions** des différents appariements dans lesquels x est impliqué. Ainsi, suivant la forme de l'algorithme de localisation ou d'extraction, les descriptions peuvent être « oubliées » — et l'espace mémoire correspondant, libéré — dès lors que l'on est assuré que l'élément en question n'interviendra plus dans aucun calcul de contribution. Ce type de mécanisme est notamment permis par FlExPat (voir en particulier la section 5.3.8.3).

De plus, l'automatisation du calcul des nouvelles descriptions n'est pas toujours indispensable. Par exemple, dans le système ImPact, l'enrichissement de la représentation (ajout des *« traits musicaux »*) est fait de façon **entièrement manuelle**, du moins au stade actuel [Ramalho 1997a]. Des cas de figure « mixtes » sont envisageables, dans lesquels l'utilisateur peut prendre en charge une partie seulement de l'opération d'enrichissement / changement de la représentation, ou encore remettre en cause, au cours de la découverte de régularités, les résultats de la phase automatique d'enrichissement de la représentation. Par contraste, dans le cadre de toutes les expérimentations présentées au 4.6, et de toutes celles qui seront présentées au 5.5.2, le changement de représentation est entièrement automatique.

4.7.2.2. Cas de la musique

Comme nous l'avons vu, le problème de la représentation des séquences et de leurs éléments se pose particulièrement lorsque l'on simule ou modélise informatiquement un processus de découverte de régularités ordinairement accompli par un « expert » humain — en l'occurrence l'analyse musicale « orientée pattern », l'objectif étant d'obtenir des résultats les plus conformes possibles à ceux de l'expert (désigné ici génériquement par « musicoloque »). A cet égard, rappelons le caractère critique du rôle du **jugement de similitude musicale** et — corrélativement — l'importance à accorder au **modèle de comparaison** entre entités séquentielles s'intégrant dans un système de découverte de régularités dans des séquences musicales.

Un nombre important de travaux concernent l'étude de la perception et de la cognition musicales, et par suite nous renseignent directement ou indirectement sur les différentes caractéristiques cognitives du **jugement de similitude** entre entités séquentielles musicales. Ces travaux sont nombreux, on peut par exemple les ouvrages généraux suivants : [McAdams & Bigand 1993 ; Sloboda 1985 ; Dowling & Harwood 1986 ; Francès 1958 ; Handel 1989 ; Deutsch 1982]. Nous avons d'ailleurs évoqué dans le chapitre d'introduction la vitalité de ce champ de recherche (cf. par exemple [Pennycook & Costa-Giomi 1996 ; Won Yi 1998]).

Dire qu'une mélodie n'est pas traitée (c'est à dire perçue, mémorisée, créée) uniquement en tant qu'une séquence de hauteurs, durées et amplitudes, est — compte tenu des résultas de ces différents travaux — un grand euphémisme… Déjà, la dimension du *timbre* intervient. Mais surtout, au-delà, les « activités musicales » au sens large, allant de la perception et l'écoute à la composition et l'improvisation en passant par la mémorisation ou encore l'interprétation, font intervenir un ensemble divers de structures cognitives et de **multiples représentations** (multiplicité évoquée par la Figure 4-29, comme expliqué plus bas). Il faut noter que ces structures et de ces représentations sont, pour une part importante, liées à une *acculturation* : la perception et la cognition musicale d'un auditeur (ou plus généralement d'un musicien) issu de la culture occidentale diffèrent à bien des égards de celle d'un auditeur issu, par exemple, de la culture indienne (au sens large).

Il serait hors de question de donner ici une vision détaillée de l'ensemble de ces travaux, ni de tous ceux, par exemple en informatique musicale, qui s'appuient directement sur eux dans le cadre de la modélisation ou de la simulation des différentes « activités musicales » qui viennent d'être mentionnées. Nous choisirons de parler prioritairement ici des travaux qui sont pour nous d'un intérêt direct du point de vue de la problématique de la représentation musicale ; nous les répartissons suivant trois axes — ce qui n'empêche pas d'inévitables recoupements :

- Premièrement, un ensemble de travaux, que l'on pourrait caractériser par une plus forte connotation « psychologie expérimentale ». En particulier, en plus des ouvrages généraux que nous avons déjà cités plus haut nous pouvons mentionner des ouvrages plus spécifiquement liés à :

    - la perception et la cognition de la hauteur [Krumhansl 1990]

    - la psychologie de l'interprétation, de l'improvisation et de la composition tels que [Sloboda 1988].

  Des références plus spécifiques, à d'autres travaux, seront faites dans la suite, lors de l'introduction des différentes descriptions musicales pouvant intervenir dans le MEVM.

- Deuxièmement, des « théories » générales ayant une connotation cognitive plus ou moins forte, telles que [Lerdahl & Jackendoff 1983, Narmour 1989 ; 1992 ; Meyer 1956]

- Troisièmement, un ensemble de travaux en informatique musicale. Dans cette discipline, une partie des travaux effectués à ce jour ont concerné les problèmes de **représentation** de connaissances, essentiellement dans le cadre de matériau mélodique ou harmonique (par exemple [Pachet 1991a ;1994 ; Horowitz 1995a]), mais aussi, par exemple, dans le cadre de matériau sonore [Rolland & Pachet 1996 ; 1998]. Plus particulièrement, un certain nombre de travaux partent du constat du manque de « sens commun » ou « bon sens » musical de la majorité des systèmes — que ceux-ci concernent la composition, l'improvisation, l'interprétation, etc. — et tentent de résoudre ce problème en introduisant le plus possible de connaissances musicales dans le 'raisonnement' (au sens large) du système[27] [Widmer 1992 ; 1994 ; Horowitz 1995a ;b ; Pachet 1991a ; Ramalho & Pachet 1994]. Par exemple, le système d'interprétation musicale de Widmer (1994) fait intervenir explicitement une phase préliminaire de changement de représentation (*transformation strategy*). Soulignons à nouveau à ce sujet que le caractère *destructif* de ce changement de représentation (« oubli » de la représentation initiale, c'est-à-dire celle au niveau des notes individuelles), lequel se retrouve aussi dans le système ImPact [Ramalho & Ganascia 1994 ; Ramalho, Rolland & Ganascia 1998], n'est pas nécessairement pertinent dans le cadre de l'analyse musicale « orientée-patterns ». Dans celle-ci en effet, manifestement tout ou partie des descriptions initiales peuvent entrer en ligne de compte. En particulier, tant le niveau des notes individuelles que celui des groupes de notes et des structures plus abstraites ont un rôle à jouer dans le jugement de similitude. Comme nous l'indiquions dans le chapitre d'introduction, une confirmation empirique de cette constatation nous a été fournie au travers à la fois de l'observation qualitative de la forme des patterns de notre lexique de référence et de leurs occurrences dans le corpus, et des expérimentations que nous avons conduites et sont, pour partie, présentées dans ce mémoire. Nos travaux se placent également dans cette perspective, les connaissances et représentations musicales que nous proposons de prendre en compte découlant principalement des données théoriques et psychologiques (au sens large) présentées plus haut.

---

[27] Ce genre de préoccupation est d'ailleurs à rapprocher de celle qui sous-tend le fameux projet CYC [Lenat 1990], qui vise à constituer une base — nécessairement gigantesque — regroupant le maximum de connaissances constitutives du 'bon sens' de tout un chacun.

En guise de récapitulation, la Figure 4-29 visualise un certain nombre de descriptions musicales (au sens le plus large du terme) que nous avons « extraites » des différentes catégories de travaux mentionnées ci-dessus. La disposition des différentes descriptions est essentiellement informelle, si ce n'est que :

- Celles situées au-dessus de la barre horizontale noire — surtout les 4 premières — sont des *descriptions de base* fréquemment utilisées dans le cadre de la représentation initiale du matériau mélodique en informatique musicale.
- Les descriptions vont en gros des plus concrètes au plus abstraites à mesure que l'on descend dans la figure.
- La proximité géométrique sur le schéma traduit souvent des liens de dépendance entre descriptions (par exemple : « métrique » et « syncopation », ou « hauteur » et « tessiture »).

Figure 4-29. Ensemble de descriptions musicales

4.7.3. Pondération

Comme nous l'avons indiqué précédemment, il n'est pas toujours nécessaire ou même souhaitable de prendre en compte la totalité des descriptions — les descriptions initiales des séquences et de leurs éléments, et celles issues de l'étape d'enrichissement / changement de la représentation. De même, il n'est pas forcément pertinent d'accorder la même importance à chacune des descriptions employées, ou encore d'y placer les mêmes enphases respectives quels que soient les objectifs poursuivis au travers de la découverte de régularités. Aussi introduisons-nous, dans le MEVM, la possibilité d'une **pondération** des différentes descriptions, traduite par un ensemble de paramètres ajustés soit par l'utilisateur, soit par le système. (Nous avons d'ailleurs proposé et implémenté des stratégies spécifiques d'ajustement de ces paramètres [Rolland & Ganascia 1996a]). Cette pondération permet de prendre en compte la notion de **point de vue** dans la comparaison d'entités (cf. [Suppes et al. 1990 ; Rifqi 1996]). Deux jeux de pondérations différents matérialisent l'adoption de deux points de vue différents.

En plus de la pondération des différentes descriptions, il est également possible dans le MEVM de donner globalement des poids à chacun des types d'appariements autorisés (des exemples seront donnés au 5.5). Pour chaque type d'appariement TA, le poids correspondant intervenient comme un facteur multiplicatif au niveau de la fonction de contribution associée à TA. Cet ensemble de poids matérialise un contrôle global, permettant en particulier de favoriser des types d'appariement au détriment d'autres, par exemple le type Remplacement par rapport aux types Insertion et Délétion.

4.7.4. Fonctions de contribution

4.7.4.1. Approche initiale

Dans le MEVM, pour tout type d'appariement, sa fonction de contribution associée est en fait une combinaison d'autres fonctions de contribution "spécialisées" chacune par rapport à une description. Soit TA un type d'appariement, X et X' les deux séquences comparées.

Rappelons que $\underline{facts}$(X) et $\underline{facts}$(X') désignent les ensembles de facteurs (y compris le facteur vide) de X et X', respectivement. On a donc :

$$contrib_{TA} : \underline{facts}(\mathrm{X}) \times \underline{facts}(\mathrm{X'}) \rightarrow \mathbf{R}$$

Pour tout appariement a, sa contribution $contrib_{TA}$ est une combinaison — traduite par une fonction notée f — des fonctions $contrib_{TA,D}$ où D décrit l'ensemble des descriptions :

Plus précisément, le mode de combinaison, matérialisé par la fonction f, est **pondéré** : il existe un ensemble de coefficients $w_1,\dots w_{ND}$ (où ND est le nombre de descriptions prises en compte) contrôlant l'importance relative de la prise en compte de chaque description dans la comparaison. Un mode de combinaison simple et intuitif est la combinaison linéaireà coefficients constants :

Equation 4-5

où les $w_{TA,Di}$ sont des réels ne dépendant pas de a.

4.7.4.2. Approche tenant compte de la saillance

Il s'avère que, suivant le contexte dans lequel on se place, on peut avantageusement choisir un mode de combinaison différent de la combinaison linéaire à coefficients constants. Sur des séquences musicales — mais l'observation se généralise aisément à d'autres domaines — il est souhaitable de rendre les coefficients de la combinaison linéaire variables en fonction de l'appariement considéré, pour tenir compte de l'**emphase** placée sur certains éléments. Plus les éléments impliqués dans un appariement sont [perçus comme] *saillants*, plus les différences entre ces éléments doivent contribuer négativement et plus, au contraire, les ressemblances doivent contribuer positivement. Prenons tout de suite l'exemple de la [mono]substitution d'une note n par une note n'. Si n est perçue comme très saillante — en raison, par exemple, d'une amplitude forte ou de sa hauteur très aiguë, soit dans l'absolu (tessiture) soit relativement aux notes voisines — alors un écart de durée entre n et n' sera d'autant plus fortement « pénalisée » (contribution très négative) ; de la même manière, une égalité de position métrique entre n et n' (cf. 4.7.7.1) sera d'autant plus fortement « récompensée » (contribution très positive).

Le modèle de comparaison de séquences monodiques de Mongeau & Sankoff (1990), décrit au 3.2.5.2.2, esquissait une telle prise en comte de la *saillance,* prise en compte qui est, par ailleurs, manifestement conforme aux différents résultats qui ont été obtenus en matière d'étude de la perception et de la cognition musicale. Nous la traduisons par l'emploi d'une combinaison linéaire de la forme suivante :

Equation 4-6

où chaque $w_{Di}(a)$ est un coefficient non constant dépendant de la saillance, quantifiée par Saillance(a), des éléments impliqués dans l'appariement a. (Lorsque a est une insertion ou une délétion d'une note n, les modalités de définition de Saillance(a) sont particulièrement aisées : il suffit de prendre la saillance de n elle-même). Ici encore, il est possible de proposer une forme, satisfaisante en pratique, de $w_{Di}(a)$:

$$w_{TA,Di}(a) = w_{TA,Di} \times [(\text{Saillance}(a)) \wedge \alpha_{TA,Di}]$$

Equation 4-7

où chaque $w_{TA,Di}$ est un coefficient constant, et chaque $\alpha_{TA,Di}$ un exposant réel permettant le cas échéant, de contrôler individuellement l'influence de la saillance sur les contributions relatives aux différentes descriptions.

Bien entendu, la saillance elle-même est une description à part entière et est, à ce titre, susceptible d'intervenir en tant que l'un des Di dans la sommation de l'Equation 4-6. Dans cette sommation, le monôme correspondant a une forme particulière, à savoir — compte tenu de l'Equation 4-7 :

$$w_{TA,Di} \times [(\text{Saillance}(a)) \wedge \alpha_{TA,Di}] \times contrib_{TA,Saillance}(a)$$

Ainsi la saillance, du point de vue de la contribution du terme qui lui est associé dans la somme (Equation 4-6), tend à jouer un rôle plus important que les autres descriptions, toutes choses étant égales par ailleurs. Cela nous paraît manifestement conforme au bon sens.

Dans les (nombreux) domaines d'application où la notion de saillance a un sens, on peut notamment faire intervenir, pour la définir, deux conceptions interdépendantes et complémentaires :

- Une conception statistique : la « saillance statistique » d'un élément est une fonction croissante de sa rareté. Nous avons indiqué plus haut que, dans une mélodie, le fait qu'une note soit particulièrement aiguë, ou jouée particulièrement fort, par rapport au reste du morceau ou du passage auquel elle appartient, a une influence positive sur sa saillance (statistique). En biologie, la non égalité des termes diagonaux dans les matrices de similarité telles que PAM-250 [Dayhoff, Schwartz & Orcutt 1978 ; Sagot 1996] peut être vue comme résultant de considérations du même genre.

- Une conception structurale : par exemple, le premier et le dernier élément d'une séquence (ou d'un groupe structural d'éléments) ont une « saillance structurale » plus forte que celle des éléments intermédiaires, toutes choses étant égales par ailleurs. Un lien peut être mentionné à ce propos avec certains résultats obtenus en psychologie relativement à la mémoire et à l'attention : les évènements situés au début et la fin d'une séance attentionnelle (par exemple, un cours) tendent à être mieux mémorisés que le milieu. En musique, d'autres facteurs contribuent à la détermination de la saillance structurale, comme la position métrique des notes (voir plus bas).

4.7.4.3. Gradualité

Suivant le contexte applicatif et la description considérée, la forme la plus adéquate à donner aux fonctions de contribution associées aux différents types d'appariements permis, peut varier. Nous souhaitons ici opposer les fonctions **graduelles**, dont les valeurs peuvent à l'extrême décrire tout le continuum de **R**, et les fonctions non-graduelles qui, à l'autre extrême, s'apparentent à des fonctions **booléennes**, ne pouvant prendre que deux valeurs différentes et antagonistes.

Illustrons cela sur l'exemple très général de la séquence X, de longueur L, contenant les valeurs successives d'une variable numérique (réelle) — exemple qui se généralise à toute description numérique intervenant dans la représentation des éléments d'une séquence. Comme cela a déjà été évoqué, il peut être avantageux d'enrichir la représentation initiale D de chaque élément — ou de remplacer celle-ci — par une autre description Δ traduisant la **variation** de D entre l'élément et son successeur (cf. à ce sujet les travaux de Saxena (1991) sur la représentation d'images graphiques).

La première manière de le faire consiste à définir Δ comme la description, elle aussi numérique, issue d'une **différence** :

$$\forall i \in \mathbf{N}_{L-1}, X[i].\Delta = X[i+1].D - X[i].D$$

Intéressons-nous à présent à la définition des fonctions de contribution spécialisées relativement à Δ, par exemple la fonction se rattachant au type d'appariement Remplacement. D prenant des valeurs appartenant à un intervalle de **R**, Δ aussi ; de ce fait il est assez naturel de définir cette fonction de contribution comme une fonction graduelle, par exemple (en faisant, pour simplifier, abstraction des considérations relatives à la saillance) :

Equation 4-8

Ainsi, en musique si D est le midipitch, Δ représente la description « intervalle chromatique entre chaque note et la suivante », exprimée en nombre de demi-tons, et la définition de la fonction de contribution issue de l'Equation 4-8 semble raisonnable. (Tant les midipitches que les tailles d'intervalles chromatiques étant assimilées à des valeurs réelles continues).

Une seconde manière de définir Δ consiste à considérer cette fois-ci la **direction** de la variation entre chaque élément et son successeur, codée par exemple de la manière suivante :

$X[i].\Delta =$ “ascendant” si $X[i+1].D > X[i].D$, et “descendant” sinon.

(Pour simplifier ici encore, il est considéré arbitrairement qu'une variation nulle est traitée de la même façon qu'une différence positive). Cette fois-ci, il est naturel de définir la fonction de contribution se rattachant au type d'appariement Remplacement comme une fonction non graduelle, pseudo-booléenne, prenant les valeurs 0 et 1 avec :

Equation 4-9

En musique, en supposant encore que D est le midipitch, Δ est la description « direction d'intervalle entre chaque note et la suivante », et à nouveau la définition de la fonction de contribution issue de l'Equation 4-9 est conforme au bon sens. Un certain nombre de travaux en psychologie de la musique ont mis en évidence l'importance du rôle joué par les directions d'intervalles, auxquels la notion de **contour mélodique** est liée (cf. par exemple [Dowling & Fujitani 1971 ; Massaro, Kallman & Kelly 1980 ; Dowling 1994]).

Nous venons donc de voir deux exemples généraux où la forme des descriptions donne lieu à des fonctions de contribution respectivement graduelles et pseudo-booléenne. Entre ces deux extrêmes peuvent intervenir des fonctions de contribution fortement discrètes, quantifiées sur quelques valeurs d'une échelle qualitative. En reprenant l'exemple ci-dessus, ce cas est rencontré si l'on définit cette fois-ci Δ comme l'**ampleur** de la variation entre chaque élément et son successeur, quantifiée selon des modalités quelconques, sur l'échelle {'grandDescendant', 'petitDescendant', 'plat', 'petitAscendant', 'grandAscendant'}. On peut alors définir la fonction de contribution se rattachant au type d'appariement Remplacement suivant un schéma traduit par le Tableau 4-8. Dans ce tableau, la valeur prise par X[i]. Δ figure en abscisse, celle prise par X[i']. Δ figure en ordonnée (ou l'inverse, puisque nous avons choisi un tableau symétrique).

Tableau 4-8. Fonction de contribution pour le type d'appariement Remplacement

	'grand Descendant'	'petit Descendant'	'plat'	'petit Ascendant'	'grand Ascendant'
'grandDescendant'	1	0	-1	-3	-5
'petitDescendant'	0	1	0	-1	-3
'plat'	-1	0	1	0	-1
'petitAscendant'	-3	-1	0	1	0
'grandAscendant'	-5	-3	-1	0	1

Soulignons que ce concept d'échelle qualitative de descripteurs se retrouve, en musique, à la fois dans le cadre de travaux théoriques généraux (voir par exemple [Narmour 1989 ;1992]) ou dans celui de travaux plus spécifiques (par exemple [Ramalho 1997a]).

Pour clore cette section, indiquons que nous avons défini dans Imprology un ensemble fonctions standard appartenant à chacune des trois catégories ci-dessus ; des exemples concrets d'utilisation de ces fonctions standard seront vus au 5.5.2.

4.7.5. Descriptions individuelles, locales et globales

4.7.5.1. Introduction

La première distinction que nous opérons entre les différentes descriptions ou catégories de descriptions concerne l'étendue horizontale, c'est-à-dire la partie de la séquence sur laquelle porte une description. Nous distinguons trois niveaux, pouvant être vus comme hiérarchiquement imbriqués :

- Une description *globale* concerne la séquence dans sa totalité.

- Une description *locale* concerne un facteur donné.
- Enfin, une description *individuelle* (ou atomique) concerne un élément et un seul

Remarque : lorsqu'on a affaire à des séquences temporelles, la notion de description locale se généralise à celle d'une description portant sur un laps de temps donné, ce laps ne coïncidant pas forcément avec un facteur. Bien sûr cette généralisation est également possible pour tout autre type de séquences dont l'ordre naturel est continu plutôt que discret.

En informatique musicale, une grande majorité de travaux se sont cantonnés aux descriptions individuelles — midipitch, durée et amplitude des notes typiquement. La prise en compte de descriptions locales et globales constitue le moyen de dépasser le paradigme du "note par note" qui, comme l'a souligné Widmer, constitue une limitation majeure de nombre de systèmes existants.

Des exemples, musicaux comme extra-musicaux, de chacune des trois catégories de descriptions ont été vus dans le chapitre introductif, et nous en présenterons de nombreux autres dans la suite. Pour l'heure, nous allons expliciter les deux rôles distincts joués par les descriptions locales et les descriptions globales.

4.7.5.2. Rôles respectifs

- Les descriptions globales jouent deux rôles distincts :
    - q Elles interviennent comme descriptions primitives (cf. 4.7.6.1) dans le calcul de descriptions individuelles *relationnelles*, cf. 4.7.7.3.1. Par exemple, pour chaque note la description individuelle 'degré dans la tonalité globale' est calculée à partir de la description de base 'hauteur absolue' (ou 'midipitch') de la note — modulo 12 — et de la description globale 'tonalité' de la séquence à laquelle la note appartient. (Ce calcul automatique est permis directement en réutilisant les fonctionnalités du système MusES [Pachet 1994])
    - q Elles interviennent *de facto* comme description supplémentaire de chaque élément d'une séquence. En tant que telles, leur prise en compte peut délibérément provoquer un biais qui, comme évoqué précédemment, favorise entre autres la mise en relation (équipollence) de paires de facteurs appartenant à la même séquence au détriment de paires de facteurs

appartenant à des séquences différentes. Ajoutons qu'il est bien sûr très aisé de provoquer, de manière analogue, un biais opposé.

Lors de l'enrichissement de la représentation (cf. 4.7.2), chaque élément de séquence se voit donc affecter deux catégories de descriptions supplémentaires. Pour la seconde, il s'agit d'un simple héritage des descriptions globales —tout se passe alors, à ce niveau, comme un ajout de descriptions individuelles.

- De même, les descriptions **locales** jouent deux rôles distincts :
    - Elles interviennent elles aussi comme descriptions primitives dans le calcul de descriptions individuelles relationnelles. Par exemple, pour chaque note la description individuelle 'degré dans la tonalité locale associée à la cellule harmonique courante' (cf. 4.7.7.2.3) est calculée à partir de la description de base 'hauteur absolue' (ou 'midipitch') de la note — modulo 12 — et de la description locale 'tonalité' de la cellule harmonique à laquelle la note appartient.
    - Elles interviennent comme description supplémentaire de chaque élément d'une séquence. En tant que telles, leur prise en compte peut, ici encore, provoquer délibérément un biais favorisant par exemple la mise en relation (équipollence) de paires de facteurs dont le support temporel coïncide avec (ou est inclus dans) celui de descriptionsLocales de type identique : prenons l'exemple de la description locale 'Cellule harmonique' ; elle peut être employée pour favoriser l'équipollence de paires de facteurs coïncidant avec des cellules harmoniques identiques, par exemple des « II-V » majeurs. Ce biais particulier a un intérêt indéniable, car on sait qu'en jazz improvisé, souvent chaque pattern du catalogue personnel de patterns de l'improvisateur (cf. 4.2.2) est — consciemment ou inconsciemment — associé à un type particulier de cellule harmonique[28] [Ramalho 1997a].

4.7.5.3. Formalisation et implémentation

Nous mettons en œuvre une formalisation spécifique pour les descriptions locales, laquelle débouche sur une implémentation elle-aussi particulière.

---

[28] Ajoutons qu'une telle prise en compte des cellules harmoniques est indispensable dans le but d'automatiser la constitution et l'enrichissement de la *mémoire musicale* (de patterns récurrents) d'un système comme ImPact — voir à ce sujet les perspectives dessinées à l'issue du présent mémoire.

Nous proposons une structure de données, appelée naturellement *descriptionLocale*, description structurée (qui serait donc représentée par un [sous-]arbre sur la Figure 4-28) avec le contenu suivant :

- Le *support temporel* (un laps de temps relatif ou absolu), qui identifie horizontalement (temporellement) la description locale
- Le *type* (par exemple : 'contour ascendant', 'cellule harmonique' ou 'phrase'), qui l'identifie sémantiquement
- Éventuellement, des *caractéristiques supplémentaires* qui spécifient l'instance particulière de cette description locale au sein de son type. Par exemple, un contour ascendant peut être spécifié par la caractéristique supplémentaire 'taille', égale à la valeur absolue de la différence de valeur entre le point initial et le point final.

Cette structure de donnée est influencée, dans l'état d'esprit, par les *PACTs* (*Potential Actions*) du système ImPact — issues elles-même des *Stratégies* proposées par Pachet (1987), ou encore par les *RegionFrames* proposées par Horowitz 1995, autant de structures de données permettant de représenter des propriétés associées à un laps de temps (relatif ou absolu) déterminé.

Dans le cadre d'Imprology, les descriptions locales sont réifiées au travers de la classe Smalltalk 'DescriptionLocale'. Cela nous a permis de réutiliser les classes et *méthodes* (au sens de la programmation objet) qui ont été développées au sein de MusES relativement à la représentation et au traitement des laps de temps (algèbre temporelle). Nous avons par ailleurs enrichi ces fonctionnalités pour prendre en charge les spécificités liées à la manipulation de descriptions locales.

Lors de l'enrichissement (ou du changement) de la représentation, la prise en compte des descriptions globales et locales est effectuée de la manière suivante :

- Ayant identifié à quelle séquence elle appartient, chaque note se voit adjoindre les deux catégories de descriptions supplémentaires issues des descriptions **globales** de la séquence en question (voir plus haut) ;
- Pour ce qui est des descriptions locales, le processus est semblable, à la différence notable que, cette fois-ci, il faut, pour chaque note, déterminer l'ensemble des supports temporels de descriptions locales auxquels elle appartient. Par exemple,

si on détermine qu'une note n appartient au support temporel d'une description locale DL1 de type 'Cellule harmonique' et à celui d'une description locale DL2 de type 'Maximum local de hauteur', l'enrichissement de n consistera en l'ajout cumulé des descriptions engendrées par DL1 et celles engendrées par DL2.

4.7.6. Dépendance entre descriptions

4.7.6.1. Descriptions de base, descriptions dérivées

Le calcul de certaines descriptions s'effectue à partir des valeurs d'autres descriptions, comme nous avons eu plusieurs fois l'occasion de le voir.

Donnons-en encore un exemple extra-musical. Plaçons-nous par exemple dans le contexte d'un agent intelligent d'interface, où chaque élément des séquences enregistrées et exploitées par le système représente une macroaction effectuée par l'utilisateur, c'est-à-dire une succession d'actions élémentaires (souris ou clavier), ayant permis d'atteindre un objectif. Un tel objectif pourrait être, par exemple, de parvenir à identifier une nouvelle espèce d'orchidée au sein du système hypermedia de classification taxinomique Satelit [Akoulchina & Ganascia 1997]. Supposons que chaque macroaction soit entre autres représentée à l'aide de la description numérique strictement positive « nombre total d'actions élémentaires (éventuellement erronées) effectuées par l'utilisateur avant d'atteindre l'objectif », notée NE. Pour caractériser le progrès ou au contraire la régression de l'utilisateur quant à son aisance à manipuler l'interface, on peut adjoindre à chaque élément un descripteur intitulé *%ΔNE*, indiquant le pourcentage de variation de NE par rapport à sa valeur précédente. Cette nouvelle description se calcule à partir de NE ; on a, pour tout élément S[i]:

$$S[i].\%\Delta NE = (S[i].NE - S[i-1].NE) / S[i-1].NE$$

Nous disons alors que %ΔNE *dérive* de NE, ou encore que %ΔNE est *dépendante* de NE. NE est dite *primitive* par rapport à %ΔNE. Nous distinguons alors deux catégories de descriptions :

- Les *descriptions de base*, données au départ et qui ne sont calculées à partir d'aucune autre description. En biologie, la structure primaire d'une macromolécule est, comme nous l'avons vu, typiquement représentée à l'aide d'une seule description de base : la lettre désignant chaque acide aminé ou nucléotide. En musique, lorsque les séquences sont fournies dans une représentation de type MIDI [SMF 1988 ; Moog 1986] les descriptions de base individuelles sont les suivantes [29] :

- Midipitch (hauteur de la note sur une échelle chromatique allant de 0 à 127, où 60 représente le do médian (*middle C*) du clavier).
- Instant de début en temps relatif (mesuré en *ticks*, unités correspondant à une subdivision égale de la noire, typiquement en 192 ou 384).
- Durée (mesurée elle-aussi en ticks).
- Amplitude (sur une échelle entière allant de 1 à 127)

C'est la situation rencontrée dans nombre de systèmes musicaux, en particulier Imprology. Parmi les autres descriptions de base pouvant être fournies, on peut mentionner les descriptions globales telles que tempo ou schéma harmonique.

- Les *descriptions dérivées*, qui viennent enrichir la représentation initiale (descriptions de base) et dont le calcul fait intervenir une ou plusieurs autres descriptions, soit de base, soit dérivées. De nombreux autres exemples de descriptions dérivées seront vus dans la suite.

4.7.6.2. Formalisation et exemple

Nous proposons de formaliser ces liens de dépendance par un **graphe orienté** dont chaque sommet représente une description et une seule et dont chaque arc D→D' traduit le fait que D' dépende de D. Ce graphe, appelé *graphe de dépendance entre descriptions*, possède une ou plusieurs racine(s) (sommet sans antécédents) correspondant aux descriptions de base. A titre d'exemple, considérons la description individuelle 'niveau métrique' (voir 4.7.7.3.2), qui est dérivée des 2 descriptions suivantes :

- description globale 'signature métrique' du morceau concerné (cf. 4.7.7.1.2).
- description individuelle 'instant de début', exprimé en temps relatif, qui ici est pris *modulo* un entier dépendant de la signature métrique (4 pour la signature 4/4).

---

[29] En réalité dans le format MIDI les choses sont un peu moins simples : chaque note est représentée par un début (évènement *NoteOn*) et une fin (évènement *NoteOff*) et chaque évènement est accompagné de la durée relative le séparant du précédent (*delta-time*). Ainsi l'instant de début et la durée des notes, implicites dans ce format mais plus naturels à manipuler, sont calculés très simplement à partir des informations contenues dans le flot (*stream*) MIDI.

Nous formalisons cela par le graphe de dépendance partiel montré à la Figure 4-30.

Figure 4-30. Graphe partiel de dépendance entre descriptions

Le graphe de dépendance entre descriptions est calculé automatiquement par Imprology et utilisé par lui pour déterminer l'ordre des calculs des différentes descriptions de la hiérarchie. Pour automatiser ce calcul, nous tirons parti des fonctionnalités du langage orienté objet utilisé : pour être précis et technique, indiquons que pour détecter le lien de dépendance D→D', nous faisons déterminer au système si la **méthode de calcul de D'** fait un (ou plusieurs) appel(s) à une **méthode d'accès à D**. Ce type de détermination se programme aisément en Smalltalk-80/VisualWorks2.5.

4.7.6.3. Intérêt

Outre le fait qu'il soit utilisé par Imprology pour ordonnancer les calculs des différentes descriptions, l'intérêt de ce graphe est double dans un contexte d'implémentation :

- D'une part, il peut permettre au système de vérifier la cohérence d'un jeu de descriptions : il s'agit de détecter, et de signaler le cas échéant à l'utilisateur, des situations d'incohérences où une description D dépend d'une description D' elle-même dépendante de D. Pour cela, l'existence de cycles dans le graphe de dépendance est testée.

- D'autre part, la visualisation graphique du graphe de dépendance possède un intérêt intrinsèque notamment lors de la phase du choix par l'utilisateur des jeux de description. Un lien peut être établi ici avec la problématique de l'invariance des descriptions à certaines *transformations* (voir 4.7.7.3.3).

4.7.7. Catalogue de descriptions

Nous présentons dans cette section un ensemble de descriptions ou catégories de descriptions intégrables dans le MEVM, les modalités générales de cette intégration ayant été abordées plus haut. Nous privilégions naturellement dans cette présentation les descriptions musicales, en incluant :

- une majorité de celles utilisés dans les travaux d'informatique musicale antérieurs, surtout ceux qui concernaient la découverte de patterns et la comparaison de mélodies.

- d'autres descriptions, ayant à notre avis un rôle à jouer dans la comparaison de séquences musicales mais qui, à notre connaissance, n'avaient jamais été indroduites auparavant

Nous nous plaçons délibérément dans le cadre de la musique tonale occidentale au sens large, incluant le jazz tonal : Be-bop, Swing, etc. (cf. 4.2). Néanmoins, une partie des descriptions proposées restent valides et pertinentes dans un contexte atonal. De même, nous nous concentrons sur la représentation de matériau monodique, toutefois un certain nombre de descriptions ci-après restent directement adéquates pour le cas de matériau polyphonique ou harmonique, d'autres descriptions pouvant l'être moyennant certaines adaptations.

Une énumération exhaustive de descriptions et de catégories de descriptions serait à la fois fastidieuse et sans grand intérêt. Nous choisissons d'en présenter quelques-unes en détail et les autres plus succintement, mettant l'accent sur la **formalisation** et l'**implémentation** dans le cadre du MEVM et, le cas échéant, dans le cadre d'Imprology. La classification issue de nos distinctions global/local/individuel, puis intrinsèque/relationnel (cf. 4.7.7.3.1) puis enfin extra-musical/musical fournit une vision plus synthétique de l'éventail de descriptions que n'en offrirait une pure énumération. Cette classification ou typologie met aussi en évidence les liens et l'analogie existant entre des descriptions qui ont été proposées dans des domaines d'application ou dans des contextes différents. En outre, cette typologie est directement utile dans une perspective de formalisation et, au-delà, d'implémentation.

Soulignons qu'en ce qui concerne le calcul (automatique) des descriptions dérivées à partir de descriptions primitives, dans tous les cas où ce calcul, implémenté dans Imprology, ne pose aucune difficulté particulière (par exemple, le calcul de la description locale 'Extremum local'), nous n'y ferons pas référence.

4.7.7.1. Descriptions globales

4.7.7.1.1. Domaines extra-musicaux

Nous avons mentionné plus haut des descriptions globales à portée générale, indépendantes du domaine d'application dont sont issues les séquences. A l'extrême, la description globale 'numéro d'ordre de chaque séquence dans le corpus' peut être employée comme biais favorisant l'extraction de patterns spécifiques à une séquence particulière du corpus (quorum 1 : répétitions internes). Naturellement, la même description peut être employée pour provoquer le biais inverse.

4.7.7.1.2. Domaine musical

Les descriptions globales pouvant être utilisées sont notamment :

- La *métrique* du morceau, généralement fixe dans les genres musicaux qui nous occupent. Pour éviter des confusions dans la suite nous la désignons par *signature métrique* du morceau. L'exemple le plus courant est la signature métrique 4/4 (4 noires par mesure), rencontrée en l'occurrence dans presque tous les morceaux transcrits par Owens (cf. 4.4).
- Son *tempo*, ou tempo moyen (si disponible).
- Sa *tonalité globale* : note racine (tonique) et mode, par exemple sol mineur.
- Le cas échéant, sa *grille d'accords (chord changes)*. La Figure 4-31 montre par exemple la grille de « Stella by Starlight » (de N. Washington & V Young). En jazz on distingue par exemple, comme types de grilles d'accords (schémas harmoniques), les *Blues* des *32-AABA* (ou encore ABA'B, ABAB', ABCD [Baudoin 1992a, p. 13-14]). On peut y ajoindre d'autres informations comme le nombre de chorus.

On peut envisager en outre, le cas échéant, toute autre description de nature musicologique, traduisant par exemple le genre musical dans lequel s'inscrit l'oeuvre (par exemple: baroque vs. romantique), le nom du compositeur ou du soliste (jazz), l'année de création (composition, improvisation) de l'œuvre, etc.

Figure 4-31. Grille d'accords de « Stella by Starlight » (N. Washington & V Young)

4.7.7.2. Descriptions locales

4.7.7.2.1. Domaines extra-musicaux

Des exemples généraux de descriptions locales ont été donnés précédemment, par exemple les contours monotones (ascendants, descendants) pouvant exister au sein de la séquence des valeurs d'une quelconque description numérique. La description locale traduisant un pic d'une certaine largeur centré sur un extremum local peut également être pertinente ; nous la verrons ci-dessous dans le cadre de séquences mélodiques.

En musique, les descriptions locales pertinentes à prendre en compte dans le MEVM sont nombreuses. Nous allons en détailler quelques-unes, en nous plaçant naturellement du point de vue de leur intégration dans le MEVM et par suite dans Imprology.

4.7.7.2.2. Extrema locaux

Liés à la notion de contour évoquée plus haut, les extrema locaux (de hauteur, d'amplitude, etc.) peuvent jouer un rôle (entre autres) dans la perception ou l'interprétation musicale, et, au-delà, dans le jugement de similitude entre passages mélodiques. Par exemple, Owens a noté que C. Parker appliquait systématiquement un accent de dynamique (note jouée plus fort) sur les notes constituant un maximum local de hauteur (et sur les notes proches de ce maximum). Inversement, les notes constituant un minimum local de hauteur étaient jouées moins fort.

La Figure 4-32 illustre la notion d'extremum local de hauteur (midipitch) sur un passage improvisé de Parker. Les extrema locaux y sont signalés graphiquement sur la représentation temps-fréquence et sur la portée.

Figure 4-32. Extrema locaux de hauteur dans un passage improvisé de Charlier Parker (occurrence du pattern 5Cau selon Owens)

Bien entendu, les extrema locaux pourraient être pris en charge sous forme d'une description **individuelle**, au travers de laquelle on distinguerait uniquement les notes qui correspondent aux extrema eux-mêmes. Cependant il paraît souhaitable d'etoffer ce schéma, en introduisant une description locale se rapportant au pic (resp. au creux) centré sur le maximum (resp. le minimum). Celle-ci pourra alors traduire les caractéristiques géométriques du pic, c'est à dire la localisation précise de son centre, sa taille et sa morphologie (étalement). Plus précisément, la descriptionLocale de type 'Extremum local' possède (au sens du 4.7.5.3) les **caractéristiques supplémentaires** suivantes :

- Polarité ('maximum' ou 'minimum')
- Valeur (de l'extremum)
- Laps central (un laps de temps relatif)
- "Amplitudes" gauche et droite
- Etendues gauche et droite (exprimées en temps relatif)

Ainsi, la représentation d'une note « appartenant » à un pic centré sur un extremum sera enriche par deux descriptions individuelles (cf. 4.7.5) :

- L'une traduisant son **appartenance** à un tel pic, simplement héritée de la description locale associée.
- L'autre traduisant le cas échéant la **position** de la note au sein du pic, en particulier si la note correspond, précisément, à l'extremum.

4.7.7.2.3. Cellules harmoniques

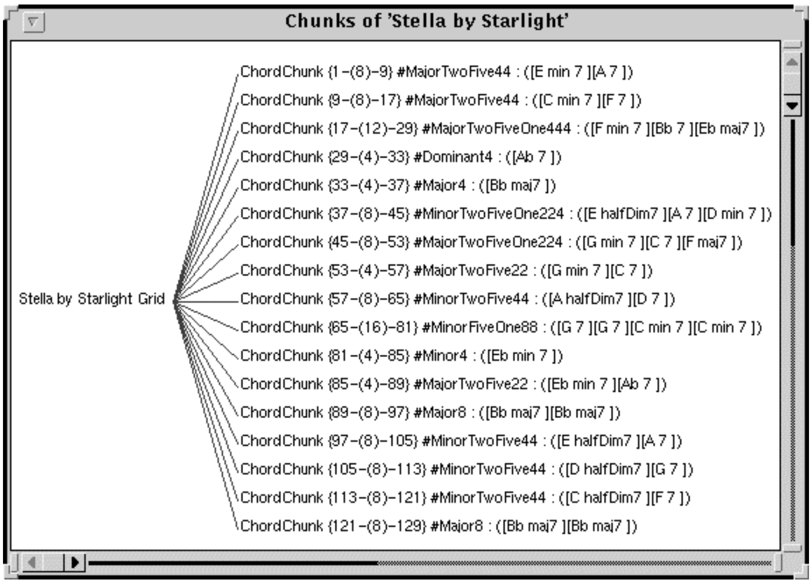
Les cellules harmoniques, schéma d'accords type (*chord chunks*) tels que II-V, V-I, II-V-I, VI-II-V-I, « Christophes » et autres « Anatoles » [Baudoin 1992a ;b]) ont été évoqués plus haut. Elles jouent un rôle indéniable en jazz tonal, en relation notamment avec la notion de pattern dans le cadre de l'improvisation : comme nous l'avons évoqué, chaque pattern du catalogue personnel de patterns de l'improvisateur (cf. 4.2.2) tend souvent à être — consciemment ou inconsciemment — associé à un type particulier de cellule harmonique [Ramalho 1997a]. Cette tendance a d'ailleurs été mise en évidence à partir de l'étude du jeu improvisé de divers grands solistes de jazz [Baker 1980 ; Owens 1974].

Corrélativement, un système comme ImPact (présenté plus haut) confère une place importante aux cellules harmoniques. Premièrement, elles sont à la base du mécanisme de segmentation mis en œuvre ; cela est suggéré comme un moyen efficace et pertinent de résoudre le problème général de la détermination des segments temporels de raisonnement dans un système produisant de la musique en temps-réel [Ramalho & Ganascia 1994 ; Ramalho, Rolland & Ganascia 1998]. Deuxièmement, elles servent pour l'indexation de la *mémoire musicale* du système.

Notre formalisation des cellules harmoniques comme descriptionLocale est inspirée de celle utilisée proposée dans le cadre d'ImPact. Les caractéristiques supplémentaires qui interviennent sont :

- **Tonalité,** par exemple ré majeur pour le II-V (majeur) Em7-A7.
- **Structure rythmique**, liste d'entiers indiquant le nombre de temps (noires) que dure chacun des accords. Par exemple, pour un « II-V-I » s'étendant sur 2 mesures 4/4 et dont les accords durent respectivement 1/2, 1/2 et 1 mesure, la caractéristique supplémentaire 'structure rythmique' sera : {2,2,4}.

Au plan de d'implémentation, nous réutilisons un ensemble de *classes* et *méthodes* de MusES et d'ImPact — également programmé en Smalltalk-80 (ParcPlace VisualWorks) — permettant de segmenter automatiquement une grille d'accord en une succession de cellules harmoniques appartenant à des types déterminés [Pachet 1991b ; Ramalho 1997]. A cette fin, un moteur d'inférence à l'ordre 1 en chaînage avant, intégré à Smalltalk/VisualWorks, est employé : le système NéOpus [Pachet 1992]. La Figure 4-33 montre l'affichage, par le système, du résultat de la segmentation automatique de la grille d'accords de *« Stella by Starlight »* (cf. Figure 4-31) en 17 cellules harmoniques *(« chord chunks »).*



```
Chunks of 'Stella by Starlight'

Stella by Starlight Grid
ChordChunk {1-(8)-9} #MajorTwoFive44 : ([E min 7 ][A 7 ])
ChordChunk {9-(8)-17} #MajorTwoFive44 : ([C min 7 ][F 7 ])
ChordChunk {17-(12)-29} #MajorTwoFiveOne444 : ([F min 7 ][Bb 7 ][Eb maj7 ])
ChordChunk {29-(4)-33} #Dominant4 : ([Ab 7 ])
ChordChunk {33-(4)-37} #Major4 : ([Bb maj7 ])
ChordChunk {37-(8)-45} #MinorTwoFiveOne224 : ([E halfDim7 ][A 7 ][D min 7 ])
ChordChunk {45-(8)-53} #MajorTwoFiveOne224 : ([G min 7 ][C 7 ][F maj7 ])
ChordChunk {53-(4)-57} #MajorTwoFive22 : ([G min 7 ][C 7 ])
ChordChunk {57-(8)-65} #MinorTwoFive44 : ([A halfDim7 ][D 7 ])
ChordChunk {65-(16)-81} #MinorFiveOne88 : ([G 7 ][G 7 ][C min 7 ][C min 7 ])
ChordChunk {81-(4)-85} #Minor4 : ([Eb min 7 ])
ChordChunk {85-(4)-89} #MajorTwoFive22 : ([Eb min 7 ][Ab 7 ])
ChordChunk {89-(8)-97} #Major8 : ([Bb maj7 ][Bb maj7 ])
ChordChunk {97-(8)-105} #MinorTwoFive44 : ([E halfDim7 ][A 7 ])
ChordChunk {105-(8)-113} #MinorTwoFive44 : ([D halfDim7 ][G 7 ])
ChordChunk {113-(8)-121} #MinorTwoFive44 : ([C halfDim7 ][F 7 ])
ChordChunk {121-(8)-129} #Major8 : ([Bb maj7 ][Bb maj7 ])
```

Figure 4-33. Segmentation automatique de la grille d'accords de *« Stella by Starlight »* en cellules harmoniques *(« chrord chunks »).*

4.7.7.2.4. Gap-fills

Parmi les structures intervenant notamment dans la perception et l'écoute musicale (4.7.2.2), nous choisissons ici de nous concentrer sur les « gap-fills » [Narmour 1989, p. 220-224], structures mélodiques constitués d'un saut (*leap*) suivis d'un contour linéaire dans la direction opposée. La Figure 4-34 en montre un exemple, issu du jeu improvisé de C. Parker (pattern 10b du lexique d'Owens). A ces structures, fréquentes dans la musique occidentale, les structuralistes ont souvent donné un statut particulier. Meyer (1973) parle de 'remplissage de creux' (gap-filling) et Schenker (1956) de 'remplissage d'intervalle', le saut étant dans ce dernier cas nécessairement ascendant. Schématiquement, le saut initial engendre chez l'auditeur une attente de descente linéaire dans la direction opposée. Les gap-fills sont notamment pris en compte dans les travaux de Widmer (1993 ; 1994) et ceux d'Horowitz (1995a,b).

Figure 4-34. Exemple de gap-fill en jazz Be-bop improvisé (C. Parker)

Nous introduisons, pour la descriptionLocale de type 'Gap-fill', des caractéristiques supplémentaires permettant notamment, ici encore, de différencier les rôles respectifs des notes appartenant à un même gap-fill (en particulier celle qui précède le saut et celle qui le suit). Ces caractéristiques supplémentaires rendent compte de la taille du saut et de la morphologie du contour linéaire qui le suit (étendue horizontale, « pente »).

4.7.7.2.5. Structure de phrase

Nous avons « gardé pour la fin » une description locale dont l'importance est à la hauteur de celle que la notion de **phrase** — et plus généralement celle de *structure de groupement* — revêt en musique [cf. par exemple Lerdahl & Jackendoff 1983 ; Narmour 1989 ; 1992]. La formalisation de la descriptionLocale 'Groupe' est fort simple, puisqu'il n'a a pas de caractéristique supplémentaire à introduire : un groupe quelconque est caractérisé entièrement par le laps de temps qui lui correspond. Aussi, au sens de la programmation par objets la descriptionLocale Groupe subsume toutes les autres descriptions locales, chaque instance de ces dernières sous-tendant une structure de groupement spécifique.

Différents critères ou combinaison de critères sont employés pour effectuer automatiquement la segmentation d'une surface mélodique (voir par exemple [Rowe 1993 ; Stammen & Pennycook 1994]) ; le plus simple, qui est mis en œuvre dans un certain nombre de systèmes, consistant à définir la frontière entre deux groupes successifs comme résultant d'un **silence** de durée suffisante — c'est-à-dire supérieure à un certain seuil. Ce critère simple, dans le contexte de notre corpus de référence, s'avère très efficace — ce qui est lié, naturellement, au type de jeu improvisé dont il s'agit.

La formalisation que nous présentons pour la prise en compte de la structure de groupement dans les séquences musicales est très **flexible** (et se généralise aisément à des domaines non musicaux) : non seulement la délimitation d'un groupe peut se faire suivant plusieurs critères indépendants ou combinés, mais surtout **la segmentation n'a nul besoin d'être unique**, déterminant une partition figée de la surface musicale. Au contraire, **une multiplicité de structures de groupement entrelacées sont permises**, ce qui selon nous reflète indéniablement mieux la réalité que les formalismes à segmentation unique. Cette multiplicité se traduit tout simplement par la création d'un ensemble de descriptions locales 'Groupe' pouvant présenter des recoupements.

La principale description dérivée de la description locale 'Groupe' est la description individuelle 'PositionDansGroupe', qui permet en particulier de distinguer la première et la dernière note d'un groupe ou d'une phrase, des autres notes qui le/la constituent. Cette distinction est d'ailleurs mise en œuvre également par Widmer (1993).

4.7.7.2.6. Autres

D'autres descriptions locales peuvent avantageusement être intègrées au sein du MEVM. Les **contours monotones** ont été mentionnés précédemment. Ils peuvent porter sur la hauteur mais aussi, par exemple, sur l'**amplitude** (crescendo, diminuendo). Des descriptions locales plus spécifiques peuvent également traduire des montées ou descentes caractéristiques effectuées par **degrés diatoniques** (*stepwise* [Horowitz 1995a ; 1995b]) ou par **degrés chromatiques**. (En ce qui concerne ces derniers, le pattern 5Cau du lexique d'Owens (voir Figure 4-32) est un exemple parlant). Par ailleurs, peuvent être introduites comme descriptions locales d'autres catégories de propriétés délimitées dans le temps comme l'existence de polyrythmes [Horowitz 1995a ; 1995b], d'arpèges ou encore de régions « fortement chromatiques » ou « de forte densité horizontale ». La prise en compte de ces régions — laquelle se généralise aisément à des domaines extra-musicaux (densité horizontale) — donne lieu à des descriptions relationnelles très intéressantes (4.7.7.3.1).

4.7.7.3. Descriptions individuelles

4.7.7.3.1. Descriptions intrinsèques versus relationnelles

Au sein de la catégorie des descriptions individuelles, comme évoqué ci-dessus nous distinguons les descriptions (ou les descripteurs) *intrinsèques* de celles, *relationnelles*, qui résultent de la mise en relation d'un (ou plusieurs) descripteur(s) individuel(s) avec un (ou plusieurs) autres descripteur(s). Il y a plusieurs possibilités pour ce(s) dernier(s) :

- Soit le descripteur individuel homologue d'un autre élément. L'exemple typique est celui de la **variation** inter-éléments, déjà évoquée plus haut : si D est une description individuelle numérique, on peut s'intéresser à la variation de la valeur de D entre chaque élément et son suivant, variation exprimée par une différence, un ratio, une direction de variation (+/-), une amplitude qualitative de variation (grande/petite), etc. Soit un descripteur local. En reprenant l'exemple d'une description individuelle numérique D, supposons qu'on ait enrichi la représentation par la description locale 'Extremum local' (pics ou creux). On peut alors adjoindre à chaque élément de séquence le descripteur (individuel) relationnel 'Position dans l'extremum local' ; cela permettra par exemple d'apparier préférentiellement des éléments ayant un statut semblable dans deux pics différents — par exemple les **centres** de ces pics.

- Soit encore un descripteur global. Considérons par exemple la description globale ‘Dmoy’ : ‘Moyenne de D sur l’ensemble de la séquence’, D étant une description numérique. On peut alors par exemple ajouter la description individuelle relationnelle ‘Comparaison avec moyenne’, traduisant pour chaque élément x le signe de x.D - Dmoy.

Les descriptions intrinsèques ont fait l’objets de maints exemples dans ce qui précède. Nous présentons à la sous-section suivante davantage de descriptions relationnelles adéquates pour les séquences musicales.

4.7.7.3.2. Descriptions relationnelles musicales

Métrique

Nous considérons en premier lieu une description résultant de la mise en relation entre une description individuelle et une description globale : le *niveau métrique* de chaque note est calculé à partir :

- De la signature métrique du morceau concerné (cf. 4.7.7.1.2).
- De l’instant de début de la note, exprimé en temps relatif modulo k, k étant un entier dépendant de la signature métrique du morceau (k=4 noires pour le 4/4).

La *structure métrique* se rapporte à la perception d’une alternance de battements forts et faibles [cf. par exemple Lerdahl & Jackendoff 1983 ; Drake 1990 ; Povel & Essens 1985]. Dans le cadre d’Imprology, nous prenons en compte simultanément la métrique binaire et la métrique ternaire, comme nous allons l’expliciter. Cette double prise en compte est souhaitable en particulier dans le cadre du traitement du corpus d’Owens, où ces deux métriques (et eux seuls) interviennent. (Indiquons que les concepts et techniques que nous proposons sont généralisables à d’autres métriques — par exemple, une métrique d’ordre 5). A cette fin, dans Imprology nous utilisons une subdivision de la mesure en un nombre d’unités (pulsations élémentaires ou *ticks*) multiple de 2×3. Pour avoir une quantification à la double-croche (binaire) et au triolet de double-croche (ternaire), ce nombre d’unités doit être au minimum de 48 pulsations par mesure 4/4, soit 12 par noire (triolets de triple croches). Imprology calcule alors séparément les niveaux métriques binaire (NMB) et ternaire (NMT) selon les schémas explicités, respectivement, par la Figure 4-35 pour le NMB et par la Figure 4-36 pour le NMT. Pour l’un comme pour l’autre, le niveau métrique d’une note quelconque est obtenu à partir de son instant de départ (modulo une mesure), traduit en un entier p compris entre 0 et 47 : il faut compter le nombre de lignes distinctes auxquels appartient cet instant de départ (nombre matérialisé, sur chacune des figures, par la longueur de la barre verticale située à la position p), et le niveau métrique est égal à ce nombre de niveaux moins 1.

```
            012345678901234567890123456789012345678901234567
N0 (1/48)   T|||||||||T|||||||||T|||||||||T|||||||||T|||||||
N1 (1/16)   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
N2 (1/8)    |     |     |     |     |     |     |     |
N3 (1/4)    |           |           |           |
N4 (1/2)    |                       |
N5 (mesure) |
```

Figure 4-35. Représentation des niveaux métriques binaires dans Imprology

```
            012345678901234567890123456789012345678901234567
N0 (1/48)   T|||||||||T|||||||||T|||||||||T|||||||||T|||||||
N1 (1/24)   | | | | | | | | | | | | | | | | | | | | | | | |
N2 (1/12)   |   |   |   |   |   |   |   |   |   |   |   |
N3 (1/6)    |       |       |       |       |       |
N4 (1/3)    |               |               |
N5 (mesure) |
```

Figure 4-36. Représentation des niveaux métriques ternaires dans Imprology

Pour la définition des fonctions de contribution spécialisées sur la description « niveau métrique », le paradigme de combinaison de NMB et NMT que nous proposons, et qui est satisfaisant en pratique, est le suivant : considérons deux notes n et n' ; la contribution — par exemple — du remplacement de n par n' fait intervenir :

$$\text{Min}(|\text{n.NMB} - \text{n'.NMB}|\, ,\, |\text{n.NMT} - \text{n'.NMT}|).$$

Cela signifie que la ressemblance entre deux notes, du point de vue de leur niveau métrique, est évaluée en fonction de celui (NMB ou NMT) sur lequel elles sont le plus semblables, d'où le Min. Concrètement, la valeur de ce Min peut être utilisée directement comme contribution de . (A titre d'information, la Figure 4-37 illustre l'imbrication des niveaux métriques binaire et ternaire).

```
                          012345678901234567890123456789012345678901234567
pulsations élémentaires   T|||||||||T|||||||||T|||||||||T|||||||||T|||||||
doubles croches triolets  | | | | | | | | | | | | | | | | | | | | | | | |
doubles croches           |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
croches triolets          |   |   |   |   |   |   |   |   |   |   |   |
croches                   |     |     |     |     |     |     |     |
noires triolets           |       |       |       |       |       |
noires                    |           |           |           |
blanches triolets         |               |               |
blanches                  |                       |
rondes (mesure)           |
```

Figure 4-37. Imbrication des niveaux métriques binaires et ternaires

La description « niveau métrique » permet notamment de prendre en compte la notion de *syncope* (dans sa définition donnée par exemple par Longuet-Higgins & Lee (1982)), que l'on peut avantageusement faire intervenir dans la définition de la saillance (cf. 4.7.4.2).

Par ailleurs, d'autres descriptions individuelles pertinentes peuvent être dérivées de cette description, par exemple la **direction** de la variation de niveau métrique entre chaque note et la précédente [Bakhmutova, Gusev & Titkova 1997] ou encore la **valeur** de cette variation.

Aspects variationnels

Nous considérons ensuite toute une catégorie de descriptions relationnelles importantes, se rapportant aux aspects variationnels dans les séquences musicales.

Sur la dimension de la hauteur, nous avons mentionné plusieurs fois la description individuelle dérivée 'Intervalle formé avec la note précédente' (ou avec la suivante), cet intervalle pouvant être soit **diatonique**, exprimé en nombre algébrique de degrés, soit chromatique, exprimé nombre algébrique de demi-tons.

Cela peut se généraliser de trois manières. Premièrement, la description individuelle peut être autre chose que la hauteur, par exemple la **durée** ou le **niveau métrique** (cf. plus haut). Deuxièmement, la variation peut être mesurée différemment que par une différence, par exemple par un **ratio** ou une **direction** (+/-). Troisièmement, la mise en relation peut être effectuée, comme nous l'avons dit d'emblée, avec une description **globale** ou **locale** plutôt qu'avec une description individuelle. Même avec ce dernier niveau de description, il est possible que la mise en relation concerne une autre note que la suivante ou la précédente : cela peut être en particulier la note située **deux**

éléments avant (« précédente de la précédente ») ou après, ce qui permet de pallier dans une certaine mesure l'effet indésirable qui fait que la modification d'une hauteur entraîne la modification de deux intervalles (celui précédant et celui suivant la note).

Nous choisissons de présenter ici en détail la description 'Ratio de durée' employée par Stammen & Pennycook (1993). Cette description individuelle est (rappel) calculée comme le quotient entre la durée de chaque note et la durée de la note qui précède, soit : X[i].durée / X[i+1].durée. Pour la première note, Pennycook et al. fixent arbitrairement[30] la valeur du ratio de durée (ainsi que celle de l'intervalle chromatique) à 0. Ainsi la séquence de ratios de durées redécrivant le passage du bas de la figure est : {0,1/3,1,1,1,1,1,2}.

Cette redescription permet une invariance aux distorsions locales de l'échelle du temps (*"local tempo changes"*). A titre d'exemple, la Figure 4-38 présente deux passages improvisés par Charlie Parker. Ces passages constituent d'ailleurs deux variantes d'un pattern du lexique proposé par Owens. (Pour plus de détails sur ce lexique de référence et son organisation, voir la section 4.4). Les ratios de durées sont indiqués sur la figure au-dessus et au-dessous des portées. La (plus longue) suite de ratios de durées commune aux deux passages est traduite par un formatage en gras italique. Bien que la durée des 5 notes concernées soit différente dans le premier et le second passage, les deux groupes sont considérés identiques au regard de la description ratio de durées. Ceci illustre la propriété d'invariance visée.

Figure 4-38. Redescription des durées en ratios de durée.

Rappelons en clôture que nous avons indiqué plus haut, à propos de la description « niveau métrique », d'autres méthodes, à fort potentiel, de prise en compte des aspects variationnels.

---

[30] Nous proposerons plus loin des moyens différents pour gérer ce genre d'exception (voir 5.5.2.2.3).

4.7.7.3.3. Prise en compte des silences en musique

Pour terminer, nous devons évoquer une description individuelle singulière qui n'entre de manière naturelle, ni dans le cadre des descriptions intrinsèques, ni dans celui des descriptions relationnelles : celle correspondant à la prise en compte des *silences* (pauses, soupirs, etc.). En informatique musicale, les silences sont souvent représentés de manière implicite ; en particulier, dans la norme Standard Midi File l'existence d'un silence est simplement traduite par un laps de temps non nul entre la fin d'une note et le début de la suivante. Nous avons au contraire choisi de les représenter explicitement, en tant que notes d'une nature particulière, ayant à la base un instant de début et une durée mais, par exemple, pas de hauteur. La « création » de ces notes particulières s'intègre dans la phase d'enrichissement de la représentation, l'algorithme spécifique étant trivial (parcours linéaire de l'ensemble notes de chaque mélodie du corpus, et, pour chacune, comparaison entre l'instant de fin de la note précédente et l'instant de début de la note). L'un des avantages significatifs de choix est la possibilité d'un traitement adéquat des appariements (substitutions) entre une note et un silence (resp. entre un silence et une note). Ce genre d'appariement, relativement fréquent (un exemple est donné au travers de l'alignement local sur la Figure 4-39 : appariement du la avec un silence de durée égale), ne peut être pris en charge de façon adéquate au moyen d'une délétion (resp. d'une insertion). La prise en compte des silences se traduit, entre autres, par une description individuelle supplémentaire 'noteVsSilence', permettant de marquer la différence entre notes et silences, et par des fonctions de contribution spécialisées se rattachant à cette description supplémentaire. Il faut préciser que la prise en compte explicite des silences en tant que notes de nature particulières se traduit par un changement (génerelement modéré) de la longueur des séquences et du corpus.

Figure 4-39. Appariement entre une note et un silence au sein d'un alignement local

4.7.8. Propriétés d'invariance

4.7.8.1. Notion de transformation

Nous appelons *transformation* toute fonction de Σ^* dans Σ^*. Les transformations qui nous intéressent ici agissent explicitement sur une ou quelques descriptions de base seulement : pour chaque élément, la transformation consiste en une modification particulière du ou des descripteur(s) concerné(s).

Exemple 4-1

Si X est une séquence, par exemple un historique d'indices boursiers, dont la représentation comprend la description numérique Taux, un exemple de transformation est la *translation verticale* d'une valeur constante τ, notée T_τ et définie par :

$$T_\tau : X=X[1]..X[L] \mapsto Y=Y[1]..Y[L]$$

avec :

$$\{\forall i\in \mathbf{N}_L\}\ \{Y[i].Taux = X[i].Taux + \tau\}$$

Bien entendu, si on représente graphiquement la séquence des valeurs de Taux comme une courbe dans le plan (Taux étant en ordonnée), T_τ correspond à la translation géométrique verticale, d'une longueur τ, de cette courbe. †

Par « explicitement », nous nous référions ci-dessus au fait que la/les description(s) apparaîsse(nt) effectivement dans l'algorithme de modification de chaque élément, comme c'est le cas pour Taux dans l'exemple. Néanmoins les effets d'une transformations peuvent ne pas se limiter aux valeurs de cette/ces description(s) explicite(s), en raison des liens de dépendance existant entre descriptions. Plus précisément, on peut énoncer en règle générale[31] la propriété suivante :

Propriété 4-3:

Si la transformation T agit sur une description D, alors elle agit aussi sur toute description D' dérivant de D.

---

[31] Les exceptions à la règle générale énoncée constituent justement des cas intéressants d'invariance, comme on le verra plus bas.

Exemple 4-2

En reprenant l'Exemple 4-1, les descriptions dérivant de Taux, comme celle caractérisant les maxima locaux (notamment la valeur atteinte au sommet), sont susceptibles d'être modifiées par T_τ. †

4.7.8.2. Définition et propriétés

Nous disons qu'une description D est invariante *à une transformation T si et seulement si la description d'une séquence quelconque X selon D n'est pas modifiée par T. Autrement dit, en désignant par Y la transformée de X par T (Y=T(X)) et par L la longueur de X :*

$$\{\forall i \in \mathbf{N}_L\} \ \{Y[i].D = X[i].D\}$$

Exemple 4-3

Reprenons l'exemple d'une description numérique D et la description dérivée Δ définie par (rappel):

$$\{\forall i \in \mathbf{N}_{L-1}\} \ \{X[i].\Delta = X[i+1].D - X[i].D\}$$

La description Δ est clairement invariante à toute translation verticale (selon Dnum) d'une valeur constante T_τ. Voici l'analogue dans le domaine musical : la description « Intervalle chromatique formé avec la note précédente » est invariante à une transposition chromatique d'un nombre quelconque de demi-tons. †

4.7.8.2.1. Intérêt

Lors du choix du jeu de description utilisé pour la comparaison de séquences, il est pertinent d'avoir à l'esprit les conséquences de ce choix en termes d'invariance descriptionnelle. Le fait que le calcul de similitude entre séquences soit « tolérant » à tel ou tel type de transformation a en effet une influence décisive sur la nature des patterns découverts. C'est ainsi que dans plusieurs systèmes [Cope 1991a ; Rowe 1993 ; Pennycook, Stammen & Reynolds 1993 ; etc.] les notes sont décrites en termes d'intervalles chromatiques entre paires de notes successives. L'invariance de cette description à toute transposition chromatique est avantageuse, car elle permet de rechercher des contours mélodiques ou mélodico-rythmiques récurrents sans se soucier de leur hauteur absolue.

4.7.8.2.2. Transitivité

Par ailleurs, il existe des règles de transitivité (ou de « propagation ») de l'invariance au sein du graphe de dépendance. Ces règles constituent d'une certaine manière la contraposée de celles énoncées dans la Propriété 4-3. On peut énoncer la propriété suivante :

Propriété 4-4

Si une description D est calculée à partir des descriptions D1, D2, ..., Dk (à l'exclusion de toute autre description) dont chacune est invariante à une même transformation Tr, alors D est aussi invariante à Tr.

Par exemple, les descriptions musicales « extremum local » ou « contour ascendant » sont toujours invariantes aux transpositions chromatiques.

4.7.8.3. Descriptions invariantes musicales et extra-musicales

Une propriété d'invariance se spécifie de manière naturelle par un couple {description ; transformation} (ou plus généralement : {type de description ; transformation}

Il est toujours possible d'énoncer un certain nombre de propriétés d'invariance {D ; T} qui n'ont aucun intérêt. Ce sont celles où la transformation T ne concerne la description D ni directement ni indirectement, c'est-à-dire que T n'agit ni sur D, ni sur aucun prédecesseur de D dans le graphe de dépendance. Ainsi, il n'est pas très intéressant d'affirmer que les durées ou les positions métriques des notes d'une séquence musicale sont inchangées lorsqu'on transpose celle-ci d'un nombre quelconque de degrés dans une tonalité…

Les deux tableaux qui suivent indiquent au contraire un certain nombre de couples description-transformation « intéressants ». Si le second tableau ne concerne que des dimensions musicales, le premier concerne des types de descriptions plus génériques.

Lorsqu'elle existe, nous indiquons pour chaque description D, en seconde colonne, la description primitive directe Dprim dont D est directement dérivée. En d'autres termes, D s'obtient à partir de Dprim sans avoir besoin des valeurs d'aucune autre description. Il est alors intéressant de noter que pour chaque description D de ce genre, l'invariance indiquée par le tableau constitue un cas d'exception à la règle générale énoncée en Propriété 4-3 : à chaque fois Dprim n'est pas invariante à la transformation citée, pourtant D, qui dérive directement de Dprim, y est invariante. Par exemple, la description Midipitch n'est pas invariante aux transpositions chromatiques, mais la description « Intervalle chromatique formé avec la note suivante », qui en dérive directement, y est invariante.

Tableau 4-10. Exemples de descriptions (générales) invariantes à différentes transformations. Lorsqu'une description dérive directement d'une autre, cette dernière description (qualifiée de primitive directe) est indiquée dans la seconde colonne.

Description	**Description primitive directe**	**Transformation invariante**
Variation (*différence*) de la valeur d'une quelconque description numérique D entre chaque élément et son successeur	La description D concernée	Translation verticale, selon D, d'une valeur constante τ
Variation (*ratio*) de la valeur d'une quelconque description numérique D entre chaque élément et son successeur	La description D concernée	Homothétie verticale, selon D, d'un facteur constant α
Ratio de durée (séquence temporelles)	Durée	Compression/dilatation subjective de l'axe du temps
Durées	—	Déplacement d'une durée quelconque

Tableau 4-11. Exemples de descriptions **musicales** invariantes à différentes transformations.

Description	**Description primitive directe**	**Transformation invariante**
Intervalle chromatique formé avec la note suivante	Midipitch	Transposition chromatique
Intervalle diatonique formé avec la note suivante	Degré dans une tonalité (locale/globale)	Transposition diatonique (relativement à la tonalité considérée)
Ratio de durée	Durée	Variation subjective du tempo (ex.: doublement)
Position métrique	—	Déplacement d'un nombre quelconque de mesures
Durée	—	Déplacement temporel d'une durée (constante) quelconque
Degré diatonique (dans une tonalité globale ou locale)	—	Transposition d'un nombre quelconque d'octaves

Remarque : le type d'invariance intervenant pour les descriptions « position métrique » et « degré diatonique » peut intervenir dans un domaine extra-musical. Il suffit pour cela qu'une description ait un caractère *cyclique* comme pour les positions métriques ou les degrés diatoniques. Deux exemples élémentaires d'une telle description sont le numéro du jour dans la semaine (cycle modulo 7) ou celui du mois dans l'année (cycle modulo 12). Il y aura alors invariance, respectivement, à un déplacement temporel d'un nombre quelconque de semaines et à un déplacement temporel d'un nombre quelconque d'années.

4.7.9. Considérations sur les tailles d'alphabet

L'enrichissement de la représentation tend à amener à un agrandissement de la taille de l'alphabet effectif (défini au 2.2.1.1), et a fortiori de celle de l'alphabet (total). La tendance est d'autant plus marquée que les différentes descriptions sont peu corrélées.

Indiquons à ce sujet qu'en biologie moléculaire, lorsqu'on décrit les monomères dans l'espace par 2 angles dihédraux (cf. 4.7.1), les alphabets géométriques résultants (issus d'un échantillonage / maillage à partir de la carte de Ramachandran [Ramachandran, Ramakrishnan & Sesisakharan 1963 ; Sagot et al. 1995b]) sont significativement plus grands que ceux employés pour représenter la structure primaire. Par exemple, pour une maille de 5°, l'alphabet dépasse 5000 symboles, à comparer avec 4 (alphabet des nucléotides) et 20 (alphabet des acides aminés).

La situation extrême — pouvant être rencontrée notamment sur des séquences musicales — est celle où il n'y a pas, au travers l'ensemble du corpus, un seul couple d'éléments ayant la même représentation. La taille de l'alphabet effectif est alors égale à la longueur cumulée L du corpus. Cette situation est systématiquement rencontrée dès lors que l'on adjoint à chaque élément la description D1=« numéro dans la séquence » et D2=« numéro de la séquence dans le corpus » afin, comme évoqué plus haut, d'introduire un biais favorisant l'extraction de patterns locaux à une séquence particulière du corpus et une zone particulière au sein de cette séquence. En effet les couples $\{(X[i].D1, X[i].D2), i \in \mathbf{N}_L\}$ sont manifestement distincts deux à deux, ce qui assure déjà que deux éléments ne peuvent être représentés par le même symbole.

Les conséquences seront évaluées à la fois lors de la présentation du fonctionnement de l'algorithme FlExPat (calculs des contributions d'appariement : section 5.3.8.3.4) que lors de la comparaison de la complexité de FlExPat avec celles des autres algorithmes généraux ou biologiques d'extraction de patterns dans les séquences (section 5.6.4).

4.7.10. Critères d’évaluation des alignements

Dans le cadre du modèle d’édition, l’évaluation de la similitude entre deux séquences est effectuée au travers de l’évaluation de la qualité du ou des meilleur(s) alignement(s) trouvé(s) entre elles.

L’approche la plus immédiate consiste à prendre, pour évaluation d’un alignement, sa valeur (cf. 3.2.4.2.7). Cependant, cette approche ne donne pas toujours de bons résultats, car il se peut que cela conduise à privilégier la longueur des alignements au détriment de leur « qualité » intrinsèque, cette dernière étant notamment liée à la proportion d’identités et de substitutions par rapport à celle des insertions et délétions. En vue d’éviter ce problème, nous avons proposé un certain nombre d’autres critères, qui sont détaillés ci-après.

4.7.10.1. Critères liés aux proportions respectives des différents types d’appariements

Une première idée en vue d’appréhender cette qualité intrinsèque est — par exemple dans le cas d’un modèle d’édition standard — de calculer la **proportion d’identités** au sein de l’ensemble des appariements constituant un alignement. Cette mesure a l’avantage d’être normalisée, prenant des valeurs comprises entre **0 et 1** ; la valeur 1 n’est atteinte que pour un **alignement à l’identique** (égalité de la séquence origine et de la séquence destination). On peut aussi d’ailleurs généraliser un peu ce schéma en calculant plutôt la proportion de **remplacements**. Cela dit, en pratique ces critères sont d’une utilité limitée par rapport à celui, préférable à de nombreux points de vue, du *ratio-identité*, voir ci-dessous.

4.7.10.2. Critère lié au « ratio-identité »

Avant de donner une définition précise du ratio-identité, donnons-en l'intuition sous-jacente. Il est raisonnable de poser comme principe que la meilleure qualité d'alignement est atteinte lorsqu'on a un alignement à l'identique (cf. 3.2.4.2.7, page 78). Pour juger de la qualité d'un alignement quelconque A, nous proposons alors de calculer le quotient entre la valeur de A et ce que serait cette valeur si A était un alignement à l'identique. Un tel ratio prend toujours des valeurs comprises entre $-\infty$ et 1, et vaut bien sûr 1 lorsque les deux séquences sont identiques — du moins au regard du jeu de descriptions employé[32]. Il est alors facile de juger de la qualité de A : on compare son ratio-identité à la valeur optimale 1.

Pour donner une définition précise et opérationnelle du ratio-identité, les choses se compliquent légèrement du fait que, pour tout alignement A qui n'est pas un alignement à l'identique, il existe en fait deux alignements à l'identique dérivant de A : celui, noté A', faisant intervenir sa séquence origine et celui, noté A", faisant intervenir sa séquence destination.

Définition 4-1

Etant donné un alignement A (et les alignements à l'identique A' et A" qui y sont associés, comme définis ci-dessus), son ratio-identité *est le réel défini par :*

$$ratioId(A)=valeur(A)/valeurId(A),$$

où valeurId(A) est un terme traduisant une valeur "idéale" pour A égale à la moyenne suivante :

$$valeurId(A)=0.5*(valeur(A')+valeur(A''))$$

Le jugement de la qualité de A fondé sur le critère "ratio-identité" s'effectue en comparant valeurId(A) à 1, sa valeur optimale.

4.7.10.3. Utilisation des différents critères

Les algorithmes d'alignement usuels comme ceux basés sur la programmation dynamique utilisent un critère bien précis d'évaluation des alignements intermédiaires (par exemple, les alignement entre **préfixes** des deux séquences comparées dans le cas de la comparaison **globale**) : la **valeur** de ces alignements intermédiaires, au sens du 3.2.4.2.7. Aussi il est nécessaire qu'il y ait adéquation entre ce critère-là et celui qui permet d'élire le meilleur alignement au total — alignement dont l'évaluation indiquera le degré de similitude entre les deux séquences comparées.

Ainsi dans le cas où on utilise un algorithme de type AG2SPD ou AL2SPD (cf. 3.2.6), les critères liés aux proportions respectives des différents types d'appariements ne sont pas utilisables directement. En revanche, dans le cas où la comparaison est

---

[32] A titre de commentaire sur la borne (infinie) négative, rappelons que la fonction de similitude ou de dissimilitude associée à un modèle d'édition valué peut prendre des valeurs négatives.

globale (AG2SPD), le critère ratio-identité est parfaitement utilisable. Il s'avère particulièrement adéquat dans le cadre de la famille d'algorithmes FlExPat (voir le Chapitre 5 et en particulier la présentation des résultats expérimentaux obtenus).

Chapitre 5. EXTRACTION DE PATTERNS — L'ALGORITHME FLEXPAT

5.1. Motivation

L’objet de cette première section est de mettre en évidence la nécessité qui nous est apparue de concevoir une nouvelle approche générale pour l’extraction de patterns dans les séquences.

Un certain nombre d’algorithmes, généraux ou à visée explicitement biologiques, ont été passés en revue au 3.4.1. Dans tous ces algorithmes intervient, implicitement ou explicitement, des critères déterminant si un facteur F est occurrence d'un pattern Pat. Dans tous les cas, ces critères sont sous-tendus, explicitement ou implicitement, par un modèle de comparaison entre entités séquentielles. Nous avons vu que l'équipollence est définie dans certaines approches d'extraction de patterns par une relation d'égalité, éventuellement généralisée au travers d'une relation d'équivalence ou d'une relation de similitude non transitive sur l'alphabet. Elle est définie dans d'autres sur la base d'une distance de Hamming valuée ou non, dans d'autres encore à l'aide d'un modèle d'édition non valué avec types d'éditions standard.

Or, compte tenu de tout ce qui a été dit et montré au Chapitre 4, nous souhaitions pouvoir effectuer l’extraction de patterns en utilisant un modèle de comparaison plus riche, en l’occurrence le modèle d'édition valué multi-descriptions (MEVM, cf. section 4.7). Aucun des algorithmes cités ne permet d’utiliser un modèle d'édition valué, encore moins un modèle d'édition valué à contributions (ou coûts) non constants — catégorie dans laquelle s’inscrit le MEVM.

En ce qui concerne les algorithmes à visée exclusivement musicale qui ont été présentés, l'un d'entre eux, à savoir l'algorithme de Pennycook, Stammen & Reynolds (1993), utilise certes un modèle d'édition valué à coûts non constants (cf. 3.2.5.2.2). Cependant, outre le fait que le modèle d'édition employé soit totalement fixe — faisant intervenir un jeu immuable de (deux) descriptions équipondérées — la nature même de l'algorithme rend son utilisation directe impossible dans le contexte où nous nous plaçons. En effet, une caractéristique majeure de l'algorithme — liée à son caractère temps-réel — est d'être fondé sur un mécanisme de **segmentation** (en temps-réel) de la séquence source (cf. 5.6.5). Si les résultats obtenus sont excellents du point de vue d'une utilisation au sein d'un système musical interactif[33] [Rowe 1993], en revanche l'aspect combinatoire qui caractérise l'analyse musicale orientée-patterns telle que nous souhaitons l'automatiser ou la simuler, y est en bonne partie occulté[34]. Il faut ajouter que la nécessité du mécanisme de segmentation limite le potentiel d'application de l'algorithme aux domaines où une telle segmentation est à la fois justifiée et susceptible d'être effectuée de manière satisfaisante par un algorithme.

Nous avons donc conçu et implémenté une nouvelle approche combinatoire, qui :

- offre la possibilité d'utiliser un modèle d'édition valué à contributions (ou coûts) non constants tel que le MEVM.
- permet d'une manière générale, au-delà des caractéristiques souhaitées dans le cadre d'application musicales, de rendre compte de la richesse combinatoire inhérente au problème de l'extraction de patterns, au même titre que les différents algorithmes généraux ou biologiques présentés plus haut.

5.2. FlExPat : caractéristiques générales

Notre approche algorithmique en matière d'extraction de patterns [Rolland 1998] se nomme FlExPat : *Flexible Extraction of Patterns*.

Il s'agit en toute rigueur d'une **famille** d'algorithmes, l'extraction dans un corpus unitaire et l'extraction dans un corpus multiple, en particulier, n'employant pas exactement les mêmes algorithmes[35]. Par commodité toutefois, nous parlons de

---

[33] Nous avons pu en juger en ayant eu l'opportunité d'assister à une démonstration du système (sur cassette vidéo).

[34] A cet égard, le paroxysme est atteint avec Cypher, où la combinatoire sur les segments se réduit aux paires de phrases **successives** (cf. 3.4.2.2).

[35] Pour être tout à fait précis, l'algorithme prévu pour un corpus multiple peut naturellement être employé sur un corpus unitaire, un tel corpus pouvant toujours être vu comme un cas particulier de

"l'algorithme FlExPat" à chaque fois qu'il est question des (nombreuses) caractéristiques communes à l'ensemble des algorithmes de la famille.

FlExPat a une portée tout à fait générale, son champ d'application ne se restreignant pas à un domaine particulier comme la musique ou la biologie.

L'entrée principale est un corpus de séquences sources, éventuellement unitaire. Il y a deux phases algorithmiques principales (cf. Figure 5-1). La première, dite *phase d'appariement de facteurs*, produit une structure de données intermédiaire désignée par *graphe de similitude*, graphe simple[36], non-orienté, dont chaque sommet représente un facteur et chaque arête représente un lien d'équipollence entre les deux facteurs correspondants (cf. Définition 2-3). La seconde, appelée *phase de catégorisation*, permet de passer du graphe de similitude à un ensemble de patterns.

Figure 5-1. FlExPat: schéma d'ensemble

Dans le présent chapitre, il est souhaitable de marquer très nettement la distinction entre les aspects intensionnel et extensionnel des patterns tels que nous les avons définis précédemment. Aussi désignerons-nous ici par *représentant* ou *prototype* d'un pattern P sa représentation intensionnelle (que nous appellions simplement « pattern » plus haut). La représentation extensionnelle de P, relativement au corpus C considéré, est (rappel) l'ensemble des occurrences de P dans C, un tel ensemble de facteurs étant également appelé *bloc* (cf. 2.2.3.3).

Le modèle de comparaison de séquences pour lequel FlExPat est conçu est le modèle d'édition valué multi-descriptions (MEVM) proposé au 4.7. Plus précisément l'équipollence de deux facteurs est déterminée sur la base de leur similitude dans une instance donnée de ce modèle, c'est-à-dire sur la base de la valeur du meilleur alignement entre les deux facteurs. Comme, du fait de sa généralité, le MEVM subsume plusieurs autres modèles de comparaison, FlExPat peut émuler un certain nombre

---

corpus multiple. Cela dit il est plus efficace, d'un point de vue de complexité algorithmique, d'utiliser l'algorithme dédié au traitement d'un corpus unitaire.

[36] La terminologie que nous adoptons relativement aux concepts de base sur les graphes est celle proposée par exemple dans [Gondran & Minoux 1979].

d'algorithmes d'extraction fondés sur ces autres modèles, comme nous le verrons au 5.6.2.

5.3. Phase d’appariement de facteurs

5.3.1. Description simplifiée

Voici une description simplifiée de l'algorithme accomplissant la phase d'appariement de facteurs, désigné par FlExPat-AF. Quoiqu'inexacte, cette description aura l'avantage de donner une intuition de l'algorithme avant de rentrer dans les détails. Chaque facteur $C[[u\,;i,\,m]]$ du corpus[37] va faire l'objet d'un traitement individuel consistant à calculer successivement sa similitude avec tous les facteurs $C[[u'\,;i'\;\,m']]$ situés à sa gauche, c’est-à-dire tels que $i'\leq i$ (ce qui assure en particulier que deux facteurs ne sont jamais comparés plus d'une fois). Après chaque calcul de similitude, l’équipollence du couple de facteurs est testée, et si ce test est positif une arête est créée entre les deux sommets représentant les deux facteurs. Si l'on a choisi de créer un graphe de similitude valué, c’est-à-dire dont chaque arête F—F' est étiquetée avec le nombre Simil(F,F'), on attribue à l'arête $C[[u\,;i\,,m]]$—$C[[u\,;i'\,,m']]$ la valeur de la similitude entre ces deux facteurs. Le fait central est que chaque calcul de similitude entre deux facteurs F et F' réutilise les valeurs de similitudes déjà calculées (et stockées) entre d'autres paires de facteurs. Concrètement, ces derniers facteurs sont des préfixes — éventuellement vides — de F et F'. Grâce à cela, chaque calcul de similitude se fait **en temps constant**[38] au lieu de prendre un temps proportionnel au produit des longueurs des deux facteurs (cf. 3.2.6, page 90).

5.3.2. Nature de la simplification

Il s'agit dans cette section de spécifier en quoi cette description est simplifiée et par suite inexacte. Quatre précisions doivent être apportées, ayant toutes en commun de résulter en une limitation très nette, en pratique, de la complexité de l'algorithme : complexité en temps et en espace pour les trois premières, complexité en espace essentiellement[39] pour la dernière.

---

[37] La notation employée est celle introduite au 2.2.3.1.2, page 56.

[38] En théorie, suivant l’ensemble des types d’appariements autorisés que l’on emploie, ce temps peut ne pas être constant. En pratique, il l’est, comme nous le verrons au 5.3.8.1.

[39] Conduisant à une réduction du nombre de sommets du graphe de similitude, cette dernière précision permet généralement, en conséquence, de réduire la complexité en temps de l'algorithme de catégorisation.

❶ En pratique il n'est pas nécessaire de considérer tous les facteurs de toutes les longueurs, loin s'en faut. Cela est particulièrement évident lorsqu'on a affaire à un corpus unitaire : chercher à découvrir des patterns de longueur presque égale à celle de la séquence source ne peut avoir de sens que dans le cas très particulier où l'on s'intéresse à des patterns "auto-répétitifs", c'est-à-dire présentant eux mêmes des répétitions internes ou des périodicités. (Dans le cas où les représentants de patterns recherchés sont des facteurs du corpus, ces facteurs sont au plus de même longueur que la/les plus longue(s) séquence(s) source). Concrètement, on se donne deux entiers m_{min} et m_{max} fixant les longueurs minimale et maximale des facteurs qui seront pris en compte, c'est-à-dire les facteurs candidats au statut d'occurrence — voire de représentant — de pattern. Dans le cadre de notre corpus musical de référence, la longueur des patterns recherchés, soit en gros entre 4 et 25, est inférieure d'un ordre de grandeur (au moins) à la longueur des séquences du corpus, soit quelques centaines à quelques milliers. De plus la longueur des patterns est en quelque sorte indépendante de celles des séquences sources.

❷ Il n'est en pratique ni indispensable, ni pertinent de comparer — en vue de tester leur équipollence — des facteurs de longueurs très différentes. Sur la base des résultats obtenus par Owens, la longueur des occurrences de patterns musicaux qu'Imprology cherche à extraire est certes variable mais dans des proportions très modérées : par exemple, une différence de longueur d'au plus 2 pour un pattern de longueur 9, typiquement. En conséquence, nous nous donnons une fonction entière Δm fixant la différence de longueur entre deux facteurs au-delà de laquelle on ne testera pas leur équipollence. Arbitraire dans certains cas, le filtrage des paires de facteurs résultant de la fonction Δm possède au contraire, dans d'autres, une justification théorique objective. Une discussion détaillée à ce sujet sera proposée au 5.3.6.; fixons néanmoins l'idée générale, en nous restreignant à $\text{ETAA}=\text{ETAA}_{\text{standard}}$ et à des contributions constantes. Notons -d la contribution strictement négative d'une délétion, et supposons par exemple que F' soit plus court que F, en d'autres termes que m'<m. Tout alignement — global — entre F et F' fait intervenir au moins m-m' délétions (d'éléments de F). Il est possible (quitte à prendre d élevé) que la contribution cumulée de ces (m-m') délétions, soit -(m-m')×d, dégrade systématiquement la valeur de l'alignement au point que Eq(F,F') soit toujours égal à FAUX. Dans ces conditions, il est clair qu'une utilisation de Δm permet d'éviter un calcul de similitude plus un test d'équipollence inutiles sans pour autant risquer d'omettre à tort la création d'un lien dans le graphe.

❸ Les liens d'équipollence entre des paires de facteurs présentant des recoupements étendus n'ont usuellement aucun intérêt dans la perspective de l'extraction de patterns. Par exemple, savoir que le préfixe, de longueur f-1, d'un facteur F de longueur f, lui est équipollent ne sera pas d'une grande utilité pour déterminer ultérieurement si F peut être représentant (resp. occurrence) de pattern, surtout si la stratégie employée pour la catégorisation est du type de celle que nous introduisons au 5.4.2. Nous nous fixons une fonction booléenne Recoup sur l'ensemble des facteurs du corpus, avec pour tout couple de facteurs (F,F') : Recoup(F,F')=VRAI si et seulement si F et F' présentent un recoupement jugé suffisamment étendu. Une approche efficace consiste à fixer le pourcentage maximal de recoupement (en longueur) autorisé entre les deux facteurs candidats à la comparaison (voir 5.3.3.1). Là aussi, un nombre potentiellement très grand de calculs de similitude et de tests d'équipollence sont évités sans aucune conséquence sur la qualité des résultats.

❹ Dans la lignée de ce que nous avons dit à l'item précédent, la création de deux liens de similitude ayant même origine et dont les facteurs-extrémités présentent un recoupement étendu, comme illustré sur la Figure 5-2, peut s'avérer au mieux redondante et au pire problématique (cf. 5.4.3). Une manière possible d'exploiter cela au plan algorithmique est d'introduire une fonction jugeant l'étendue des recoupements ; en pratique, la même fonction Recoup (cf. ci-dessus) peut-être utilisée. Au moment de créer une arête entre F et F', s'il existe déjà une arête F→F'' et que F' et F'' présentent un recoupement trop étendu, alors un choix sera fait entre les deux arêtes, une seule étant conservée dans le graphe de similitude.

Figure 5-2. Illustration d'un recoupement (zone hachurée) entre facteurs-extrémités d'arêtes

Remarque : les quatre points qui précèdent peuvent être vus comme spécifiant autant de *filtres* sur facts(C)^2, éliminant des couples de facteurs sur la base de critères spécifiques :

❶ Longueur des facteurs candidats au statut d'occurrence ou de représentant de pattern

❷ Différence de longueur entre facteur origine et facteur destination

❸ Etendue du recoupement entre facteur origine et facteur destination

❹ Etendue du recoupement entre facteurs-extrémités d'arêtes

Il est naturel, au moins pour les trois premiers, que ces filtrages soient effectués en ligne, c'est-à-dire au cours de la phase d'appariement de facteurs. C'est pourquoi Δm et Recoup sont introduites en tant qu'entrées de FlExPat-AF. On peut toutefois envisager que, dans des contextes très spécifiques et au prix d'une augmentation sensible de la complexité algorithmique notamment en espace, tout ou partie des filtrages ne soient effectués qu'au moment de la phase de catégorisation. Dans ce cas, Δm et/ou Recoup constitueront plutôt des entrées de l'algorithme de catégorisation.

5.3.3. Entrées-sorties principales

L'épithète "principales" tient à ce que, pour la clarté de l'exposé, nous ferons ici un certain nombre d'omissions, comblées ultérieurement, par rapport aux entrées. (En particulier, nous passerons sous silence les paramètres régissant les mécanismes, qui seront vus au 5.4.3, de gestion des redondances dans le graphe de similitude. Ces mécanismes, lorsqu'ils sont mis en œuvre dès la phase d'appariement de facteurs, en constituent des paramètres d'entrée au même titre que les fonctions Δm et Recoup.)

5.3.3.1. Entrées principales

L'entrée principale est bien sûr le corpus C de séquences sources, éventuellement unitaire. Par ailleurs, un certain nombre d'autres entrées peuvent être vues comme des paramètres de l'algorithme :

- L'ensemble ETAA des types d'appariements permis, avec leurs fonctions de contribution associées. Le cas standard est naturellement celui où ETAA est $\text{ETAA}_{\text{standard}}$={Insertion, Délétion, Remplacement}. Le fonctionnement de FlExPat-AF tel qu'il est décrit ici nécessite la symétrie des fonctions de contribution employées (cf. 3.2.5.2), condition fort naturelle et, en pratique, très fréquemment vérifiée. Elle l'est en particulier pour toutes les fonctions de contribution qui seront mises en œuvre dans les expérimentations présentées dans ce mémoire.

- Une fonction d'équipollence Eq permettant de décider si un arc peut être créé entre deux sommets/facteurs donnés (*test d'équipollence*). Typiquement, il s'agit d'un seuillage sur la valeur de l'alignement optimal trouvé entre les deux facteurs (cf. 3.2.3), ou sur le ratio-identité (cf. 4.7.10.2) de cet alignement optimal.

- Deux entiers m_{min} et m_{max} fixant la longueur minimale (resp. maximale) des facteurs candidats au statut d'occurrence ou de représentant de pattern. En pratique, m_{max} est le plus souvent très inférieure à la longueur globale du corpus, et même à la longueur moyenne des séquences du corpus. L'existence de ces deux paramètres, et en particulier celle de m_{max}, est importante car elle permet une réduction très efficace de la complexité algorithmique temporelle. Néanmoins, que ce soit dans une perspective théorique ou dans un domaine d'application très particulier, il est possible de n'imposer ni borne inférieure ni borne supérieure à la taille des patterns, en fixant $m_{min}=1$ et $m_{max}=L$.

- Une fonction entière Δm fixant la différence maximale de longueur entre deux facteurs au-delà de laquelle on choisit de ne pas tester leur équipollence. Ainsi, lors du traitement (voir plus bas) du facteur F de longueur *m*, l'équipollence de F ne sera testée qu'avec des facteurs F' dont la longueur m' vérifie :

  $$|m-m'| \leq \Delta m(m)$$

  Equation 5-1

  De même que pour les valeurs de m_{min} et m_{max} par rapport à L, en pratique Δm prend des valeurs faibles devant *m*, ce qui permet là aussi une limitation efficace de la complexité en temps de l'algorithme. Une forme pratique et efficace donnée à Δm est celle d'une fonction pseudo-linéaire :

  où α est un réel compris entre 0 et 1 traduisant, en pourcentage, l'écart maximum de longueur entre F et F'. Évidemment, plus α est proche de 1 et plus des écarts importants sont autorisés et pour $\alpha=1$ l'effet de Δm est annulé. A l'inverse, toutes choses égales par ailleurs, plus α est proche de 0 et plus il devient probable que l'usage de Δm soit de nature heuristique au sens strict du terme [Shapiro 1992], c'est-à-dire risque d'empêcher la création de liens d'équipollence pourtant valides.

- Une fonction booléenne Recoup sur facts(C)² testant si deux facteurs présentent un recoupement d'une étendue suffisante pour que l'on choisisse délibérément de ne pas tester leur équipollence. Définir cette fonction requiert à la fois de spécifier la manière dont l'étendue d'un recoupement est calculée et la façon dont on juge le résultat de ce calcul. Un choix efficace consiste à définir l'étendue du recoupement entre deux facteurs F et F' comme le quotient entre la longueur du facteur (éventuellement vide) matérialisant ce recoupement, et la longueur du plus court parmi F et F' ; ce quotient est alors comparé à un seuil ρ. En notant F^F' le facteur-recoupement, on a l'égalité booléenne suivante :

Dans l'exemple de la Figure 5-3, on a |F|=7, |F'|=5, |F^F'|=3 et min(|F|,|F'|)=5. Avec une valeur typique pour ρ de 1/3, on obtient un quotient de 3/5 > 1/3 donc Recoup(F,F') = vrai.

Figure 5-3. Exemple de recoupement entre deux facteurs candidats à la comparaison

Remarque terminologique : nous désignons par *configuration* [d'une instance de l'algorithme FlExPat-AF] l'ensemble des valeurs des différentes entrées qui viennent d'être décrites. Cela nous servira dans la suite, notamment pour préciser les conditions expérimentales spécifiques à chaque exécution de FlExPat décrite en section 5.5.

5.3.3.2. Sortie:

Il s'agit du graphe de similitude Gsimil ayant les caractéristiques suivantes :

- Les sommets représentent l'ensemble des facteurs qui ont été déterminés équipollents, au sens de la fonction Eq, à au moins un autre facteur.
- Les arêtes représentent l'ensemble des liens d'équipollence entre facteurs.
- Orientation : la fonction de similitude étant symétrique, Gsimil peut être pris non-orienté.
- Valuation : usuellement, il est intéressant de tenir compte, lors de la phase de catégorisation, non seulement de l'existence de liens de ressemblance entre facteurs, mais aussi de la qualité de chacun de ces liens, c'est-à-dire de la valeur de la similitude entre les deux facteurs concernés. A cette fin, Gsimil est alors valué avec ces similitudes numériques.

5.3.4. Principe

L'algorihme FlExPat-AF correspondant à la phase d'appariement de facteurs ("AF") s'appuie sur des techniques de programmation dynamique pour la comparaison globale de paires de séquences (cf. 3.2.6.1).

Dans un premier temps, la séquence globale S est calculée (concaténation du corpus). Puis les facteurs du corpus susceptibles d'être des représentants ou des occurrences de patterns sont traités dans un ordre spécifique, l'ordre lexicographique sur les facteurs que nous définissons ci-dessous (5.3.4.1). Le traitement d'un tel facteur, appelé par commodité *facteur origine*, consiste à calculer sa similitude puis tester son équipollence avec un ensemble d'autres facteurs — *facteurs destination*[40] — pris en compte eux aussi dans un ordre spécifique. A chaque fois qu'un test d'équipollence est positif, i.e. que Eq(F,F')=VRAI, une arête est créée dans le graphe, moyennant éventuellement la vérification de certaines propriétés évitant des redondances dans le graphe de similitude (cf. 5.4.3). Lorsqu'un facteur origine F et un facteur destination F' présentent un recoupement significatif (c'est-à-dire lorsque Recoup(F,F') vaut VRAI), on se contente de calculer leur similitude sans tester leur équipollence. De même, d'autres facteurs qui ne sont pas potentiellement candidats au statut de représentant ou d'occurrence de pattern font également l'objet de comparaisons sans test d'équipollence, comme cela sera détaillé au 5.3.4.4.

Il nous faut maintenant décrire l'ordonnancement l'algorithmique, à savoir :

- Quels facteurs sont traités, et à quels autres facteurs ils sont comparés.
- L'ordre dans lequel sont traités et comparés les facteurs

C'est cette organisation spécifique qui permet que chaque calcul de similitude se fasse en temps constant.

Pour la clarté d'exposition, nous supposons dans cette section que *ETAA*=$\text{ETAA}_{\text{standard}}$. La généralisation à un ensemble plus grand ou différent de types d'appariements sera traitée au 5.3.5.

---

[40] Le choix des appelations "facteur origine" et "facteur destination" est commode dans la mesure où, lorsqu'on les compare, l'un est la *séquence source* et l'autre la *séquence destination* (voir plus haut) de l'alignement optimal dont la valeur fournit la similitude entre les deux facteurs.

5.3.4.1. Ordre lexicographique sur les facteurs d'une séquence

Définition 5-1.

Soient X une séquence, F=X[[i,m]] et F'=X[[i',m']] deux facteurs de X. F' est lexicographiquement inférieur *(strictement) à F si et seulement les couples d'entiers (i',m') et (i,m) satisfont l'inégalité lexicographique :*

$$(i',m')<(i,m)$$

c'est-à-dire si et seulement si l'une des conditions suivantes est vérifiée :

- $(i'<i)$
- $(i'=i)$ *ET* $(m'<m)$

Cet ordre, que nous baptisons LexCoord, est indépendant du "contenu" des facteurs, puisqu'il ne prend en compte que leurs coordonnées. Il est donc distinct de l'ordre lexicographique LexContenu pouvant éventuellement exister sur les facteurs et basé, lui, sur leur contenu. En particulier, pour des séquences sur l'alphabet latin LexContenu est l'ordre lexicographique naturel selon lequel, par exemple, ababb < abbaa.

5.3.4.2. Ordonnancement

L'ordre dans lequel sont considérées les paires de facteurs exploite entre autres la symétrie supposée de la fonction de similitude (qui induit la symétrie du graphe). S désignant la séquence globale, les facteurs origine sont traités dans l'ordre lexicographique qui vient d'être défini. Il s'agit de tous les facteurs de longueur m_{min} à m_{max} et qui sont compris — lexicographiquement parlant — entre S[[1, m_{min}]] et S[[L - m_{min} + 1, m_{min}]] à l'exception des facteurs mixtes, c'est-à-dire chevauchant deux séquences, qui doivent évidemment être écartés.

Remarque : nous évoquions plus haut le fait que FlExpat est une famille d'algorithmes plus qu'un algorithme unique. Cela devient clair au niveau de l'ordonnancement des paires de facteurs : quoique l'algorithme tel que nous le décrivons dans la suite puisse traiter un corpus unitaire comme un cas particulier (en prenant C={S1}, N=1), il est plus naturel et plus efficace d'employer plutôt un autre algorithme fondé sur les mêmes principes. Celui-ci, qui se dérive facilement de l'algorithme décrit ici, tire parti des spécificités d'un corpus unitaire (en particulier, pas de prise en compte nécessaire des facteurs mixtes).

Sont d'abord traités les facteurs de rang 1, par ordre croissant de longueur en allant de m=m_{min} à m_{max}, sauf si cela provoque un débordement au-delà de l'extrémité de S1 (i.e. si $m_{max}>|S1|$), auquel cas on s'arrête juste avant que le débordement ne se produise. Sont ensuite traités les facteurs de rang 2 ; pour ceux-ci il y a débordement dès lors que $m>|S1|-1$. Etc. Lorsque le rang $|S1|-m_{min}+1$ est atteint, un seul facteur est traité: celui de longueur m_{min}.

On passe alors au facteur non mixte immédiatement supérieur à ce dernier facteur. Il s'agit de ($|S1|+1$, m_{min}), et on fait croître la longueur jusqu'à ce que l'on atteigne le facteur ($|S1|+1$, min(m_{max}, $|S2|$)).

Le parcours se poursuit jusqu'à ($|S1|+\ldots+|S_N|-m_{min}+1$, m_{min}), c'est-à-dire ($L-m_{min}+1$, m_{min}).

La Figure 5-4 énumère la succession des coordonnées des facteurs traités.

[Facteurs de la 1e séquence source :]

(1, m_{min}), (1, $m_{min}+1$), … , (1, m_{max}) — ou (1, $|S1|$) si $m_{max}>|S1|$
(2, m_{min}), (2, $m_{min}+1$), … , (2, m_{max}) — ou (2, $|S1|-1$) si $m_{max}>|S1|-1$

…

($|S1|-m_{min}+1$, m_{min})

[Facteurs de la 2e séquence source :]

($|S1|+1$, m_{min}), ($|S1|+1$, $m_{min}+1$), … ($|S1|+1$, m_{max}) — ou ($|S1|+1$, $|S2|$) si $m_{max}>|S2|$
($|S1|+2$, m_{min}), ($|S1|+2$, $m_{min}+1$), … ($|S1|+2$, m_{max}) — ou …

…

($|S1|+|S2|-m_{min}+1$, m_{min})

…

[Facteurs de la N-ième séquence source :]

($|S1|+\ldots+|S_{N-1}|+1$, m_{min}), ($|S1|+\ldots+|S_{N-1}|+1$, $m_{min}+1$), … ($|S1|+\ldots+|S_{N-1}|+1$, m_{max}) — ou
($|S1|+\ldots+|S_{N-1}|+1$, $|SN|$) si $m_{max}>|SN|$
($|S1|+\ldots+|S_{N-1}|+2$, m_{min}), ($|S1|+\ldots+|S_{N-1}|+2$, $m_{min}+1$), … ($|S1|+\ldots+|S_{N-1}|+2$, m_{max}) — ou …

…

($|S1|+\ldots+|S_N|-m_{min}+1$, m_{min})

Figure 5-4. Succession des facteurs origine

Notons $F= S[[i,m]]$ le facteur origine devant être traité à l'instant courant.

F va être comparé avec l'ensemble des facteurs non mixtes, de longueur m_{min} à m_{max}, situés [strictement] à gauche de F et non *filtrés* sur la base de Δm ou sur celle de Recoup. (Ces facteurs destination lui sont donc nécessairement inférieurs au sens de LexCoord). Ils sont donc compris entre S[[1, m_{min}]] et S[[i-1, $m_{max}+\Delta m(m)$]] dans le cas général. (Les cas d'exception se produisent notamment lorsque le facteur (i-1, m_{max}) est mixte, auquel cas on prend à sa place le plus long facteur non mixte de rang i-1). Cet ensemble de facteurs destination est non vide dès lors que i>1.

Pour désigner la similitude entre les facteurs $F=S[[i,m]]$ et $F'=S[[i',m']]$, c'est-à-dire la valeur maximale d'un alignement entre F et F', nous introduisons la notation . (Nous employons Sim plutôt que Simil qui désigne la fonction de similitude elle même : =Simil(F,F')). Cette notation matricielle a l'avantage de faire apparaître les deux facteurs en vis à vis horizontal, comme dans l'alignement dont cette notation représente la valeur. Cette notation garde sa validité si F ou F' est vide (c'est-à-dire si m=0 ou m'=0), tout alignement entre F et F' étant dans ce cas constitué exclusivement d'insertions (resp. de délétions). Notons que, compte tenu des hypothèses, on a, pour tout quadruplet (i,i',m,m') :

$$=$$

Supposons dans un premier temps que ETAA=ETAA$_{standard}$ (le cas général sera traité au 5.3.5), et énonçons la propriété centrale sur laquelle repose FlExPat-AF. Elle exprime la similitude de la paire de facteurs {S[[i,m]], S[[i',m']]} en fonction de la similitude entre d'autres paires de facteurs plus courts (mais de mêmes rangs i et i') :

Propriété 5-1

Pour tout couple de facteurs {(i,m), (i',m')} on a :

Equation 5-2

(Par rapport à l'Equation 4-1, page 134, noter que le terme constant 0 a disparu de la maximisation).

Avant de donner ci-dessous la preuve de cette propriété, deux observations importantes :

- Lorsque m=1 ou m'=1, l'Equation 5-2 fait intervenir des valeurs singulières de Sim avec une ou les deux longueurs nulle(s), valeurs qui sont calculées lors d'une phase d'initialisation (cf. 5.3.4.5).
- Les trois valeurs de Sim intervenant dans la maximisation ont toutes été déjà calculées en raison de l'ordonnancement mis en œuvre. En effet, cette maximisation intervient lors de la comparaison du facteur origine F=S[[i,m]] avec le facteur destination F'=S[[i',m']]. Or :

    q correspond à la comparaison entre le facteur origine S[[i,m-1]] et le facteur destination S[[i',m'-1]]. Le fait que (*i*, *m*-1)< (*i*, *m*) suffit à garantir que S[[i,m-1]] a été traité avant F.

    q correspond à la comparaison entre S[[i,m-1]] et S[[i',m']]. Là aussi, le fait que (*i*, *m*-1)< (*i*, *m*) suffit à garantir que S[[i,m-1]] a été traité avant F.

    q Enfin, correspond à la comparaison entre S[[i,m]] et S[[i',m'-1]]. Ce qui assure que cette valeur de Sim a déjà été calculée est, cette fois-ci, l'inégalité lexicographique existant entre les [coordonnées des] facteurs destinations : (*i'*, *m'*-1)< (*i'*, *m'*).

- Aussi, pour autant que ces 3 valeurs aient été mémorisées après avoir été calculées, la maximisation dans l'Equation 5-2 se fait en temps constant : accès à ces 3 valeurs ; 3 additions ; 3 calculs de contribution ou, suivant les cas, accès à des contributions précalculées — cf. 5.3.8.3). A noter qu'il n'est (heureusement!) pas nécessaire de stocker toutes les valeurs Sim au fur et à mesure qu'elles sont calculées. On peut au contraire, en mettant en œuvre une stratégie dynamique, n'en avoir à chaque instant en mémoire qu'un nombre relativement faible, comme nous le verrons au 5.3.8.2.

5.3.4.3. Preuve de la propriété

L'alignement optimal entre *S*[[*i*, *m*]] et *S*[[*i'*, *m'*]], c'est-à-dire celui de plus haute valeur, est désigné pour les besoins de cette preuve par AOpt(*i*,*m*,*i'*,*m'*). Il se termine, soit par un remplacement, soit par une délétion, soit par une insertion. (Il est possible d'avoir deux alignements optimaux ex-aequo, voire trois. Sans rien oter à la validité de cette preuve, nous nous plaçons toutefois dans la situation où l'alignement optimal est unique). Supposons par exemple que ce soit un remplacement. Alors AOpt(*i*,*m*,*i'*,*m'*) est composé de la concaténation entre :

- un alignement entre *S*[[*i*, *m*-1]] et *S*[[*i'*, *m'*-1]], que nous désignons par A(*i*, *m*-1, *i'*, *m'*-1), et :
- le remplacement de l'élément d'indice *i*+*m*-1 de S par celui d'indice *i'*+*m'*-1,

et l'on a :

$$= \text{valeur}(A(i, m-1, i', m'-1)) + \mathit{contrib}$$

Equation 5-3

Cela permet d'affirmer que A(*i*, *m*-1, *i'*, *m'*-1) est optimal, et donc, par définition de la notation Sim, que :

$$= \text{valeur}(A(i, m-1, i', m'-1))$$

Equation 5-4

(En effet, en raisonnant par l'absurde : s'il existait un alignement A' entre *S*[[*i*, *m*-1]] et *S*[[*i'*, *m'*-1]] strictement meilleur que A(*i*, *m*-1, *i'*, *m'*-1), alors la concaténation de A' avec serait un alignement entre *S*[[*i*, *m*]] et *S*[[*i'*, *m'*]] strictement meilleur que la concaténation de A(*i*, *m*-1, *i'*, *m'*-1) avec . Or cette concaténation n'est autre que AOpt(*i*, *m*, *i'*, *m'*), lequel ne serait donc pas optimal : *contradiction* avec les hypothèses).

La confrontation entre Equation 5-3 et Equation 5-4 conduit alors à :

$$= \mathit{contrib} \qquad\qquad +$$

Nous montrerions de manière semblable que si la succession optimale d’appariements entre $S(i, m)$ et $S(i', m')$ se termine par une délétion alors on a :

$$= \mathit{contrib} \qquad\qquad +$$

et que si elle se termine par une insertion alors on a :

$$= \mathit{contrib} \qquad\qquad +$$

Cela complète la preuve.

5.3.4.4. Comparaisons additionnelles

Un certain nombre de calculs de similitudes supplémentaires sont nécessaires. Il s'agit de pures comparaisons, sans test d'équipollence, de paires de facteurs entre lesquels on ne souhaite donc pas créer de liens de similitude dans le graphe. Ce sont les paires vérifiant l’une des conditions suivantes :

- l'un au moins des deux facteurs est de longueur strictement inférieure à m_{min}.
- la différence de longueur, lors de la comparaison du facteur origine F=S[[i,m]] avec le facteur destination F’=S[[i’,m’]], entre F et F’, dépasse Δm(m), ou encore Recoup(F,F’)=VRAI.

Détaillons par exemple la nécessité d’un tel calcul de similitude lorsque la première de ces deux conditions est vérifiée : dans l'Equation 5-2, lorsque l'une au moins des longueurs apparaîssant dans le membre de gauche est égale à m_{min}, la maximisation exige de connaître au moins une valeur de similitude faisant intervenir un (ou deux) facteur(s) de longueur m_{min}-1. Or, dans le cas général, pour comparer une paire comprenant un tel facteur, il faut connaître la similitude d'une paire de facteurs dont l'un est de longueur m_{min}-2, et ainsi de suite. Ainsi, en l'absence d'heuristiques particulières, à l'intervalle $\{m_{min}, .. , m_{max}\}$ des longueurs de facteurs à comparer (cf. 5.3.4.2) doit être adjoint l'intervalle $\{1,..,m_{min}-1\}$ correspondant aux comparaisons supplémentaires dont il est question ici.

Indiquons que le nombre de comparaisons supplémentaires peut être réduit heuristiquement. Nous nous contenterons ici de donner l'idée générale d'une telle heuristique. Il s'agit, suivant les valeurs d'indices, de restreindre la maximisation figurant dans l'Equation 5-2 à une maximisation de 2 termes. Cela revient à interdire les successions d'insertions (resp. de délétions) terminales, c'est-à-dire situées en fin d'alignement, au-delà d'un seuil — nombre maximal d'insertions (resp. de délétions) consécutives — que l'on s'est fixé.

5.3.4.5. Initialisation

L'initialisation de FlExPat-AF consiste essentiellement à calculer les valeurs de pour m et/ou m' nuls. A première vue, il faut donc calculer :

- L×(L-1)×(m_{max}+1) valeurs[41] pour
- L×(L-1)×(m_{max}+1) valeurs pour
- L×(L-1)/2 valeurs pour

En réalité :

- (pour m'=0 :) ne dépend pas de i'. Nous le noterons donc , et la relation de récurrence :

$$= \mathit{contrib} \qquad +$$

permet de calculer pour les L×m_{max} valeurs de (i,m)

---

[41] Plutôt que L×(L-1)×(m_{max}-m_{min}+1), en raison des nécessaires comparaisons additionnelles (cf. 5.3.4.4).

- (pour m=0 :) ne dépend pas de i. Nous le noterons donc , et la relation :

$$= contrib \qquad +$$

permet de calculer pour les L×m_{max} valeurs de (i',m')

- (pour m=m'=0 :) ne dépend, ni de i, ni de i', et vaut toujours 0.

5.3.5. Généralisation à un ensemble de types d’appariements plus grand ou différent

Il est aisé de modifier la description ci-dessus pour accomoder un ensemble de types d'appariements différents (cf. 3.2.4.2.2). Le seul[42] changement sensible se situe au niveau de la maximisation intervenant dans l'Equation 5-2. Pour chaque type d'édition que l'on ajoute (resp. que l'on retire), un certain nombre de lignes est ajouté (resp. retiré) dans la maximisation. Ce nombre de lignes est égal à 1 si le type d'appariement concerné est à signature fixe, et au cardinal de sa signature variable dans le cas contraire.

La présentation de l'exemple typique ci-après est plus parlante qu'un exposé général qui ferait intervenir une généralisation complexe et abstraite de l'Equation 5-2. La généralisation à un ETAA quelconque à partir de l'exemple présenté est immédiate.

Exemple 5-1

On souhaite passer d'ETAA$_{\text{standard}}$ à ETAA'={Insertion, Délétion, Remplacement, Permutation, Fragmentation, Consolidation}, avec les signatures variables suivantes :

- l'ensemble {(1,2), (1,3)}[43] pour le type Fragmentation
- l'ensemble {(2,1), (3,1)}) pour le type Consolidation

L'Equation 5-2 devient alors :

---

[42] Sauf si ETAA contient les types InsertionMultiple et DélétionMultiple, auquel cas des modifications mineures sont susceptibles d'invervenir au niveau de l'initialisation.

[43] Le cas d’ensembles de signatures dépendant de m et/ou de m’ sera envisagé au 5.3.8.1.

Equation 5-5

Naturellement, dans l'Equation 5-5, pour certaines valeurs des indices certains termes (lignes) seront omis dans la maximisation. Par exemple, si m=m'=2 :

- il est impossible que l'alignement optimal entre les deux facteurs se termine par une fragmentation de signature (1,3). Le 6ème terme n'entre donc pas en considération.
- cet alignement optimal ne peut pas non plus se terminer par une consolidation de signature (3,1). Le dernier terme est donc omis.

A l'extrême, si m=m'=1 l'alignement optimal entre les deux facteurs ne peut se terminer ni par une permutation, ni par une fragmentation, ni par une consolidation; ainsi la maximisation se ramène dans ce cas à celle de l'Equation 5-2. †

5.3.6. Caractère heuristique ou non de l'emploi de Δm

L'emploi de Δm est, suivant les cas, de nature heuristique ou non. Il l'est lorsqu'il empêche, à tort, la création d'au moins un lien d'équipollence. C'est-à-dire lorsqu'il existe au moins deux facteurs *F* et *F'* dont l'équipollence n'est pas testée en raison d'un écart de longueur trop grand mais qui satisferaient cependant au test d'équipollence. A l'extrême, en prenant pour Δm la fonction nulle, toute paire de facteurs de longueurs différentes est écartée du test.

En revanche, l'emploi de Δm n’est pas de nature heuristique quand un écart de longueur supérieur à la valeur prise par Δm empêche déjà, à lui seul, que les deux facteurs puissent être jugés équipollents. Illustrons ce second cas de figure :

Exemple

Nous nous plaçons dans les conditions suivantes :

- $ETAA=ETAA_{standard}$.
- Les fonctions de contribution sont constantes et valent :
  - -1 pour Insertion et Délétion
  - +1 pour Identité, -1 pour Substitution
- La fonction Eq est un seuillage sur le ratio-identité (cf. 4.7.10.2) de l'alignement optimal avec Seuil = 0.7.
- $(\forall m\in \mathbf{N}_L)$ $(\Delta m(m)=0.2*m)$.

Soient F et F' deux facteurs de longueurs respectives m et m' avec par exemple $m'<m$ (sans perte de généralité). L’équipollence de F’ avec F ne sera pas testée dès lors que :

$$(m-m')>\Delta m(m)$$

C’est-à-dire dès lors que :

$$m'<0.8*m$$

Equation 5-6

Or il est clair que tout alignement entre F et F' fait intervenir au moins (m-m') délétions (d'éléments de F). Dès lors, la valeur de l'alignement optimal est au plus égale à m'-(m-m'), correspondant à m' identités et (m-m') délétions. Par suite le ratio-identité vaut au plus :

dans la mesure où l'alignement idéal entre F et lui-même (m identités) a pour valeur m et celui entre F' et lui-même a pour valeur m'. Compte tenu de l'Equation 5-6 ce quotient est majoré par :

Cela montre que toute paire (F,F') éliminée sur la base de Δm ne satisferait de toutes façons pas au test d'équipollence, et ainsi Δm n'a pas un caractère heuristique "strict". †

5.3.7. Algorithme proprement dit

Une version en pseudo-code du corps principal de l'algorithme FlExPat-AF est donnée à la Figure 5-5. Par souci de clarté et de concision :

- Nous choisissons ETAA = $\text{ETAA}_{\text{standard}}$.
- La gestion des recoupements entre facteurs-extrémités d'arêtes (5.4.3) n'est pas faite en-ligne — autrement dit, n'est pas intégrée à FlExPat-AF.
- Nous n'intégrons pas les mécanismes d'optimisation proposés au 5.3.8.2 pour la mémorisation des valeurs de similitude ni ceux proposés au 5.3.8.3 pour le calcul et la mémorisation des contributions d'appariements.

Précisons ou rappelons par ailleurs que :

- les paramètres entiers **mmin**, **mmax** et les fonction **Δm,** **Recoup,** et **Eq** (fonction équipollence), sont ceux correspondant à la configuration de FlExPat-AF (cf. 5.3.3.1).
- **u** (variable entière) représente la numéro d'ordre, au sein du corpus, de la séquence (**Su**) en cours de traitement. **|Su|** désigne la longueur de Su.
- **RgInit(u)** (entier) est le rang du premier élément de Su dans la séquence globale **S** du corpus. (Pour des raisons d'efficacité, les valeurs de cette fonction sont préalablement stockées dans un tableau)

- **i** et **i'** (variables entières) représentent les rangs des facteurs en cours de traitement, **m** et **m'** (variables entières) leurs longueurs respectives.
- **S[[i,m]]** est le facteur de rang i et de longueur m dans la séquence globale **S** du corpus.
- (réel) est la similitude entre les facteurs S[[i,m]] et S[[i',m']]

5.3.8. Complexité

(*N.B. :* un résumé des différents résultats relatifs à la complexité de FlExPat-AF présentés dans les sections qui suivent, sera donné au 5.3.8.4)

5.3.8.1. En temps

Nous évaluons la complexité en temps de FlExPat-AF en déterminant le nombre total NbAdd d'additions intervenant dans la maximisation, cf. Equation 5-2 et Equation 5-5. (Comme il y a un accès à une valeur Sim stockée en mémoire pour une addition et vice versa, il revient au même de compter le nombre d'accès et le nombre d'additions ; aussi nous contenterons-nous dans la suite de parler de NbAdd).

Les dénombrements liés au calcul de la contribution des différents appariements (appels des fonctions de contribution associées à chacun des types d'appariements) intervenant au cours de l'algorithme seront effectués au 5.3.8.3. Nous verrons que même en employant la stratégie la plus coûteuse (qui, en pratique, n'est pas utilisée), la complexité temporelle correspondant à l'ensemble des calculs de contributions est du même ordre de grandeur que NbAdd. En employant la stratégie dite "composite" (celle utilisée en pratique), le nombre de calculs de contributions est petit devant NbAdd.

En ce qui concerne les tests d'équipollence, leur nombre est par définition inférieur ou égal au nombre de comparaisons, l'égalité se produisant lorsque toute paire de facteurs dont on calcule la similitude fait l'objet d'un test d'équipollence ($\Delta m = \infty$). Ainsi dans ce cas limite le nombre total de tests d'équipollence est égal au nombre total NbSim de calculs de valeurs Sim. La forme de l'Equation 5-2 (ou de l'Equation 5-5) montre que NbSim est égal à NbAdd divisé par le nombre de lignes de la maximisation. Cela entraîne que, quel que soit le paramétrage (Δm notamment) la contribution (en termes de nombre de tests) des tests d'équipollence ne changera pas l'ordre de grandeur fourni par NbAdd.

```
Algorithme FlExPat-AF

Début
    Initialisation
    Pour u ← 1 à N faire
        Pour i ← RgInit(u) à RgInit(u)+lSul faire

            Pour m ←1 à min{mmin, (RgInit(u)+lSul-i+1)}-1 faire      {"Il faut que i+m-1 <= RgInit(u)+lSul"}
                {"Simples calculs de similitude (pas de test d'équipollence), quel que soit le facteur destination : "}
                Pour u' ← 1 à u faire
                    Pour i'← RgInit(u') à min(RgInit(u')+lSu'l, i-1) faire
                        Pour m' ← 1 à min{mmax, (RgInit(u')+lSu'l-i'+1)} faire

                                                                                        (E1)

                        FinPour {"m' "}
                    FinPour{"i' "}
                FinPour {"u' "}
            FinPour {"m"}

            Pour m ← min{mmin, (RgInit(u)+lSul-i+1)} à min{mmax, (RgInit(u)+lSul-i+1)} faire
                {"Il faut que i+m-1 reste <= RgInit(u)+lSul pour assurer la non-mixité du facteur origine S[[i,m]]"}
                Pour u' ← 1 à u faire
                    Pour i'← RgInit(u') à min(RgInit(u')+lSu'l, i-1) faire

                        Pour m' ← 1 à min{mmax, (RgInit(u')+lSu'l-i'+1)} faire
                            {"Remarque : Le min fait bien intervenir mmax et non m+Δm(m), cf. 5.3.4.4.
                            Par ailleurs, il faut que i'+m'-1 reste <= RgInit(u')+lSu'l
                            pour assurer la non-mixité du facteur destination S[[i',m']]"}
                                    ← comme dans (E1) ci-dessus

                            {" test d'équipollence, sauf si filtrage dû à Δm ou à Recoup"}
                            Si (m' ≥ m - Δm(m)) et (m' ≤ m + Δm(m)) et non(Recoup(S[[i,m]],S[[i',m']])) alors
                                {"Pas de test supplémentaire,
                                la gestion des recoupements (voir section 5.4.3) n'étant pas effectuée en-ligne"}
                                Si Eq(S[[i,m]],S[[i',m']]) alors
                                    Créer une arête entre S[[i,m]] et S[[i',m']]
                                FinSi
                            FinSi
                        FinPour {"m' "}
                    FinPour{"i' "}
                FinPour {"u' "}
            FinPour {"m"}

        FinPour {"i"}
    FinPour {"u"}
Fin
```

Figure 5-5. FlExPat : corps principal de l'algorithme dans le cas où l'ensemble des types de transformations est $ETAA_{standard}$={Insertion, Délétion, Remplacement}

Au total, il est légitime d'évaluer la complexité temporelle de FlExPat-AF en calculant NbAdd. Nous allons en donner un majorant en faisant abstraction des mécanismes de filtrage (cf. Remarque du 5.3.2) basés sur le recoupement :

- entre facteur origine et facteur destination
- entre facteurs-extrémités d'arêtes.

Considérons le calcul d'une valeur Sim pour des valeurs d'indices "normales", c'est-à-dire pour lesquelles le nombre de lignes dans l'équation de maximisation est égal au nombre, noté ||ETAA|| dans cette section, d'appariements terminaux possibles. (Autrement dit, aucun terme n'est omis dans la maximisation pour cause d'indices, cf. 5.3.5). ||ETAA|| est supérieur à |ETAA| dès lors qu'ETAA comporte des types d'appariement à signature variable. Par exemple, dans l'Equation 5-5 le nombre d'appariements effectifs (nombre de termes intervenant dans la maximisation) est de 8, alors que |ETAA|=5. En théorie, ||ETAA|| peut dépendre des valeurs de m et m' (cf. Equation 5-5) ; par exemple, on peut faire intervenir dans la maximisation (m-1) consolidations de signatures respectives (2,1), (3,1), … , (m,1). En pratique néanmoins, cela n'est pas nécessaire[44], et ||ETAA|| est majoré par une constante indépendante de m, m' et de L [Mongeau & Sankoff 1990].

Pour des valeurs d'indices "normales" donc, le nombre d'additions intervenant dans le calcul d'une valeur Sim est clairement égal à ||ETAA|| (et serait inférieur si des termes étaient omis dans la maximisation). Aussi pouvons-nous majorer NbAdd par ||ETAA||×NbSim. Or nous pouvons donner la majoration suivante pour NbSim (nombre de facteurs origine traités × nombre de facteurs, "strictement à gauche" — cf. 5.3.4.2) :

$$\text{NbSim} \leq 1/2 \times \text{L} \times (\text{L}-1) \times (m_{max})^2$$

Au total, **la complexité en temps de FlExPat-AF**, estimée par le nombre total d'additions intervenant dans l'équation d'optimisation et en l'absence de toute heuristique **peut être grossièrement majorée par** $\mathbf{O(L^2 \times m_{max}{}^2 \times ||ETAA||)}$.

5.3.8.2. En espace

L'exécution de FlExPat-AF exige le stockage :

---

[44] Quel sens aurait par exemple, sur des séquences mélodiques tonales au sens large (classique, jazz, etc.), la fragmentation d'une note — même très longue — en 100 autres ?

- "permanent", du graphe de similitude
- temporaire, de valeurs de similitude
- temporaire également, de contributions d'appariements. Cette mémorisation-là sera envisagée au 5.3.8.3. De manière analogue à ce nous avons vu pour la complexité en temps, nous verrons que, quelle que soit la stratégie employée, le stockage des contributions d'appariements n'entraîne pas de changement dans l'ordre de grandeur de la complexité totale en espace de FlExPat-AF.

En ce qui concerne les : dans FlExPat-AF, il n'est pas nécessaire de conserver au fur et à mesure les NbSim valeurs qui sont calculées, ce qui pourrait — compte tenu de ce que nous avons vu précédemment — exiger un espace en O(m_{max}^2.L²). Déjà, lors du traitement d'un quelconque facteur origine F=S[[i,m]], toutes les valeurs avec k≠i sont inutiles compte tenu de la forme de l'Equation 5-2 (ou de l'Equation 5-5). De plus toutes les comparaisons de F avec les facteurs de rang i'<i (et de longueur 1, .. m_{max} dans le cas général) n'utilisent que des valeurs de la forme , où les astérisques traduisent des valeurs quelconques des indices. Ainsi, sans compliquer exagérément la gestion de valeurs Sim mémorisées, on n'a besoin à tout moment que de stocker **au plus m_{max}^2 valeurs**, qui sont de la forme avec i et i' fixés (i'<i) et (m,m') ∈ $\mathbf{N}m_{max}^2$; Ces valeurs pouvant être oubliées dès que l'on passe au i' suivant (ou a fortiori au i suivant).

En ce qui concerne le graphe de similitude Gsimil, l'espace mémoire requis dépend évidemment de tous les paramètres d'entrée de l'algorithme, y compris du choix de gérer en ligne ou en différé les recoupements entre facteurs-extrémités d'arêtes (section 5.4.3). Aux deux extrêmes :

- G est vide si, par exemple, Eq est la fonction constante FAUX
- G est le graphe complet dont les sommets représentent tous les facteurs de longueur comprise entre m_{min} et m_{max}, si, par exemple, aucun mécanisme de filtrage (hormis Δm) n'est activé et Eq est la fonction constante VRAI. Ce graphe possède environ L×(m_{max}-m_{min}+1) sommets et (1/2)×L²×(m_{max}-m_{min}+1)² arêtes.

Il nous est clairement impossible de donner une estimation générale, en fonction des différentes entrées, de l'espace requis pour stocker Gsimil. Aussi nous contenterons-nous d'indiquer qu'au travers des différentes expérimentations que nous avons effectuées il est apparu que, pour des valeurs sensées de paramètres, l'ordre[45] de Gsimil était bien inférieur à sa valeur maximale théorique . Le Tableau 5-1 présente des résultats expérimentaux typiques[46] sur des corpus mélodiques multiples dont les longueurs L sont comprises entre 50 et 5000. La 3e ligne du tableau indique les valeurs prises par L^2 en vue de souligner la petitesse du nombre d'arêtes par rapport au majorant $(1/2)\times L^2\times(m_{max}-m_{min}+1)^2$.

L	**50**	**500**	**5000**
$\lvert Gsimil \rvert$	48	647	10731
L^2	*2500*	*250000*	25.10^6
Nombre d'arêtes	91	993	78564

Tableau 5-1. Ordre et nombre d'arêtes du graphe de similitude

5.3.8.3. Calcul et mémorisation des contributions d'appariements

Nous avons laissé pour ce paragraphe les considérations relatives au calcul des contributions d'appariements parce que, suivant la stratégie adoptée pour ce calcul (calcul en-ligne, précalcul et stockage intégraux, ou stratégie composite), les influences relatives sur les complexités en temps et en espace sont totalement inversées.

(N.B. : La présente discussion n'a pas lieu d'être dans le cas singulier (dégénéré) où les différentes fonctions de contributions sont toutes constantes. Dans ce cas, en effet, parler de "calcul" pour les contributions des appariements individuels n'a pas de sens, et il n'y a rien d'autre à stocker que les valeurs des constantes.)

5.3.8.3.1. Calcul en ligne

La première stratégie possible, la plus simple, consiste à calculer la contribution d'un appariement lorsque celui-ci se présente, c'est-à-dire lorsque sa contribution

---

[45] C'est-à-dire son nombre de sommets.

[46] Pour les 3 longueurs de corpus, les paramètres communs sont :

l ETAA=$ETAA_{standard}$

l Fonction d'équipollence : définie par un seuillage à la valeur 0.7 sur le ratioIdentité. Autrement dit, Eq(F,F') ⇔ l'alignement optimal entre F et F' a un ratioIdentité supérieur ou égal à 0.7.

l Fonction Δm : linéaire, définie par , de sorte que, par exemple, $\Delta m(10)=3$ et $\Delta m(9)=2$.

l Fonction Recoup : définie par un seuillage (cf. 5.3.3.1) avec $\rho = 1/3.001$.

intervient dans l'un des termes de la maximisation (Equation 5-2 ou Equation 5-5). Il est tout à fait clair que la complexité additionnelle est alors :

- exactement égale, en temps, à celle calculée au 5.3.8.1, puisque dans la maximisation un appariement différent est associé à chaque accès à une valeur Sim.
- nulle en espace (pas de stockage)

Ainsi, pour cette stratégie, l'influence marginale (i.e. spécifique) du calcul des contributions d'appariements résulte schématiquement en un doublement de la complexité en temps, la complexité en espace restant inchangée.

Il est aisé de voir que cette façon de faire occasionne de multiples redondances dans les calculs de contributions d'appariements. En effet dans le cas général — c’est-à-dire sauf pour des valeurs très particulières d'indices — un appariement app intervient dans le calcul d'une multitude de valeurs Sim différentes. Il s'agit de toutes les valeurs Sim pour lesquelles :

- le facteur origine admet comme suffixe la séquence origine de app
- le facteur destination admet comme suffixe la séquence destination de app

Considérons par exemple un corpus unitaire de longueur 1000 et fixons m_{min}=10, m_{max}=20. La substitution intervient dans les calculs de toutes les paires "valides" de facteurs F,F' dont les rangs terminaux sont respectivement 200 et 100. (Cette validité signifie que F est strictement à gauche de F' et que les longueurs de F et F' sont comprises entre 1 et m_{max}.). Même en se limitant aux paires de facteurs donnant lieu à un test d'équipollence — c’est-à-dire dont la longueur est au moins m_{min} — cela représente un total de 121 calculs de similitude, se répartissant de la manière suivante :

- pour i=181 et m=20 : 11 calculs, se décomposant comme suit :
    - pour i'=81 et m'=20 : 1 calcul
    - pour i'=82 et m'=19 : 1 calcul
    - …
    - pour i'=91 et m'=10 : 1 calcul

- pour i=182 et m=19 : 11 calculs, se décomposant comme suit :
    - pour i'=81 et m'=20 : 1 calcul
    - …
    - pour i'=91 et m'=10 : 1 calcul
- …
- pour i=191 et m=10 : 11 calculs, se décomposant comme suit :
    - pour i'=81 et m'=20 : 1 calcul
    - …
    - pour i'=91 et m'=10 : 1 calcul

Cela signifie que, moyennant cette évaluation (partielle!), est recalculé inutilement 120 fois.

Ces recalculs augmentent sans nécessité effective la complexité en temps de FlExPat-AF. D'autant plus que le temps requis pour le calcul est proportionnel au nombre de descriptions (cf. Equation 4-5) ce qui peut résulter globalement en une augmentation sensible du temps de calcul. D'où l'idée d'un stockage de ces valeurs dès qu'on les a calculées.

5.3.8.3.2. Précalcul et stockage intégraux

La seconde stratégie, diamétralement opposée à la première, vise à éviter tout recalcul, en précalculant et en stockant les contributions de tous les appariements qui interviendront pendant l'exécution de FlExPat-AF. Contentons-nous d'envisager le cas de figure où $ETAA=ETAA_{standard.}$ Nous faisons l'hypothèse que, comme c'est le cas pour les applications musicales (cf. 4.7.9), la taille de l'alphabet est au moins du même ordre de grandeur que la taille totale du corpus (le cas d'un petit alphabet sera envisagé au 5.3.8.3.4). L'ensemble des contributions dont on a besoin se divise en trois parties :

- Les contributions des différents remplacements. Elles peuvent commodément être stockées sous forme d'une matrice L×L. Il n'est nécessaire que de calculer une demi-matrice, en vertu des hypothèses de symétrie sur les fonctions de contribution. En ce qui concerne la diagonale, son calcul est inutile en pratique :

les éléments diagonaux représentent des appariements entre un élément et lui-même, et n'interviennent que dans les calculs de similitude de paires de facteurs ayant le même dernier élément, c'est-à-dire des facteurs préfixe/suffixe l'un de l'autre. Or les mécanismes de gestion des recoupements tels que ceux présentés au 5.3.3.1 écartent systématiquement ce genre de paires des calculs de similitude.

- Celles des différentes insertions, au nombre de L.
- Celles des différentes délétions, au nombre, elles aussi, de L.

Au total, l'ensemble peut être organisé sous forme d'une matrice $(L+1)\times(L+1)$ dont la première ligne concerne les insertions et la première colonne, les délétions.

Si cette stratégie permet que tout recalcul soit évité, en revanche la taille de l'espace requis peut s'avérer problématique : $O(L^2)$ pour $ETAA=ETAA_{standard}$, davantage pour des ETAA plus riches.

Aussi, combinant les deux stratégies qui viennent d'être décrites, nous en proposons une troisième qui représente un bon compromis entre espace requis et évitement de redondances.

5.3.8.3.3. Stratégie composite

Le stockage intégral des contributions n'est en fait pas nécessaire. Une observation simple permet de s'en convaincre : compte tenu de l'ordonnancement mis en œuvre dans FlExPat-AF, lorsque dans l'algorithme on atteint le traitement de F=S[[i,m]] et que l'on compare celui-ci avec un certain nombre de facteurs de rang i'<i (i' étant fixé) et de longueur croissante, il est inutile de connaître la contribution des appariements « situés à gauche de i et i' ». Par là nous entendons tous les appariements dont :

- le rang terminal de la séquence origine est < i.
- le rang terminal de la séquence destination est < i'.

En effet, dans toutes les comparaisons ultérieures (jusqu'à la fin de la phase d'appariement de facteurs) tous les appariements qui interviendront auront des séquences origine (resp. destination) de rang au moins i (resp. au moins i').

Supposons ici encore, pour la clarté de l'exposé, que :

- le corpus C est unitaire (longueur : L)

- $ETAA=ETAA_{standard}$
- par rapport à L et m_{max}, i est suffisamment faible pour qu'aucun facteur de longueur 1, … m_{max} ne « sorte » de C, autrement dit : $i+m_{max}-1 \leq L$.
- par rapport à m_{max}, la différence entre i et i' est suffisante pour qu'il n'y ait à aucun moment recoupement entre le facteur origine F et les facteurs destination

Alors, pour effectuer toutes les comparaisons entre F et les facteurs destination S[[i',m']] il suffit de connaître les contributions de :

- m_{max} délétions :

, …

- ($i+m_{max}-1$) insertions :

, …

- $m_{max} \times (i+m_{max}-1)$ remplacements, que l'on pourrait énumérer très simplement en « combinant » les séquences origines des délétions listées ci-dessus avec les séquences destinations des insertions également listées.

De plus, dans l'ordonnancement algorithmique le passage de i à i+1 nécéssite le calcul de la contribution d'**une** insertion et d'**une** délétion et de ($i+\boldsymbol{m_{max}}$) remplacements, et permet "**l'oubli**" de la contribution d'une délétion et de m_{max} remplacements. Au total, pour effectuer tous les calculs de similitude relatifs au facteur origine S[[i,m]] il n'est nécessaire de connaître que $((m_{max}+1) \times (i+m_{max}-1))+m_{max}$ contributions, disons $(m_{max}+1)(i+m_{max})$ pour simplifier. Cette expression est fondée sur les hypothèses faites plus haut quant à i et i', et sa valeur maximale est en réalité (i=L) : $m_{max}+L+(m_{max} \times L)$ soit $\mathbf{O}(\mathbf{L} \times \boldsymbol{m_{max}})$.

Cette observation, qui se généralise aisément à un ETAA quelconque, conduit sans difficulté particulière à la définition d'une stratégie composite. Nous ne détaillerons pas celle-ci davantage, nous contentant d'indiquer que, du point de vue de la complexité en temps, elle permet d'éviter tout recalcul avec un espace mémoire requis bien moindre que pour le stockage intégral.

5.3.8.3.4. Considérations relatives à la taille de l'alphabet

Une remarque s'impose quant à la taille des alphabets sur lesquels on travaille. Comme nous l’avons évoqué au 4.7.9, nous nous situons dans un cadre où la prise en compte de multiples descriptions peut conduire à des alphabets de très grande taille — la taille de l’alphabet effectif pouvant atteindre celle du corpus. Il est dont souhaitable d'indexer les différents appariements sur les séquences sources et non sur l'alphabet. En revanche, lorsqu'on a affaire à un alphabet de taille [fixe et] petite, ce qui est notamment le cas pour la description usuelle en biologie (4 ou 20 caractères — cf. 2.1.1.1), il y a mieux à faire que de stocker les valeurs de tous les appariements effectifs si $ETAA=ETAA_{standard}$. Il est dans ce cas préférable de stocker après avoir précalculé (s'il y a lieu) les contributions des différents appariements en indexant ces derniers sur l'alphabet, prenant la forme d'une matrice à une dimension (indels) ou deux dimensions (remplacements). Ainsi, en biologie, les contributions des substitutions entre les différents acides aminés sont des valeurs standard, stockées dans des matrices (20×20) qui ont été calculées une fois pour toutes sur la base des substitutions observées dans des groupes spécifiques de protéines homologues (cf. par exemple [Dayhoff, Schwartz & Orcutt 1978 ; Dayhoff, Barker & Hunt 1983 ; Henikoff & Henikoff 1992]).

5.3.8.4. Résumé

Nous avons donc vu que :

- La complexité en temps de FlExPat-AF, évaluable par le nombre total d'additions effectuées dans les maximisations (Equation 5-2 et Equation 5-5), est — très largement — **majorable** par $\mathbf{O(L^2 \times mmax^2 \times ||ETAA||)}$. Pour des valeurs de m_{max} petites devant L, ce qui comme nous l'avons vu est valide dans un certain nombre de contextes d'application, on peut dire que FlExPat-AF a une complexité temporelle quadratique (avec un facteur constant potentiellement important). Cela est, bien sûr, d'autant plus vrai que l'ensemble des types d'appariements autorisés est peu étoffé — par exemple $ETAA=ETAA_{standard}$, pour lequel $||ETAA||=|ETAA|=3$

- Sa complexité en espace est la somme :

  - d'un terme en $O(m_{max}^2)$ lié au stockage des valeurs Sim

  - d'un terme éventuel lié au stockage des contributions d'appariements, terme qui, moyennant l'emploi d'une stratégie composite (évitant toute redondance) pour le calcul de ces dernières, peut être ramené à O(L×m_{max}) avec ETAA=ETAA$_{standard}$

- d'un terme lié au stockage du graphe de similitude, et dont nous ne pouvons donner qu'une estimation expérimentale et une majoration très grossière — jamais même approchée en pratique — de O(L×(m_{max}-m_{min}+1)) pour les sommets et L^2×(m_{max}-m_{min}+1)^2 pour les arêtes.

Même lorsqu'on emploie les stratégies les plus coûteuses pour le stockage des valeurs Sim et des contributions d'appariements, la somme des deux termes peut être (grossièrement) **majorée** par **O(L^2×m_{max}^2)**

Soulignons que la complexité, tant en temps qu'en espace, est **indépendante de la taille de l'alphabet**, ce qui n'est pas le cas pour des algorithmes comme HMoivre ou LeMoivre (basés sur la notion de modèle).

5.4. Phase de catégorisation

5.4.1. Introduction

Cette seconde phase de l'algorithme FlExPat prend en entrée le graphe de similitude issu de la première phase et fournit en sortie des patterns caractérisés par un représentant, ou prototype, et par (les coordonnées d')un ensemble d'occurrences.

N.B. Il n'est pas exclu que plusieurs représentants puisse être donnés de manière équivalente pour un même pattern, quoiqu'en pratique (cf. 5.5.2) cela soit extrèmement rare[47]. Dans la suite, nous ne considérerons qu'un seul représentant par pattern — quitte à ce qu'un *choix* soit implicitement ou explicitement opéré entre plusieurs alternatives).

Généralement, il est intéressant d'accompagner chaque occurrence de la valeur de sa similitude avec le représentant. Un ensemble de post-traitements peuvent être effectués ensuite sur la base des patterns extraits, comme des calculs statistiques (nombres d'occurrences, localisation type etc.).

Cette extraction est appelée catégorisation dans la mesure où les patterns sont vus comme des catégories dont on s'efforce de donner un prototype. Divers travaux ont concerné le problème de la catégorisation au sens large (par exemple [Rosch 1978 ; Tversky 1977 ; Osherson & Smith 1981 ; Diday 1980]), abordant des concepts centraux

---

[47] Sauf, naturellement, lorsque le critère d'équipollence entre facteurs est par exemple **l'égalité**.

comme ceux de prototype ou de typicalité. Ces travaux s'inscrivent au sein de plusieurs champs : psychologie , sciences cognitives, informatique. Dans cette dernière discipline, des travaux récemment conduits au LIP6 [Rifqi 1996] ont notamment donné lieu à des méthodes et techniques en vue de la génération de prototypes et de la classification à partir de ceux-ci. Cependant, ces méthodes sont conçues pour s'appliquer à des objets flous [Bouchon-Meunier 1993 ; 1995]. Elles ne sont pas applicables ou transposables, du moins dans leur forme actuelle, à des entités séquentielles.

Dans la suite, nous désignons par Gsimil={Sommets,Arêtes} un graphe de similitude tel que nous l'avons défini, produit par la phase d'appariement de facteurs. Soulignons qu’en réalité, la majorité de ce qui sera présenté dans la suite reste valide si l’on enlève la contrainte que le graphe soit non-orienté (symétrie de la fonction de similitude). Etant donné un tel graphe, il y a une multitude de manières possibles d'en extraire des patterns. Des contraintes fixées a priori, comme des contraintes de quorum [cf. par exemple Wang et al. 1994 ; Sagot 1996 ; Schuler, Altschul & Lipman 1991], peuvent être prises en compte lors de cette phase.

Une méthode d'extraction venant naturellement à l'esprit est la recherche de cliques voire de cliques maximales dans le graphe [Babel 1991 ; Balas & Yu 1986 ; Gerhards & Lindenberg 1979 . Soldano, Viari & Champesme 1995]. Rappelons que, étant donné un graphe non-orienté G, une k-clique dans G est en un sous-graphe complet — c’est-à-dire dans lequel il existe une arête entre toute paire de sommets distincts — d'ordre k. Les facteurs représentés par l'ensemble des sommets d'une clique dans un graphe de similitude sont deux à deux équipollents, ce qui constitue une propriété intéressante dans le but de la définition d'un pattern. (Cette propriété est par exemple requise pour les blocs de facteurs recherchés dans un algorithme implémenté dans le programme MACAW [Schuler, Altschul & Lipman 1991], cf. 3.4.1.6.2).

Une approche par recherche de cliques présente néanmoins deux inconvénients. Le premier est qu'elle ne permet pas, en général, d'obtenir directement un représentant. Le second, bien plus ennuyeux, est que dans le cas général la recherche de cliques est un problème NP-complet [Aho, Hopcroft & Ullman 1974 ; Garey & Johnson 1979]…

5.4.2. L'approche "Centre d'Etoile"

Nous avons mis en œuvre une approche plus simple que nous appelons *Centre d'Etoile*, non seulement par analogie avec une approche similaire utilisée en biologie moléculaire ("Center Star" [Gusfield 1993]) mais également en raison du pouvoir évocateur de cette expression : nous définissons une *Etoile* comme étant un sous-graphe de G constitué d'un sommet central et d'un certain nombre de sommets périphériques reliés au centre par des arêtes, lesquelles constituent les *branches* de l'Etoile. (N.B. Cette définition diffère de celle de Gusfield). Une Etoile définit alors un pattern dont le sommet central est le représentant et dont les sommets périphériques sont les différentes occurrences (Cf. Figure 5-6).

Figure 5-6. Illustration de l'approche "Centre d'Etoile"

Proposée par Gusfield en vue d'approximer les scores *sum of the pairs* (SP) dans le cadre de l'alignement multiple [Altschul & Lipman 1989], l'approche Center Star a une signification plus claire que les scores SP d'un point de vue phylogénétique/évolutionnel [Sagot 1996]. Dans notre contexte musical il nous a semblé à la fois naturel, et proche de la philosophie sous-tendant les analyses musicales que nous cherchons à simuler ou automatiser, de définir ainsi un pattern autour d'un représentant dont les différentes occurrences, si elles ne lui sont pas identiques, en sont alors vues comme des *variations*. Les observations que nous avons faites, de la forme prise par les occurrences de différents patterns indiquées dans l'étude d'Owens (cf. 4.5.1, 4.6.2.1), ont confirmé cette intuition. (Voir également la discussion proposée au 5.4.2.5).

5.4.2.1. Algorithme principal

Plaçons-nous dans le cas où le graphe de similitude est valué. Le cheminement algorithmique permettant d'obtenir tous les patterns du "meilleur" au "moins bon" est alors le suivant :

Algorithme 5-1

1. Parcourir l'ensemble des sommets et, pour chaque sommet ν*, calculer :*

*où adj(*ν*) désigne l'ensemble des sommets adjacents à* ν *et, bien sûr, valeur(*ν*,*ν*') désigne la valeur numérique de l'arête (*ν*,*ν*').*

2. Ordonner les sommets par ordre décroissant suivant totalValuation.

Les patterns considérés les meilleurs (tête de liste) sont ceux qui maximisent totalValuation. Pour opérer un choix parmi ces patterns extraits, les possibilités les plus simples consistent :

- soit à ne garder que les k meilleurs, k étant fixé au départ.
- soit à appliquer un seuillage sur totalValuation.

Soulignons que d'autres critères d'ordonnancement des sommets dans la liste peuvent être employés, soit **départageant** les sommets ex-aequo au regard du critère totalValuation, soit **alternatifs** à ce dernier. Des exemples de tels critères, tels que la longueur, le quorum ou le nombre d'occurrences des patterns, seront proposés et illustrés expérimentalement au 5.5.

Si l'on souhaite au contraire conserver tous les patterns extraits, un "post-filtrage" est en général nécessaire pour éliminer la redondance pouvant subsister dans cet ensemble de patterns.

5.4.2.2. Post-filtrage sur les patterns

En pratique, la liste décroissante complète fournie par l'Algorithme 5-1 est potentiellement redondante, au sens où certains des patterns qu'elle contient sont trop similaires entre eux et il convient donc d'effectuer des choix. Cette similarité est ici synonyme de recoupement (des représentants), par suite on fait succéder à cet algorithme, un algorithme de **filtrage**. Ce dernier permet d'éliminer de la liste des patterns extraits ListeExtraits tout pattern dont le représentant présente un recoupement significatif avec le représentant d'un autre pattern meilleur que lui. Il est naturel de reprendre, pour définir un tel recoupement, la même fonction Recoup que celle qui est notamment utilisée pour éviter les recoupements entre facteurs origine. Ce filtrage, qui ne pose pas de difficulté particulière, peut s'effectuer simplement en traitant séquentiellement la liste des sommets/patterns issus de l'Algorithme 5-1 de la façon suivante. Chaque pattern P est comparé à la liste courante ListeMeilleurs(P) des patterns meilleurs que lui, c'est-à-dire ceux qui le précèdent dans ListeExtraits. Si son représentant Repr(P) vérifie Recoup(Repr(P), P') pour un quelconque P'∈ ListeMeilleurs(P), alors P est enlevé de ListeExtraits. En pratique, sur les tailles de données auxquelles nous avons eu affaire, le post-filtrage est effectué de manière quasi-instantanée par Imprology (quelques secondes tout au plus), ce qui permet une utilisation du système en mode interactif. Tout cela sera illustré expérimentalement au 5.5.

5.4.2.3. Cas où le graphe de similitude est non-valué

Dans ce cas, l'Algorithme 5-1 peut être légèrement simplifié pour effectuer la recherche d'Etoiles. Tout se passe comme si les arêtes portaient toutes la valeur 1, les patterns considérés les meilleurs (c'est-à-dire ceux qui maximisent totalValuation) étant alors ceux dont le sommet-représentant dans le graphe a le degré le plus élevé. L'inconvénient par rapport au cas valué tient essentiellement à cette définition plus pauvre de la qualité d'un pattern : deux patterns dont les sommets associés ont le même nombre de voisins sont considérés comme de qualité égale, quelles que soient par ailleurs la force comparée des liens d'équipollence respectifs entre représentant et occurrences. Cela dit il est, en pratique, avantageux de considérer, en plus des noeuds les meilleurs au regard de *totalValuation,* ceux qui ont le plus de voisins — et qui correspondent donc aux patterns ayant le plus grand nombre d'occurrences.

5.4.2.4. Satisfaction de contraintes supplémentaires

Des contraintes supplémentaires, fixées au départ, sur la forme au sens large des patterns, peuvent être prises en compte et intégrées dans l'algorithme *Centre d'Etoile*, typiquement une contrainte de quorum : un tel quorum q étant fixé, il suffit d'intégrer, dans la phase 1 de l'algorithme, le calcul, pour chaque sommet ν, d'un entier NbSeqsDiff dénombrant le nombre de séquences différentes auxquelles appartiennent les différents facteurs adjacents à ν. (Il est très simple, comme nous l'avons fait dans Imprology, de faire en sorte que le calcul d'un numéro de la séquence à laquelle appartient un facteur donné ne nécessite qu'un simple accès à un élément d'un tableau de taille L, longueur cumulée du corpus). Si, au terme du traitement de ν on a

NbSeqDiff<q alors ν est éliminé de la liste des patterns extraits quel que soit par ailleurs la valeur de totalValuation.

5.4.2.5. Qualités

Nous avons évoqué plus haut, dans le domaine musical, la conformité de la philosophie sous-tendant l'approche Centre d'Etoile avec les observations qualitatives pouvant être faites [Owens 1974] ; les résultats présentés en 5.5 apporteront un appui expérimental à cette affirmation. Dans une perspective plus théorique, plusieurs avantages de l'approche peuvent être mis en avant :

- Elle fournit directement un représentant pour chaque pattern. En fait, il est possible qu'un même pattern possède *plusieurs* représentants tout aussi valables. Cela se produit lorsque la somme totalValuation(νi) est la même pour un ensemble de nœuds différents ν1, …, νr définissant la même Etoile en terme de bloc de facteurs (mais avec un centre à chaque fois différent). Notons que les phénomènes de non-unicité de prototype sont des postulats de théories en psychologie cognitive [Rosch 1978].

- Chaque pattern est assorti d'une mesure de sa qualité, ou de sa "cohésion interne" — la valeur de totalValuation — qui est utilisée pour ordonner les patterns extraits du meilleur au moins bon. Cette approche apporte une solution au problème, évoqué par Sagot & Viari (1996), du tri problématique devant être fait entre les patterns, éventuellement très nombreux, extraits par un programme.

- Sa **complexité** est satisfaisante en pratique. En effet, en ce qui concerne la complexité temporelle, deux termes s'ajoutent, liés respectivement à l'étape 1 et l'étape 2 de l'Algorithme 5-1. (Dans la suite nous notons g l'ordre de Gsimil, c'est-à-dire card(Sommets))

    - Pour l'étape 1, dès lors que Gsimil est non-orienté il est clair que chaque valeur intervient exactement deux fois dans le calcul d'une somme. Ainsi, si la complexité temporelle de l'étape 1 est évaluée comme le nombre d'accès aux valeurs, cette complexité temporelle est alors égale à deux fois le nombre total d'arêtes de Gsimil lequel sera ici noté a. Si au contraire on compte le nombre d'additions effectuées, on obtient 2a auquel il faut soustraire le nombre de sommets du graphe soit g. (En effet additionner k nombres requiert (k-1) additions).

- Pour l'étape 2, comme il s'agit d'un tri standard on peut appliquer les algorithmes classiques par exemple le tri par bulles ou quicksort[48], résultant respectivement en une complexité maximale en $O(g^2)$ ou $O(g.\ln(g))$.

Au total, sur l'ensemble des deux étapes, la complexité temporelle est au plus en **$O(g^2)+O(a)$**. En majorant a par $g^2/2$ (graphe non orienté), on obtient un total en **$O(g^2)$**. Mais, ici encore (cf. 5.3.8.2 et notamment le Tableau 5-1) cette majoration surestime très largement les valeurs obtenues expérimentalement. Des exemples de ces dernières sont donnés dans le Tableau 5-2. Les temps d'exécution indiqués ont volontairement été mesurés sur une machine modeste de type ordinateur personnel : Intel Pentium 120 MHz, 32Mo de RAM EDO ; les temps mesurés sur un Intel Pentium II 400MHz avec 64Mo de SDRAM sont inférieurs d'un facteur 4.5 à ceux indiqués, soit par exemple 0.8 s pour L=5000. Les paramètres expérimentaux (en particulier g et a) sont les mêmes que pour le Tableau 5-1. L'algorithme de tri utilisé est un quicksort.

L	**50**	**500**	**5000**
temps en secondes	0.015	0.098	3.74

Tableau 5-2. Mesures expérimentales du temps pris par l'exécution de l'algorithme Centre d'Etoile (étape 1 + étape 2)

La petitesse relative de ces temps de calcul permet ici aussi une utilisation d'Imprology en mode interactif. Considérons à présent la complexité en espace. En plus de l'espace — déjà alloué — requis pour stocker le graphe Gsimil lui-même, l'étape 1 nécessite le stockage de g nombres et, moyennant l'utilisation des algorithmes classiques mentionnés ci-dessus, l'étape 2 ne nécessite aucun stockage supplémentaire. Compte tenu de ce que nous avons dit au 5.3.8.4, il apparaît donc que les besoins supplémentaires en espace spécifiquement rattachés à une phase de catégorisation "Centre d'Etoile" sont, en pratique, négligeables devant les besoins requis par la phase d'appariement de facteurs.

Ajoutons qu'ici aussi les complexités en temps et en espace sont **indépendantes de la taille de l'alphabet.**

---

[48] Voire, le cas échéant, un tri par compartiments (*bin sorting*), de complexité O(g) [Aho, Hopcroft & Ullman 1983].

5.4.3. Gestion des recoupements entre facteurs-extrémités d'arêtes

Ces phénomènes de recoupement, évoqués plus haut et illustrés par la Figure 5-7, se produisent fréquemment lors de la phase d'appariement de facteurs et ne sont généralement pas souhaitables (redondance). Si nous traitons cette question ici plutôt qu'au 5.3, c'est que si les recoupements peuvent dans certains cas être évités "à la source" et en-ligne au sein de FlExPat-AF, en revanche dans le cas général ils font l'objet d'un traitement en différé lors de la phase de catégorisation. Ce choix (en-ligne vs. différé) dépend fortement de l'approche employée pour la catégorisation.

Figure 5-7. Recoupement de longueur 5 entre deux facteurs-extrémités d'arêtes de longueurs respectives 9 et 11.

La fréquence de ces recoupements est principalement due au fait que lorsque deux facteurs F et F' sont jugés équipollents, il existe un certain nombre d'autres couples $(F,F'_k)_{k \in \mathbf{N}n}$ eux-aussi équipollents, dont le nombre k est d'autant plus grand que F et F' sont similaires. Par exemple, si m est la longueur de F', le préfixe de F', et son suffixe, de longueurs m-1, ont de bonnes chances d'être équipollents à F.

En effet, pour ce qui concerne par exemple ce préfixe noté PréF', la valeur de l'alignement optimal entre F et PréF' est au moins égale à celle de l'alignement optimal entre F et F' — soit Simil(F,F') — moins la valeur absolue de la contribution de l'insertion de F'[i]. Si Eq est un seuillage sur la similitude (3.2.3), il suffit que cette valeur absolue soit inférieure à (Simil(F,F')- *seuil*) pour que F et PréF' soient équipollents.

Et, au-delà, il peut en être de même pour le suffixe, le préfixe et le facteur central de longueurs m-2, etc.

Pour éviter des recoupements de ce genre qui soient trop étendus, on peut par exemple ne créer qu'un seul lien, F—F' ou F—F'' (cf. Figure 5-7) dès lors que l'étendue du recoupement entre F' et F'' dépasse un certain seuil. Ici encore, il nous faut préciser comment l'étendue d'un recoupement est mesurée et évaluée (voir ci-dessous). C'est cette approche que nous mettons en œuvre dans FlExPat, avec les modalités précises qui suivent :

5.4.3.1. Mesure de l'étendue ("taille") d'un recoupement

Il est naturel de définir l'étendue d'un recoupement à partir de la même fonction Recoup utilisée pour le filtrage des recoupements entre facteur origine et facteur destination (cf. 5.3.3.1). Par exemple, en prenant :

,

et ρ=0.25, comme les facteurs F' et F" de la Figure 5-7 sont de longueurs respectives 9 et 11 et que la longueur du facteur intersection est de 5, l'étendue du recoupement est égale à 5/9. Par conséquent Recoup(F,F') vaut VRAI.

5.4.3.2. Critère de choix entre deux facteurs candidats

Une fois un recoupement significatif mis en évidence entre deux facteurs F' et F" tous deux équipollents à un facteur f, il faut un critère, ou une combinaison de critères, pour choisir lequel des deux liens possibles d'équipollence, F←→F' ou F←→F", va être créé. Au moins deux critères peuvent être raisonnablement proposés :

1. Les valeurs de similitudes respectives

2. Les longueurs respectives de F' et de F"

En pratique, dans Imprology nous utilisons généralement une combinaison de ces deux critères qui s'avère pertinente en pratique : le critère 1 est le critère principal (choisir le facteur permettant la plus forte similitude), le critère 2 servant à départager lorsque c'est nécessaire (choisir le facteur le plus long en cas d'égalité des similitudes).

5.4.3.3. Intégration dans FlExPat

Comme nous l'avons évoqué, cette gestion des recoupements entre facteurs-extrémités d'arêtes peut être, soit intégrée à FlExPat-AF, soit effectué au début ou pendant la phase de catégorisation. Dans le cas de son intégration en-ligne, réduisant l'ordre de Gsimil elle permet une limitation de la complexité en espace.

5.5. Implémentation dans Imprology et résultats expérimentaux

Dans la présentation de l'implémentation que nous avons faite de l'algorithme FlExPat au sein du système Imprology, nous nous concentrerons essentiellement sur les aspects relatifs au calcul et à la visualisation d'alignements, le reste découlant naturellement de la description de FlExPat proposée dans les sections qui précèdent.

5.5.1. Calcul et visualisation d'alignements

5.5.1.1. Principe

Nous avons présenté plus haut différents algorithmes relatifs à la comparaison globale, loco-globale et locale. Ces trois modes de comparaison se traduisent respectivement par une recherche d'alignements optimaux, respectivement globaux, loco-globaux et locaux. Nous avons vu que les algorithmes basés sur la programmation dynamique font intervenir une matrice M, explicite ou implicite, dont les dimensions sont fixées par les longueurs respectives des séquences comparées. Cette matrice est remplie progressivement, la détermination de chaque élément Cij résultant d'une maximisation (cf. par exemple Equation 3-3, page 102) ou d'une minimisation, et on y extrait, au final, la valeur optimale d'alignement. Pour simplifier l'exposé, et sans perte de généralité, nous supposerons dans la suite de la présente section qu'il s'agit de **maximisation**s.

Dans certains cas il est suffisant de connaître, non pas l'alignement optimal (ou les alignements optimaux, s'il y en a plusieurs), mais simplement sa valeur. Dans d'autres cas, notamment lorsqu'on souhaite connaître les facteurs (origine et destination) correspondant à cet/ces alignement(s),cette valeur ne suffit pas. Pour la détermination d'alignement souhaitée, une méthode consiste à faire usage d'une matrice annexe P calculée en même temps que M et dont un élément Pij (pointeur) garde une trace de la ou des ligne(s) ayant réalisé la maximisation lors du calcul de l'élément correspondant de M, soit Mij. Certains éléments de P restent vides : dans le cas d'une comparaison globale, c'est l'élément « supérieur gauche » (0,0). Dans le cas local, c'est l'ensemble des éléments correspondant au cas où la valeur issue de la maximisation dans l'Equation 4-1 (page 134) est 0, soit la valeur minimum possible.

La détermination de chaque alignement optimal se fait alors en **partant d'un élement déterminé** (voir plus bas) de la matrice M puis en **remontant** (*backtrack*) au fil des pointeurs jusqu'à ce qu'une certaine **condition d'arrêt** soit vérifiée. La condition d'arrêt est le fait d'avoir atteint un élément vide (sans pointeur). Les éléments initiaux, quant à eux, dépendent du type de comparaison que l'on effectue : pour une comparaison locale, c’est ceux ayant la valeur la plus élevée dans M. Par contraste, par exemple pour une comparaison globale, l’élément de départ (unique) est l’élément « inférieur droit » de la matrice. Chaque « chemin » ainsi obtenu dans la matrice détermine de manière unique un alignement (voir exemple ci-dessous).

5.5.1.2. Exemple

La Figure 5-8 est une illustration de la matrice P obtenue dans le cadre de la comparaison locale des deux passages de l’exemple du 4.5.2.3.A titre d’information, nous faisons figurer en superposition la matrice M « enrichie » qui avait été montrée à la Figure 4-10.

Les pointeurs sont matérialisés par des flèches de trois types (gauche, haut, oblique) correspondant à chacune des lignes de la maximisation intervenant dans l’Equation 4-1 (page 134).

Figure 5-8. Visualisation des pointeurs permettant la détermination de l’alignement optimal

Le chemin donnant l’alignement (optimal) est obtenu en remontant à partir de la valeur maximale de la matrice M, soit 6.3. Nous avons mis en évidence ce chemin sur la figure en mettant en double trait les 4 flèches qui le définissent. L’alignement correspondant est obtenu en concaténant, dans l’ordre, les appariements correspondant à chacune de ces flèches. Comme ce sont toutes des flèches **obliques**, cela conduit à 4 appariements de type Remplacement, visualisés ci-dessous (Une flèche 'haut' correspondrait à une délétion, une flèche 'gauche' à une insertion).

Notons que dans cet exemple l'alignement optimal est unique, mais, comme nous l'avons vu, ce n'est pas toujours le cas. Remarquons par ailleurs que, dans cet alignement local, les facteurs optimaux trouvés sont de taille quasiment égale à celle des séquences comparées. Il est fréquent, dans des contexte en « vraie grandeur », de trouver au contraire des facteurs optimaux significativement plus courts que les séquences comparées, ce qui traduit un caractère proprement local de l'alignement.

5.5.1.3. Multiplicité des alignements optimaux

Voyons maintenant sous quelles conditions il y a, non pas un, mais **plusieurs** alignements optimaux (ayant tous la même valeur, maximale). Deux facteurs, pouvant se combiner entre eux, peuvent être à l'origine de cela :

- Lors du backtrack dans la matrice P, lorsqu'un élément est rencontré qui contient plus d'un pointeur, cela donne lieu à deux chemins différents dans la matrice — plus exactement deux **groupes** de chemins différents, puisque d'autres éléments situés sur ces deux chemins peuvent contenir eux aussi plusieurs pointeurs, etc. Cela conduit ainsi à une combinatoire sur les chemins, et par suite sur les alignements optimaux.

- Dans le cas local ou loco-global, il peut y avoir **plusieurs** éléments maximaux dans M. Cela conduit, là aussi, à autant de groupes de chemins, et donc de groupes d'alignements optimaux, distincts.

5.5.1.4. Implémentation

Imprology permet à la fois de calculer et visualiser des alignements **globaux**, issus notamment de l'exécution de FlExPat, mais aussi des alignements **locaux** ou encore **loco-globaux** (voir plus haut). Le calcul des alignements optimaux met en oeuvre les techniques que nous avons présentées au 5.5.1.

La visualisation d'un alignement se fait usuellement au travers de la visualisation de sa **trace :** celle-ci, que nous avons rencontré maintes fois dans ce qui précède, est proposée à la Figure 5-9 pour l'alignement optimal trouvé dans l'exemple ci-dessus (5.5.1.2). On peut observer que cet alignement rend compte du contour ascendant commun de hauteurs (arpèges).

Figure 5-9. Visualisation d'alignement local

5.5.2. Résultats expérimentaux

5.5.2.1. Introduction

Il serait exclu de présenter l'ensemble des expérimentations que nous avons effectuées avec Imprology / FlExPat sur les séquences du corpus de transcriptions d'Owens ou sur d'autres séquences musicales. La présentation de résultats issus du système n'a de sens que si les conditions expérimentales sont précisées. Or, comme cela va apparaître dans la suite, exposer l'ensemble des valeurs des paramètres (entendus au sens large) requiert un espace relativement important : spécification du corpus étudié, configuration de FlExPat (ensemble des types d'appariements autorisés, m_{min}, m_{max}, fonction d'équipollence, fontion Δm, fonction Recoup, etc.), jeu de description employé, pondérations et fonctions de contribution associées à chaque description. Il est clair qu'une accumulation de tels ensembles d'informations n'est pas souhaitable dans le cadre du présent mémoire, aussi nous concentrerons-nous sur un sous-ensemble de nos expérimentations, que nous aurons ainsi la possibilité de décrire plus en détail. Dans un premier temps, nous considérons le sous-corpus C_a constitué des solos de Charlie Parker sur *Cool Blues Take 1* et *Cool Blues Take 2* (enregistrés le 19 février 1947), corpus dont la longueur cumulée est de 500 environ. L'un des avantages de ce choix est de nous permettre de proposer au lecteur (Annexe B) la transcription complète des deux solos. (Là encore, il serait hors de question de reproduire au sein de ce mémoire davantage de partitions (séquences sources), tant pour des raisons de propriété scientifique / littéraire / artistique que pour des considérations relatives à la taille du mémoire…). Nous emploierons successivement, entre autres, différents jeux de descriptions et différentes configurations pour FlExPat (modèle de comparaison, fonctions d'équipollence, pondérations, etc.). Dans un second temps, nous nous placerons sur le corpus C_b, sur-ensemble de C_a constitué de 10 morceaux du groupe ‘do majeur Blues’, dont la longueur cumulée est de 2000 environ. A nouveau, nous mettrons en œuvre différents ensembles de conditions expérimentales. Nous aurons l’opportunité de confronter les résultats obtenus avec C_a et C_b, en particulier au niveau des patterns du lexique d’Owens qui sont extraits par Imprology/FlExPat. Enfin, nous présenterons les tendances majeures observées dans les résultats, en nous basant cette fois-ci sur l’ensembles des expérimentations que nous avons effectuées, y compris toutes celles qui ne sont pas décrites dans ce mémoire.

5.5.2.2. Cool Blues Take 1 et Take 2

Le corpus C_a est constitué des solos Cool Blues Take 1 (séquence source S_a1) et Cool Blues Take 2 (séquence source S_a2). Cool Blues (également appelé « Hot Blues » ou « Blow Top Blues ») est le blues préféré de Parker en do majeur [Owens 1974, p. I-157]. Les deux prises ont été enregistrées le 19/2/47 (Dial LP 202). Leurs transcriptions sont proposées à l’Annexe B. Les caractéristiques quantitatives générales des deux morceaux ont été données au Tableau 4-3, page 144.

5.5.2.2.1. Configuration

Les paramètres d’exécution pour la phase d’appariement de facteurs (FlExPat-AF) sont :

- $ETAA=ETAA_{standard}$
- $m_{min}=3$, $m_{max}=27$.
- Fonction d'équipollence : définie par un seuillage à la valeur 0.7 sur le ratioIdentité. Autrement dit :

Eq(F,F') ⇔ l'alignement optimal entre F et F' a un ratioIdentité supérieur ou égal à 0.7.

- Fonction Δm : linéaire, définie par , de sorte que, par exemple, $\Delta m(10)=3$ mais $\Delta m(9)=2$.
- Fonction Recoup : définie par un seuillage (cf. 5.3.3.1) avec $\rho = 1/3.001$.

5.5.2.2.2. Jeu de description employé

- Description globale : **signature métrique** [de chaque morceau].
- Description locale : **extremum local de hauteur**

- Descriptions individuelles :
  - q Description intrinsèque (de base) : **durée** (exprimée en temps relatif)
  - q Descriptions relationnelles (dérivées) :
    - ♦ **Intervalles chromatiques** formés avec la note précédente et avec la note suivante. (Une telle prise en compte simultanée, pour chaque note, des deux intervalles « arrière » et « avant », permet d'éviter les problèmes qui sont rencontrés systématiquement lorsqu'on ne considère que l'un des deux intervalles)
    - ♦ **Niveaux métriques** binaire et ternaire (cf. 4.7.7.3.2)
    - ♦ **Position dans le pic d'extremum local** (cf. 4.7.7.2.2)
  - q Description « particulière » : **silences** (cf. 4.7.7.3.3).

5.5.2.2.3. Fonctions de contribution

Nous détaillons les fonctions de contribution rattachées aux différentes descriptions (cf. 4.7.4), pour celui des types d'appariements autorisés qui en fait intervenir le plus grand nombre, en l'occurrence le type Remplacement :

- 'Durée' :

  Fonction linéaire, d'une forme que nous utilisons souvent :

  ,

  avec BiaisAdditif=1 et Facteur=4 (valeurs par défaut), les durées étant exprimées en noires.

- 'IntervallesChromatiques' :

Le calcul des valeurs de cette fonction de contribution spécialisée fait intervenir toute une arborescence de cas, permettant de gérer toutes les « exceptions » inhérentes à la notion de description variationnelle, et que les systèmes antérieurs éludent généralement : l'intervalle « arrière » ne peut être calculé pour la première note d'une mélodie, ou pour une note précédée d'un « long » silence. Il en est bien sûr de même pour le calcul de l'intervalle « avant » pour la dernière note d'une mélodie, ou pour une note suivie d’un tel silence. En revanche, lorsque la note est précédée (resp. suivie) d’un « court » silence, il s’avère pertinent de calculer la valeur de l’intervalle arrière (resp. avant) comme la taille algébrique de l’intervalle formé avec la note antépénultième (resp. la note suivante de la suivante (silence)). En pratique, définir un **court** silence comme étant strictement plus court que la note considérée, est une solution satisfaisante. La Figure 5-10 illustre l’arbre des cas. Le calcul de la contribution d’un remplacement fait donc intervenir une combinatoire non négligeable, résultant de la « multiplication » des arborescences respectives de Su[i] et de Sv[i’].

La fonction de contribution que nous définissons sur ces bases est une fonction fortement discrète standard (cf. 4.7.4.3) prenant un ensemble de 5 valeurs {Min, Bas, Moyen, Haut, Max}, en fonction des positions respectives dans l’arbre des cas. La valeur Max est atteinte lorsque non seulement les tailles algébriques des intevalles précédant Su[i] et précédant Sv[i’] sont égales, et il y également égalité sur les intervalles **suivant** chacune des deux notes. La valeur Min est atteinte lorsque l’une des deux notes (et elle seulement) est un silence. Nous prenons ici Min=-4, Bas=-2, Moyen=0, Haut=2, Max=4 (valeurs par défaut).

Figure 5-10. Arborescence de cas intervenant dans le calcul de la fonction $\text{contrib}_{\text{Remplacement,IntervallesChromatiques}}$

- ‘NiveauxMétriques’ :

Comme pour la description ‘Durée’, nous employons une fonction linéaire de forme standard :

avec (voir plus haut) :

MinNM(Su[i], Sv[i’]) = Min(|Su[i].NMB - Sv[i’].NMB| , | Su[i].NMT - Sv[i’].NMT|).

Ici encore, pour BiaisAdditif et Facteur nous adoptons les valeurs par défaut respectives 1 et 4.

- ‘PositionPicExtremumLocal’ :

  Comme pour la description ‘IntervallesChromatiques’, il existe ici pour chacune des deux notes appariées une arborescence de cas : note appartenant ou n’appartenant pas à un pic d’extremum local ; si oui, maximum vs. minimum local, et note correspondant à l’extremum vs. note située sur les « bords » du pic. Nous employons ici aussi une fonction fortement discrète standard (cf. 4.7.4.3) prenant un ensemble de 5 valeurs {Min, Bas, Moyen, Haut, Max}, la valeur Max étant atteinte lorsque les deux notes appariées sont deux maxima locaux ou deux minima locaux et la valeur Min, lorsque l’une est un minimum, l’autre un maximum. Pour Min, Bas, Moyen, Haut, et Max nous adoptons à nouveau les valeurs par défaut, soit respectivement -4, -2, 0, 2 et 4.

- NoteVsSilence’ :

  Une fonction pseudo-booléenne (cf. 4.7.4.3) est naturellement utilisée, valant IdConst lorsque les deux notes comparées sont deux silences ou deux notes à proprement parler, et DiffConst dans le cas contraire. Les valeurs des deux constantes sont celles par défaut : IdConst=1, DiffConst=-1.

5.5.2.2.4. Pondération

Les poids globaux respectifs attribués aux différents types d’appariements autorisés (cf. 4.7.3) sont les suivants :

- Insertion : 2.5
- Délétion : 2.5
- Remplacement : 1

5.5.2.2.5. Pondération des descriptions individuelles

L’indication des pondérations respectives est accompagnée, comme cela est nécessaire, des contributions moyennes correspondantes. Le calcul des contributions moyennes au plan global donne les valeurs suivantes :

- Pour le type d'appariement Remplacement : -4.41
- Pour le type d'appariement Insertion : -6.60
- Pour le type d'appariement Délétion : -6.60

Voyons à présent les pondérations et la répartition de la contribution moyenne sur les différentes descriptions, pour le type d'appariement Remplacement (les valeurs des contributions moyennes indiquées tiennent déjà compte de la pondération) :

- 'Durée' :
  - Pondération : 1
  - Contribution moyenne : -0.68
- 'IntervallesChromatiques' :
  - Pondération : 1
  - Contribution moyenne : -2.43
- 'NiveauxMétriques' :
  - Pondération : 0.5
  - Contribution moyenne : -1.95
- 'PositionPicExtremumLocal' :
  - 'Pondération : 1
  - Contribution moyenne : -1.56
- NoteVsSilence' :
  - Pondération : 3
  - Contribution moyenne : +2.21

5.5.2.2.6. Résultats obtenus

Caractéristiques numériques générales

(Toutes les informations qui suivent peuvent être obtenues directement et instantanément par l'utilisateur d'Imprology).

- La longueur cumulée du corpus après enrichissement de la représentation (ajout des silences) est de **454**, soit 221 notes ou silences pour S_a1 et 233 pour S_a2.
- Le graphe de similitude comporte (avant tout filtrage) **1995 sommets** et **3829 arêtes**. Ce premier nombre est à comparer au nombre de facteurs traités par FlExPat-AF, c'est à dire le nombre total de facteurs de C_a dont la longueur est comprise entre m_{min} et m_{max}. Une majoration raisonnable de ce nombre est 11003, nombre obtenu en faisant abstraction des facteurs 'mixtes' (à cheval sur deux séquences), qui ne devraient pas être inclus dans le dénombrement. Par ailleurs, à titre d'indication le nombre de paires de facteurs possibles {(i,m),(i',m')} de C_a, avec i'<i et (m,m')$\in \{m_{min},\ldots,m_{max}\}^2$, est de l'ordre de 60 000 000. Ce nombre est un majorant théorique du nombre d'arêtes maximal du graphe de similitude — majoration très large en raison notamment de l'intervention de la fonction Δm qui limite considérablement le nombre d'arêtes pouvant être créées.
- le nombre maximum de sommets adjacents à un sommet est égal à **9** ; en d'autres termes la plus grande Etoile, au sens du 5.4.2, est de taille 10.
- Le quorum maximal, soit 2, est atteint.
- Le plus long pattern, au sens de la stratégie de catégorisation Centre d'Etoile et sans contrainte de quorum, est de longueur 27. Cet unique pattern de longueur maximale sera explicité ci-après.

A présent, nous allons présenter les résultats de la phase de catégorisation dans le cadre de la stratégie Centre d'Etoile, et cela en employant différents critères d'ordonnancement de la liste des Etoiles (cf. 5.4.2.1) et différents filtrages de cette liste (cf. 5.4.2.2).

Ordonnancement basé (exclusivement) sur *totalValuation*

Dans un premier temps, considérons la liste obtenue en rangeant les patterns (Etoiles) suivant l'ordre décroissant de leur totalValuation, lequel vaut (rappel) :

Le meilleur[49] pattern Pat_a1 obtenu — totalValuation=**454** — a pour représentant le passage de S_a2 (Cool Blues Take 2) défini par le laps[50] {"10:2"—10—"12:4"}, la notation étant celle introduite au 4.6.2.2. Ce passage est signalé sur la Figure 5-11 par un trait noir horizontal au dessus de la portée. La valeur **'idSim'=189** indiquée au-dessus du trait est l'*autosimilitude* du passage, c'est à dire la valeur de son alignement à l'identique (cf. 3.2.4.2.7). Cette autosimilitude (qui intervient dans le calcul du ratioIdentité de tout alignement ayant ce passage comme séquence origine ou comme séquence destination) servira ultérieurement comme référence dans le cadre du jugement de la qualité de chacune des occurrences du pattern.

Figure 5-11. Représentant du pattern Pat_a1 : Passage {"10:2”--10--”12:4”} du solo *Cool Blues Take 2*

Pat_a1 a un quorum égal à 2 : 2 occurrences dans S_a2 et 1 occurrence dans S_a1. Celles-ci sont montrées dans les trois figures qui suivent, par ordre décroissant de similitude (165, 149 et 140, respectivement).

Figure 5-12. Première occurrence du pattern Pat_a1 : Passage {“35:1.5"--(21/2)--"37:4"} du solo *Cool Blues Take 2*

---

[49] Voir « remarque concernant la valuation du graphe de similitude » un peu plus bas.

[50] A titre d'information, Pat_a1 a pour coordonnées (275,18) dans le corpus (C_a).

Figure 5-13. Seconde occurrence du pattern Pat_a1 : Passage {"22:2"--10--"24:4"} du solo *Cool Blues Take 2*

Figure 5-14. Troisième occurrence du pattern Pat_a1 : Passage {"35:3"--8--"37:3"} du solo *Cool Blues Take 1*

La seconde partie du pattern (précisément, le passage {"10:4.5"--6--"12:2.5"} dans *Cool Blues Take 2*) apparaît de façon exacte, du moins au regard du jeu de descriptions employé, dans chacune des 3 occurrences. (Cette partie est parfaitement « conservée », dirait le biologiste). La première partie, un contour de hauteur descendant en croches[51], est bien prise en compte dans les occurrences en dépit de variations diverses — traduites par des indels et/ou des substitutions. Anticipant un peu sur la suite, indiquons que la première partie, « moins conservée », du pattern, dont il est manifestement souhaitable qu'elle soit intégrée à celui-ci, **échappe complètement** à un algorithme d'extraction exigeant des répétitions exactes, même si celles-ci sont définies sur le même jeu de descriptions que celui employé ici. Nous avons pu le constater en faisant des expérimentations où nous avons configuré FlExPat-AF de la manière qui sera décrite au 5.6.2.3. Ainsi, les trois exemples d'équipollence montrés, tout comme les divers autres exemples qui seront montrés dans la suite, mettent déjà en évidence, qualitativement et quantitativement, la qualité des résultats atteints par le MEVM[52] en matière de comparaison de passages mélodiques.

Le pattern extrait est parfaitement conforme à ce qui se dégage d'une audition des deux solos : la répétition (approximative) du passage au travers des deux solos — et en

---

[51] Éventuellement contrarié localement, comme dans la 2e occurrence.

[52] Au travers de l'instance particulière du modèle mise en oeuvre, caractérisée par l'ensemble des types d'appariements autorisés, le jeu de descriptions, leurs pondérations respectives, et les différentes fonctions de contributions employés.

particulier dans Cool Blues Take 2 — est entendue de manière très sensible. Owens n'avait, soit pas trouvé, soit pas signalé, ce pattern Pat_a1 (quoique une variante du pattern 3Aa du lexique — cf Annexe A — y ressemble quelque peu, et que Pat_a1 « contienne », à une transposition chromatique près, le pattern de quatre notes 5B — voir plus bas). L'une des raisons en est, peut-être, que ce pattern possède un caractère local, c'est à dire n'apparaît pas dans l'ensemble du corpus étudié mais uniquement dans un groupe de solos de tonalité do majeur et de grille d'accords Blues. Cela met en lumière l'un des intérêts clés d'un système d'analyse automatique comme Imprology, permettant d'extraire de tels patterns 'locaux' (du point de vue du corpus), spécifiques à un sous-corpus arbitraire, et cela d'une façon interactive. Ajoutons que Pat_a1 exhibe une caractéristique fort intéressante d'un point de vue harmonique : Considérons les régions respectives dans lesquelles apparaîssent le représentant et ses deux occurrences dans *Cool Blues Take 2*. Ces régions sont centrées sur les mesures 11, 23 et 36, autrement dit elles se succèdent avec une séparation d'environ 12 mesures. Or c'est là précisément la longueur de la grille d'accords (Blues sur 12 mesures). Cela signifie en particulier que le pattern et ses occurrences apparaîssent sur une région correspondant la même cellule harmonique, en l'occurrence un II-V-I, faisant ainsi echo à ce que nous avions dit précédemment au sujet de l'association patterns-cellules harmoniques (cf. 4.7.5.2). Ce phénomène sera à nouveau rencontré (mais non signalé) à diverses reprises dans la suite.

Remarque concernant la valuation du graphe de similitude. Nous employons ici une valuation des arêtes du graphe issu de FlExPat-AF par les **similitudes**, c'est à dire les valeurs d'alignements optimaux entre paires de facteurs. Cela dit, il est également possible dans Imprology, via un simple réglage de paramètre, de basculer (interactivement[53]) vers une valuation par les **ratioIdentités** de ces alignements. Moyennant un tel choix, en particulier, le meilleur pattern trouvé n'est plus Pat_a1, mais le pattern de longueur 3 « 'Cool_Blues_Take1' {'33:3.5'—(3/2)—'34:1'} », qui se trouve maximiser la taille d'étoile, soit 10, sur l'ensemble des 1995 patterns. Schématiquement, la valuation par les similitudes tend à favoriser l'extraction de longs patterns tandis que la valuation par les ratiosIdentités tend à favoriser l'extraction de patterns ayant de nombreuses occurrences (« grandes » étoiles). Ces deux tendances ont chacune leur pertinence, et pouvent donc avantageusement être employées en complément l'une de l'autre.

---

[53] Dans Imprology, le temps de passage des similitudes aux ratiosIdentités — ou l'inverse — est, en pratique, quasiment instantané.

Filtrage

Il aurait été « quelque peu » fastidieux, et surtout très redondant, d'examiner individuellement les 1994 autres Etoiles, dont une proportion importante — 940 d'entre elles — ne possèdent qu'une seule branche (traduisant simplement des liens d'équipollence entre facteurs).

A la place de cela, nous effectuons en premier lieu, sur la liste des patterns (Etoiles) triée par ordre décroissant de totalValuation, le filtrage décrit au 5.4.2.2. La Figure 5-15 montre le résultat de ce filtrage au travers du profil de totalValuation obtenu : à chaque pattern correspond un cylindre vertical dont la longueur est proportionnelle au totalValuation du pattern. On retrouve naturellement en première position le pattern Pat_a1 (totalValuation=454). La première observation pouvant être faite est que **le nombre de patterns a été très fortement réduit**, étant ramené de 1995 à 22 soit presque un facteur 100. L'examen individuel de chacun des patterns (et de son ensemble des occurrences, défini par l'Etoile qui lui correspond), qui était totalement exclu avant filtrage, est maintenant envisageable. La seconde observation est que **le profil distingue très nettement le groupe des 4 meilleurs patterns**. A l'examen, les 4 représentants correspondent approximativement, dans cet ordre, au représentant de Pat_a1, puis à sa première, sa troisième et sa seconde occurrences. (L'ordre des occurrences est celui des trois figures proposées plus haut, c'est à dire celui des similitudes décroissantes avec le représentant).

Figure 5-15. Profil obtenu après filtrage (élimination des recoupements entre représentants de patterns) à partir de la liste des patterns triée selon les "totalValuation" décroissants

Ainsi, d’un côté le résultat du filtrage est tout à fait satisfaisant au sens où il fait ressortir de manière très nette les passages récurrents les plus saillants à l’oreille ; d’un autre côté, il est possible, d’un certain point de vue, de voir la présence des second, troisième et quatrième patterns dans le profil comme une redondance résiduelle. Celle-ci peut être éliminée au moyen d’un nouveau filtrage. Cet algorithme de « post-post-filtrage » est très simple (et, en pratique, instantané sur les tailles de données auxquelles nous avons affaire), analogue à celui décrit au 5.4.2.2. La différence réside dans le fait que, dans l’algorithme initial, on évalue la taille des recoupements entre le représentant du pattern P en cours de traitement et les représentants des patterns meilleurs que lui (c’est à dire situés avant lui dans la liste courante). Dans ce second algorithme au contraire, on évalue la taille des recoupements entre les **occurrences** du pattern et les représentants des patterns meilleurs que lui. Ici, l’execution de cet algorithme permet de diviser encore la taille de la liste des patterns extraits par 3 environ, et fait, entre autres, ressortir de manière encore plus contrastée le pattern Pat_a1.

Comme cela a été évoqué plus haut, l’ordonnancement de la liste des patterns (Etoiles) au moment du filtrage n’est pas nécessairement basé (uniquement) sur totalValuation. Imprology offre la possibilité d’ordonner toute liste de patterns selon différents critères ou combinaisons de critères ; en pratique, ces tris sont généralement instantanés (moins d’une seconde), ce qui offre à l’utilisateur la possibilité d’un travail en mode interactif. La combinaison de critères que nous allons ici considérer est la suivante :

- longueur
- totalValuation
- taille d’Etoile (c’est à dire nombre de sommets adjacents + 1)
- quorum

Cela signifie que les patterns sont ordonnés du plus long au plus court, que chaque sous-liste de patterns de même longueur est elle-même ordonnée selon les totalValuation décroissants, etc. Voyons à présent le résultat du filtrage décrit au 5.4.2.2 sur la liste triée ainsi obtenue. La liste filtrée contient cette fois-ci 30 patterns (nombre 67 fois moindre que les 1995 initiaux et pouvant, là aussi, être encore réduit par une étape de filtrage supplémentaire). Le premier de la liste est l’unique pattern Pat_a2 de longueur 27 montré à la Figure 5-16.

N.B. : à partir de la Figure 5-16 (incluse), les passages sont repérés au moyen de la **barre horizontale sombre** située au-dessus de la portée. Les graduations indiquent les numéros de mesure et les temps.

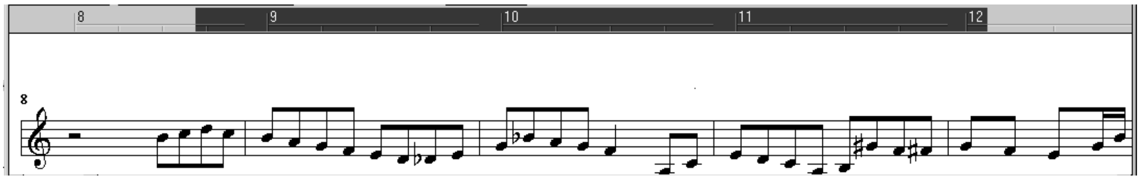


Figure 5-16. Représentant du pattern Pat_a2 : passage {'8:4'--(27/2)--'12:1.5'} du solo *Cool Blues Take 1*

Pat_a2 correspond à une Etoile de taille 2, l'occurrence étant le passage montré à la Figure 5-17. Il s'agit d'une occurrence interne (même séquence S_a1), avec une similitude au représentant égale à 185.8. La (forte) similitude détectée par le MEVM entre les deux passages est indéniable, visuellement comme auditivement, quoique n'étant pas une égalité. L'occurrence est notamment caractérisée par un décalage métrique minimal (blanche[54]), constant tout au long de l'occurrence, ce qui contribue à justifier la stratégie de prise en compte des aspects métriques que nous avons adoptée au sein du MEVM. Des exemples d'intervention particulièrement pertinentes de multidélétion (celle du mi et du sol à cheval sur les mesures 9 et 10) et de substitution généralisée (le passage la-do en fin de mesure 10 substitué par le passage, de même durée totale, sol#-la-do de la mesure 35), sont également fournis.

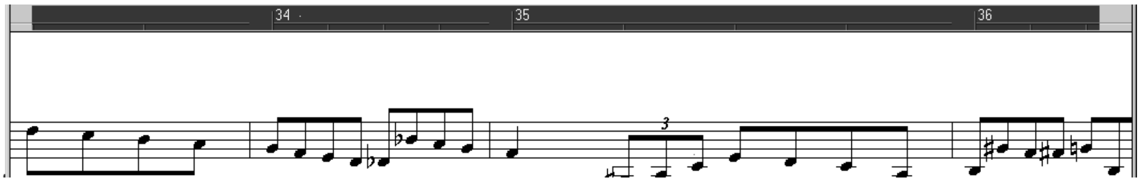


Figure 5-17. Occurrence du pattern Pat_a2 : passage {"33:3"--(25/2)--"36:3.5"} du solo *Cool Blues Take 1*

Avant de confronter ces résultats avec ceux qui avaient été obtenus avec un filtrage sur une liste de patterns triée différemment, examinons le second pattern de la liste, soit Pat_a3. Son représentant est le passage {"35:1.5"--14--"38:3.5"} du solo *Cool Blues Take 2* (Figure 5-18). Il est suivi, dans la liste, d'un autre pattern de même

---

[54] Il s'agit bien du décalage minimum « perceptible », un décalage d'une (ou plusieurs) mesure(s) passant pour sa part « inaperçu », en vertu de l'invariance de le description 'NiveauxMétriques' à un tel décalage.

longueur, mais le second critère de tri (totalValuation) les départage très nettement : 347.5 contre 185.8.

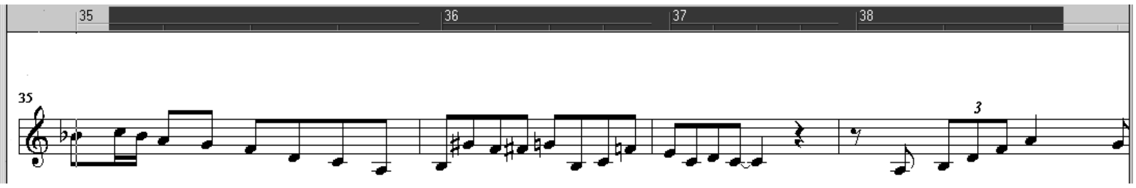



Figure 5-18. Représentant du pattern Pat_a3 : passage {"35:1.5"--14--"38:3.5"} du solo *Cool Blues Take 2*

Les deux occurrences, de similitudes respectives174 et 173.5, sont présentées sur les deux figures suivantes. Notons que, de manière parfaitement adéquate, la montée en pseudo-arpèges (triolet de croches) en fin de pattern — qui est d'ailleurs le **pattern 1Ad d'Owens** (cf. Annexe A) — est bien prise en compte, en dépit notamment de la durée différente du silence dans la 1e occurrence[55]. Cela est dû notamment aux caractéristiques métriques de cette montée, caractéristiques qui sont identiques pour le représentant et les 2 occurrences. Ces exemples montrent que des structures locales telles que les arpèges peuvent dans certains cas être prises en compte, même sans introduire explicitement les **descriptions locales** correpondantes (cf 4.7.7.2). Dans le cas présent, c'est la description IntervallesChromatiques qui a induit cette prise en compte.

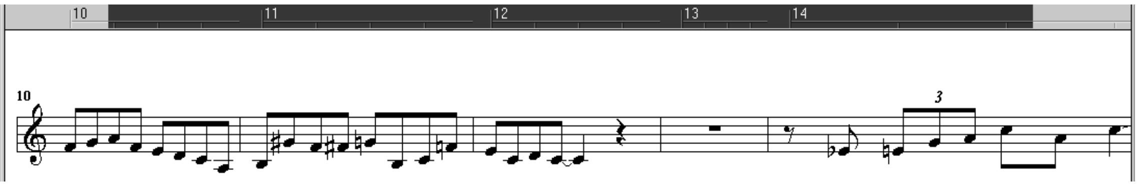


Figure 5-19. Occurrence du pattern Pat_a3 : passage {"10:2"--(35/2)--"14:3.5"} du solo *Cool Blues Take 2*

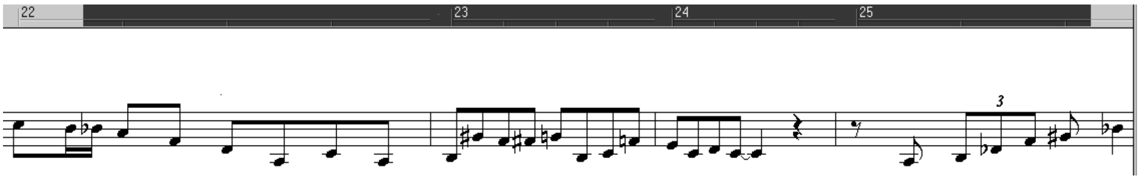



Figure 5-20. Occurrence du pattern Pat_a3 : passage {"22:1.75"--(53/4)--"25:3"} du solo *Cool Blues Take 2*

---

[55] On peut imaginer qu'un musicologue utilisateur ne considère pas souhaitable cette tolérance relative à la longueur du silence. Une simple augmentation de la pondération de la description 'Durée' suffit alors à satisfaire un tel souhait.

Confrontation des deux approches de filtrage

Des liens évidents existent entre, d'un côté, les quatre passages montrés de la Figure 5-17 à la Figure 5-20 et, de l'autre, les quatre passages correspondant au représentant du pattern Pat_a1 et à ses trois occurrences (Figure 5-11 à Figure 5-14). En revanche, le représentant du pattern Pat_a2 n'apparaît pas comme l'une des occurrences de Pat_a1. Ces fait peuvent s'analyser de la manière suivante : le premier filtrage, plaçant la priorité sur la maximisation de **totalValuation**, a conduit à l'extraction, comme meilleur pattern, de Pat_a1 dont l'ensemble d'occurrences est non seulement nombreux (par comparaison) mais aussi de quorum 2. Par contraste, le second filtrage, mettant la priorité sur la maximisation de la **longueur** des patterns, a amené à l'extraction, comme meilleurs patterns, de Pat_a2 et Pat_a3 qui, s'ils sont plus longs que que Pat_a1, possèdent moins d'occurrences et sont tous deux de quorum 1. Pat_a2 et Pat_a3 sont des patterns 'locaux' du point de vue du corpus (cf. plus haut), **spécifiques** respectivement à la première séquence et à la seconde séquence du corpus. A ce titre, les validités de Pat_a1, Pat_a2 et Pat_a3 sont **complémentaires**. Ceci met en évidence une autre composante de la « flexibilité » de FlExPat : la possibilité offerte à l'utilisateur d'effectuer successivement et interactivement différents traitements (stratégies de catégorisation) à partir d'un même graphe de similitude qui n'est donc pas recalculé à chaque fois. Cela est possible grâce la décomposition en deux phases distinctes de fep distinctes de FlExPat (appariement de facteurs et catégorisation).

Patterns du lexique d'Owens extraits

Bien que le corpus C_a ne soit constitué que de deux solos de longueur moyenne, un certain nombre de patterns du lexique d'Owens en sont extraits par Imprology/FlExPat ; d'autres en revanche y sont présents mais ne sont pas extraits (facteurs n'étant équipollents à aucun autre). Nous considérerons 2 exemples de la première catégorie et un de le seconde (« contre-exemple »), un intérêt supplémentaire étant de comparer les résultats obtenus avec le corpus C_a à ceux obtenus avec le corpus C_b (section 5.5.2.4).

Pattern 4Ea

Ce pattern est une descente chromatique de cinq croches (Figure 5-21). Soulignons que, sur la figure, le fait que la dernière note apparaisse comme une noire et non une croche résulte des heuristiques d'affichage de l'éditeur de partitions utilisé. Imprology/FlExPat extrait un pattern identique[56] :

- Représentant : Cool_Blues_Take1 {'5:3'--(5/2)--'6:1.5'}
- Occurrences :
  - Cool_Blues_Take2 {"16:3"--(5/2)--"17:1.5"}, similitude 34.5 ; puis :
  - (occurrence plus approximative :) Cool_Blues_Take2 {"31:3"--(5/2)--"32:1.5"}, similitude 32.5

Pattern 5B

Ce pattern à la position métrique spécifique consiste en un saut descendant (tierce mineure) suivi d'une remontée chromatique (voir Figure 5-21), le tout en croches. (Même remarque que ci-dessus en ce qui concerne l'affichage de la dernière note). Le motif extrait par Imprology/FlExPat, identique[57] au pattern 5B a les caractéristiques suivantes :

- Représentant : Cool_Blues_Take2 {'11:1.5'--2--'11:3.5'}
- Occurrences :
  - Cool_Blues_Take2 {"23:1.5"--2--"23:3.5"}
  - Cool_Blues_Take1 {"36:1.5"--2--"36:3.5"}
  - Cool_Blues_Take1 {"11:3.5"--2--"12:1.5"}

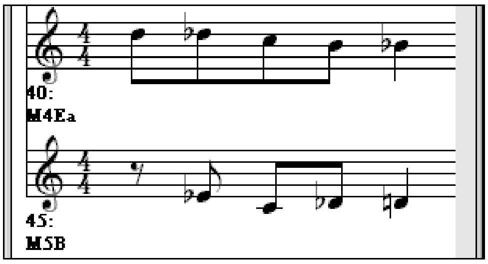


Figure 5-21. Patterns 4Ea et 5B du lexique d'Owens

---

[56] A une transposition près, ce qui est parfaitement adéquat.

[57] Ici aussi, à une transposition (chromatique) près.

Pattern 11A

Ce pattern, bien que possédant une occurrence dans le corpus C_a (Cool_Blues_Take2 {'22:2'--(7/2)--'23:1.5'}), n'en est pas extrait. Nous verrons qu'en revanche Imprology/FlExPat l'extrait à partir du corpus C_b.

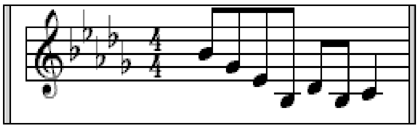


Figure 5-22. Pattern 11A du lexique d'Owens

Autres patterns extraits

Sans aller jusqu'à enumérer tous les autres patterns extraits par Imprology/FlExPat à partir du corpus C_a, nous montrons à la Figure 5-23 un pattern extrait, de longueur intermédiaire, dont la forme est intéressante : il s'agit d'un contour linéaire diatonique descendant (à l'exception de la dernière note) en croches.

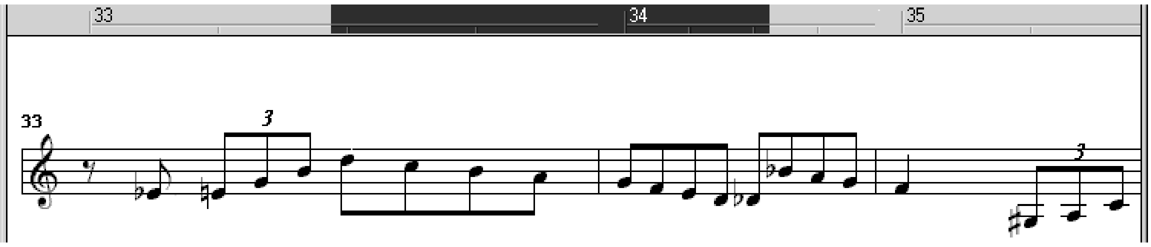


Figure 5-23. Représentant de pattern extrait : passage {'33:3'--(9/2)--'34:3.5'} du solo *Cool Blues Take 1*

L'occurrence (unique) de ce pattern fait intervenir un décalage métrique constant, mais cette fois-ci d'une valeur de 1 noire. Ce genre de décalage constant peut d'ailleurs être pris en charge de manière encore plus adéquate en utilisant la description individuelle dérivée 'Variation de niveau métrique entre chaque note et la suivante' proposée plus haut. Cet exemple montre à nouveau la prise en compte implicite d'une structure locale, en l'occurrence une descente diatonique caractéristique, sans pour autant que la description locale spécifique correspondante (cf 4.7.7.2) soit introduite explicitement.

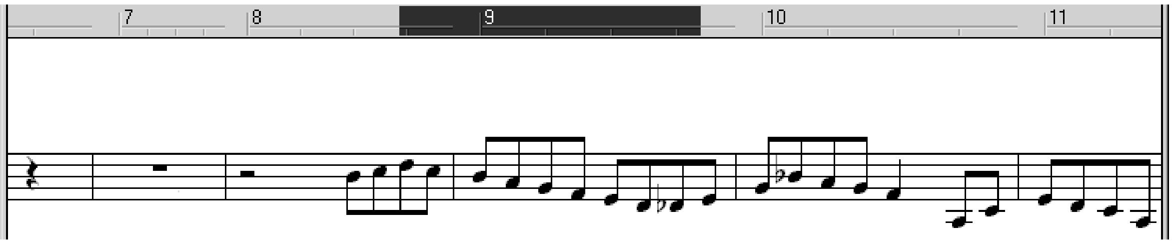


Figure 5-24. Occurrence (interne) du pattern : passage {"8:4"--(9/2)--"9:4.5"} du solo *Cool Blues Take 1*

5.5.2.3. Comparaison avec un jeu de description plus restreint

Nous comparons, dans cette section, les résultats qui viennent d'être présentés à ceux obtenus avec un sous-ensemble des descriptions. Il nous faut à nouveau préciser les conditions expérimentales, lesquelles, compte tenu de l'objectif de comparaison, ont été délibérément choisies identiques, en majorité, à celles qui précèdent.

5.5.2.3.1. Configuration

Les paramètres d'exécution pour la phase d'appariement de facteurs (FlExPat-AF) sont identiques à ceux du 5.5.2.2.1 :

- $ETAA=ETAA_{standard}$
- $m_{min}=3$, $m_{max}=40$.
- Fonction d'équipollence : définie par un seuillage à la valeur 0.7 sur le ratioIdentité.
- Fonction Δm définie par
- Fonction Recoup : définie par un seuillage (cf. 5.3.3.1) avec $\rho = 1/3.001$.

5.5.2.3.2. Jeu de description employé

- Descriptions individuelles :
    - Description relationnelle (dérivée) : **Intervalles chromatiques** formés avec la note précédente et avec la note suivante.
    - Description « particulière » : **Silences**.

5.5.2.3.3. Fonctions de contribution

Nous utilisons ici les mêmes fonctions que celles détaillées au 5.5.2.2.3.

5.5.2.3.4. Pondération des descriptions individuelles et contributions moyennes

Contributions moyennes globales :

- Pour le type d'appariement Remplacement : -0.22
- Pour le type d'appariement Insertion : -6.60
- Pour le type d'appariement Délétion : -6.60

Pondérations et répartition de la contribution moyenne sur les différentes descriptions pour le type d'appariement Remplacement :

- 'IntervallesChromatiques' :
    - Pondération : 1
    - Contribution moyenne : -2.43
- NoteVsSilence' :
    - Pondération : 3
    - Contribution moyenne : +2.21

5.5.2.3.5. Résultats obtenus

Les caractéristiques numériques générales sont les suivantes :

- (Les longueurs des différentes séquences du corpus après enrichissement de la représentation sont bien sûr les mêmes qu'avec le jeu de description précédent : 221 notes ou silences pour S_a1 et 233 pour S_a2.)
- Le graphe de similitude comporte (avant tout filtrage) **3218 sommets** et **11396 arêtes,** à comparer avec les valeurs précédentes, soit respectivement 1995 et 3829.
- la plus grande Etoile, au sens du 5.4.2, est de taille **28** (contre 10 précédemment).

- Le plus long pattern, au sens de la stratégie de catégorisation Centre d'Etoile et sans contrainte de quorum, est de longueur **29**, valeur très proche de celle (27) obtenue précédemment.

L'analyse des résultats de la phase de catégorisation fait ressortir **l'insuffisance** de ce jeu de description. L'examen du meilleur pattern, aussi bien au sens du totalValuation qu'à celui de la taille d'Etoile, est éloquente à ce sujet. Ce pattern Pat_a4 (passage {'33:3'--(3/2)--'33:4.5'} du solo *Cool Blues Take 1*) s'avère maximiser, sur l'ensemble des sommets du graphe de similitude, à la fois le totalValuation (strictement), soit[58] 21.97, et la taille d'Etoile, soit 28. Il s'agit d'une descente diatonique très « banale » sur 3 croches (cf. Figure 5-25). Le problème apparaît lorsqu'on examine les occurrences manifestement aberrantes, montrées en Figure 5-26 et Figure 5-27, compte tenu de leurs **structures rythmiques** respectives, **totalement différentes** de celle du représentant.

Figure 5-25. Représentant de pattern : passage {'33:3'--(3/2)--'33:4.5'} du solo *Cool Blues Take 1*

Figure 5-26. Première occurrence du pattern : passage {"17:3"--(5/2)--"18:1.5"} du solo *Cool Blues Take 1*

Figure 5-27. Seconde occurrence du pattern : passage {"3:3.5"--(1/2)--"3:4"} du solo *Cool Blues Take 2*

---

[58] A tire d'exemple, totalValuation est cette fois-ci exprimé comme le total des **ratioIdentités** (des différents alignements avec les facteurs — sommets — adjacents), et non plus comme le total des **similitudes** (valeurs des alignements).

5.5.2.4. Autres jeux restreints

Les résultats obtenus avec d'autres jeux de descriptions restreints tels que 'Durées' seulement, voire, à l'extrême, 'Silences' seulement, conduisent eux aussi à la même conclusion : l'insuffisance de la représentation. La raison en est totalement évidente pour ce second jeu restreint ; pour le premier, l'observation de la forme rhythmique des deux solos du corpus montre clairement que la liste de patterns extraits est dominée par des patterns constitués d'une série insignifiante de croches (jusqu'à 20), avec par suite une multitude d'arêtes sans signification musicale.

5.5.2.5. Morceaux « do majeur Blues » (10 séquences)

Ce corpus, désigné par C_b, se constitue des 10 morceaux suivants, parmi les 17 du groupe do majeur Blues (dont les caractéristiques quantitatives générales ont été données au Tableau 4-3, page 144) :

- S_b1 : *Cool Blues Take1*
- S_b2 : *Cool Blues Take2*
- S_b3 : *Cool Blues Take3*
- S_b4 : *Perhaps Take1*
- S_b5 : *Perhaps Take3*
- S_b6 : *Perhaps Take4*
- S_b7 : *Perhaps Take5*
- S_b8 : *Relaxin at Camarillo Take1*
- S_b9 : *Relaxin at Camarillo Take3*
- S_b10 : *Relaxin at Camarillo Take4*

5.5.2.5.1. Conditions expérimentales

Elles sont identiques à celles proposées plus haut (corpus C_a), au plan de la configuration de FlExPat, du jeu de descriptions, des fonctions de contribution et des pondérations employés.

Le calcul des contributions moyennes au plan global donne les valeurs suivantes :

- Pour le type d'appariement Remplacement : -4.68
- Pour le type d'appariement Insertion : -6.33
- Pour le type d'appariement Délétion : -6.33

On note que ces valeurs sont proches de celles obtenues avec le corpus C_a (respectivement -4.41, -6.60 et -6.60), ce qui est conforme à ce que l'on pouvait attendre.

5.5.2.5.2. Résultats obtenus

Caractéristiques numériques générales

- La longueur cumulée du corpus, après enrichissement de la représentation (silences), est de 1967 notes et silences.
- Le graphe de similitude comporte (avant tout filtrage) **11734 sommets** et **39228 arêtes**.
- La plus grande Etoile est de taille **35** (contre 10 avec le corpus C_a).
- Le plus long pattern, au sens de la stratégie de catégorisation Centre d'Etoile et sans contrainte de quorum, est de longueur **27**. Notons que ce nombre est limité par la valeur choisie pour m_{max} (soit 27), quoique, à l'examen des résultats, cette limitation ne se soit pas avérée significative.
- Le quorum maximal, soit 10, est atteint (voir ci-dessous).

Patterns de quorum 10

Les 58 patterns de quorum 10 sont de longueur au plus 4. Le « meilleur » d'entre eux, en termes de totalValuation, possède 21 occurrences pour un totalValuation atteignant 749. Il s'agit du passage {"1:3"--2--"2:1"}de Perhaps_Take3 : ré4-do4-si3-do4 sur 4 croches.

Filtrage

Nous allons considérer les résultats du filtrage effectué sur la liste des patterns (Etoiles) triée sur la combinaison de critères :

- longueur
- totalValuation
- taille d'Etoile (c'est à dire nombre de sommets adjacents + 1)
- quorum

La liste résultante contient (avant « post-post-filtrage ») 95 patterns, soit une réduction de taille d'un facteur supérieur à 100. Compte tenu de la possible « non-maximalité » de la longueur des 4 premiers patterns de la liste (soit 27), nous examinons la longueur immédiatement suivante, soit 26. Le meilleur pattern de longueur 26, que nous désignons par Pat_b1 a pour représentant le passage montré sur la Figure 5-28.

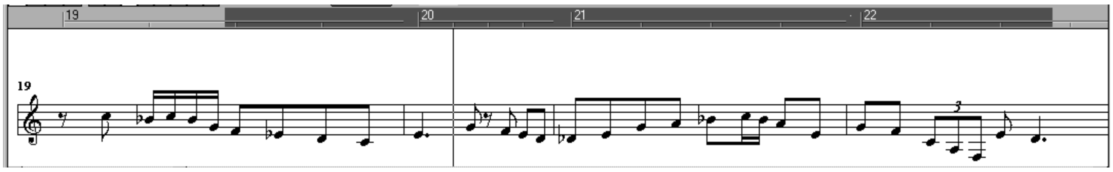


Figure 5-28. Représentant du pattern Pat_b1 : passage {"19:3"--14--"23:1"} du solo *Relaxin_at_Camarillo_Take4*

Ses 3 occurrences sont montrées dans les figures qui suivent, suivant l'ordre décroissant de leurs valeurs de similitude soit 213, 171.5 et 171.

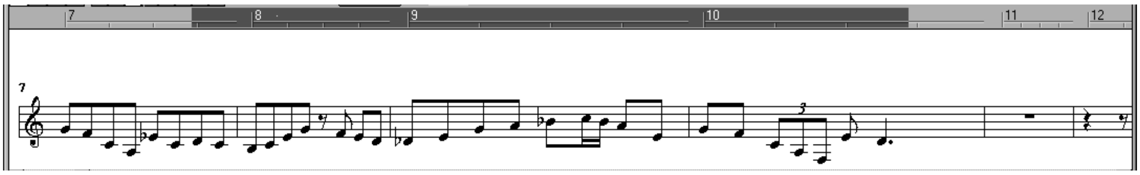


Figure 5-29. Occurrence 1 du pattern Pat_b1 : passage {"7:4"--13--"11:1"} du solo *Relaxin_at_Camarillo_Take4*

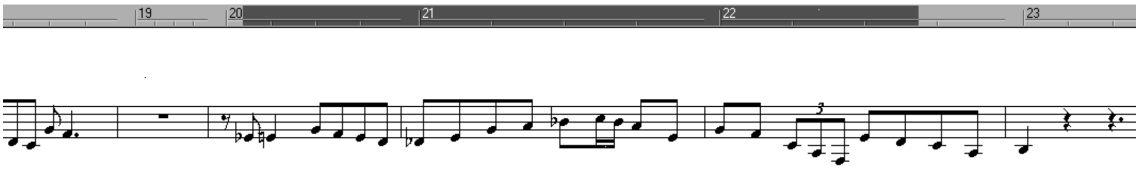



Figure 5-30. Occurrence 2 du pattern Pat_b1 : passage {"32:1.75"--(45/4)--"35:1"} du solo *Perhaps_Take3*

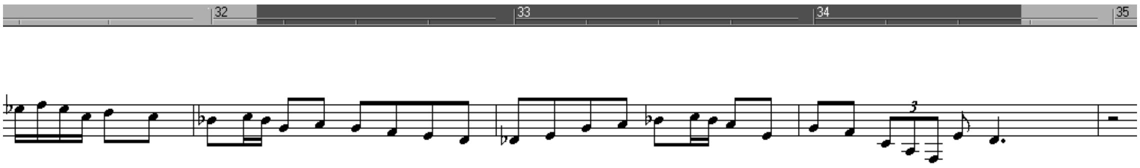


Figure 5-31. Occurrence 3 du pattern Pat_b1 : passage {"20:1.5"--(21/2)--"22:4"} du solo *Relaxin_at_Camarillo_Take3*

Ce pattern fait manifestement sens, et son exemple met en lumière les qualités du jugement de similitude du système, dues aux caractéristiques du MEVM. En particulier, la mise en correspondance, par ce dernier, des descentes diatoniques précédant le premier do dans le représentant et la première occurrence (fa-mib-ré et ré-do-si, respectivement), est tout à fait pertinente.

Patterns du lexique d’Owens extraits

Pattern 4Ea

7 occurrences sont trouvées pour ce pattern (à comparer à 2 pour le corpus C_a). Son quorum est **4**, et l’aspect le plus intéressant est la nette spécificité de ce pattern à 2 morceaux, Cool_Blues et Perhaps — il n’a aucune occurrence dans Relaxin' at Camarillo ; et, surtout, sa spécificité (partielle) à deux prises particulières : Cool_Blues_Take2 et Perhaps_Take1, qui totalisent 5 occurrences sur 7. Ainsi on voit ici apparaître la possibilité d'une **discrimination multi-niveau de morceaux** sur la base des **patterns extraits** (ou, le cas échéant, sur celle des **occurrences** de patterns donnés au départ). Nous verrons dans la suite d’autres exemples de ce genre, et reviendrons ultérieurement sur les possibilités de discrimination induites.

Pattern 5B

Ce pattern, possède **9** occurrences au travers du corpus C_b (contre 3 sur C_a), avec un quorum de **5**. Ici à nouveau, il s’agit d’un pattern rattaché spécifiquement aux deux morceaux Cool_Blues et Perhaps.

Pattern 11A

Ce pattern, qui n’était pas extrait dans C_a, l’est dans C_b avec **4** occurrences et un quorum de **3**. Les trois morceaux concernés sont **tous** des prises de Perhaps, ce qui indique une **forte spécificité** du pattern 11A.

Autres patterns extraits

Parmi les patterns extraits par Imprology/FlExPat à partir du corpus C_b, nous avons eu la « bonne surprise » de trouver également des patterns que nous avions personnellement et plus ou moins inconsciemment « extraits » à l'écoute de jazz improvisé. Il s'agit de patterns joués soit par C. Parker lui-même, soit par d'autres — par exemple le tromboniste J.J. Johnson, et ce, sur d'autres morceaux que ceux du corpus C_b, par exemple « *Drifting on a Reed* » ou encore « *Chasin' the Bird* ».

5.5.2.6. Enseignements généraux

Un certain nombre d'enseignements sont ressortis des résultats expérimentaux présentés. Certains sont repris synthétiquement dans cette sous-section finale, aux côtés de conclusions additionnelles, tirées d'autres expérimentations que nous avons effectuées mais pas été présentées dans ce mémoire. Ainsi, en plus d'une validation expérimentale des caractéristiques essentielles du modèle d'édition valué multi-descriptions et de l'algorithme FlExPat, nous notons :

- L'**intérêt de la séparation en deux phases** de FlExPat (appariement de facteurs puis catégorisation) : cette séparation permet notamment à l'utilisateur de mettre en œuvre, à partir d'un même graphe de similitude, plusieurs modalités ou stratégies d'extraction de patterns, et cela de manière interactive.

- Le **potentiel offert en matière de classification automatique**, lié à l'existence de patterns « locaux », rattachés spécifiquement à des sous-corpora spécifiques : morceau, groupe de prises (*takes*), groupe de morceaux ou encore groupe harmonique (tonalité + type de grille d'accords, par exemple si bémol majeur 'I Got Rhythm'). Comme on peut le constater de manière manifeste sur une **large majorité** des patterns montrés plus haut, pour un pattern donné la séquence décroissante des valeurs (similitudes) rattachées à chaque occurrence délimite de manière très nette les différentes séquences dans lesquelles elles se trouvent. A titres d'exemple, considérons le pattern Pat_b1 ci-dessus, dont le représentant appartient à S_b10 (*Relaxin at Camarillo Take4).* Ses occurrences, rangées par ordre décroissant de similitude avec le représentant, appartiennent respectivement à S_b10 (*Relaxin at Camarillo Take4*), S_b5 (*Perhaps Take3*) et S_b9 (*Relaxin at Camarillo Take3*). Les valeurs de similitude correspondantes, marquent une claire démarcation entre l'occurrence interne (similitude : 213) et les deux occurrences externes (similitudes : 171.5 et 171).

- Que, comme pour la localisation (cf 4.6.2), les qualités d'un algorithme d'extraction peuvent se formuler en termes de **sensibilité** et de **sélectivité**, qualités complémentaires et de statut classique (par exemple en ECD ou en apprentissage automatique). Nous avons notamment vu ou évoqué les effets d'une **ultrasensibilité** lorsque nous avons employé des jeux de descriptions restreints. A ce propos, soulignons plus généralement — comme nous l'avons fait pour la localisation — l'**influence décisive** du **choix de la représentation** (jeu de descriptions) sur les résultats obtenus par l'algorithme d'extraction de patterns. Soulignons par ailleurs l'importance majeure d'une **structure de correspondance variable** dans le cadre de la comparaison de facteurs (modèle d'édition) : les résultats expérimentaux que nous avons obtenus avec des paramétrages du MEVM exigeant l'égalité des facteurs (cf. sections 3.4.1.1 à 3.4.1.3) ou encore de type Hamming (section 3.4.1.6) le montrent de manière très nette. Ces paramétrages seront détaillés à la section 5.6.2 . Au niveau du modèle d'édition, les apports d'une **valuation**, à proprement parler (fonctions de contributions non constantes) ont également été mis très clairement en évidence expérimentalement, compte tenu des déficiences manifestes des résultats que nous avons obtenus en utilisant, au sein du MEVM, des fonctions de contribution constantes (distance de Levenshtein avec substitutions ; cf. section 3.4.1.7).

5.6. Discussion

La présente section est organisée comme suit : en premier lieu, FlExPat est resitué, du point de vue de ses caractéristiques générales, par rapport aux différents algorithmes d'extraction de patterns dans les séquences, généraux/biologiques ou musicaux, qui ont été présentés au 3.4. Ensuite, les possibilités offertes par FlExPat d'émuler un certain nombre de ces algorithmes sont explicitées (5.6.2). Puis nous discutons de son efficacité combinatoire, avant de faire au 5.6.4 et au 5.6.5 une comparaison critique entre FlExPat et les deux catégories d'algorithmes d'extraction, en particulier sous l'angle de la complexité algorithmique.

5.6.1. Caractéristiques générales

Hormis les possibilités spécifiques qu'offre l'algorithme au niveau du modèle de comparaison employé, la *flexibilité* de FlExPat tient aussi à la possibilité offerte de mettre en œuvre différentes approches pour la catégorisation.

Moyennant la mise en œuvre de l'approche "centre d'Etoile", FlExPat s'inscrit dans la catégorie des algorithmes d'extraction où les patterns sont des mots (sur le même alphabet que les séquences sources). Bien entendu, la mise en œuvre d'une approche différente peut conduire à des patterns extraits de nature différente, par exemple des expressions régulières.

Quoi qu'il en soit, dans le cadre du fonctionnement usuel de FlExPat présenté plus haut, la phase de détermination des représentants de pattern se fait à partir des blocs trouvés dans une phase antérieure. Ceci distingue FlExPat des algorithmes où les patterns (ou "modèles") sont déterminés en même temps que leurs ensembles d'occurrences (cf. par exemple les algorithmes présentés aux sections 3.4.1.6.1 et 3.4.1.7.1).

Un avantage majeur de FlExPat est lié à la séparation des deux phases, phase d'appariement de facteurs et phase de catégorisation. Cela offre à l'utilisateur le potentiel d'explorer interactivement, à partir du graphe de similitude issu de la première phase, plusieurs approches de catégorisation ou plusieurs jeux de paramètres différents pour une même approche. Ces paramètres sont par exemple, dans le cadre de l'approche « Centre d'Etoile », la valeur du quorum ou les options de post-filtrage sur les patterns.

5.6.2. Subsomption et émulation d'algorithmes existants

Comme nous l'avons vu, le MEVM subsume un certain nombre de modèles classiques qui peuvent donc, suivant les besoins, être utilisés à sa place dans FlExPat. Il est en particulier possible d'extraire des patterns en fondant la définition de l'équipollence, et donc celle de la fonction de similitude, sur :

- un modèle d'édition à fonctions de contribution constantes (algorithmes présentés à la section 3.4.1.7) ;
- un modèle de similitude de type Hamming (algorithmes présentés en section 3.4.1.6) ;
- un modèle de similitude exigeant l'égalité des facteurs (algorithmes présentés dans les sections 3.4.1.1 à 3.4.1.3).

5.6.2.1. Modèle d'édition à fonctions de contribution constantes

FlExPat offre en particulier la possibilité de travailler avec un modèle d'édition à fonctions de contributions **unitaires**, correspondant à la distance de Levenshtein avec substitutions — cf. 3.2.5.1. Il peut ainsi émuler l'algorithme LeMoivre dans le cadre d'un fonctionnement particulier : celui où l'on exige que les patterns (modèles) aient au moins une occurrence exacte dans le corpus [Sagot 1996 ; p. 201].

Cela peut être fait en configurant FlExPat de la manière suivante (on note e le nombre maximum « d'erreurs » permises — indels ou substitutions) :

- $\text{ETAA} = \text{ETAA}_{\text{standard}}$
- fonctions de contributions: elle sont pseudo-booléennes (cf. 4.7.4.3), pouvant prendre uniquement les deux valeurs ConstId et ConstDiff, qui représentent respectivement :
    - la contribution de toute identité, et :
    - la contribution de toute substitution ou indel

  Il est commode de prendre ConstId = 0 et ConstDiff = -1 en vue de la définition du critère d'équipollence (voir ci-après). Pour tout couple (F,F') de facteurs, Simil(F,F') est donc un entier négatif ou nul.
- Critère (fonction) d'équipollence :

$$\text{Eq}(F,F') \Leftrightarrow \text{Simil}(F,F') \geq -e$$

- Δm = e (fonction constante)
- Recoup = FAUX (fonction constante)

5.6.2.2. Modèle de similitude de type Hamming

FlExPat peut également émuler l'algorithme HMoivre, là aussi dans le cadre du fonctionnement particulier où l'on exige que les patterns (modèles) aient au moins une occurrence exacte dans le corpus. Cela peut être fait en configurant FlExPat de la manière suivante (on note e le nombre maximum d'erreurs permises) :

- ETAA = {Remplacement} (modèle d'édition dégénéré)
- Fonction de contribution, associée au type d'appariement Remplacement : pseudo-booléenne, valant :
    - ConstId pour toute identité
    - ConstDiff pour toute substitution

  Ici aussi, il est commode de prendre ConstId = 0 et ConstDiff = -1 en vue de la définition du critère d'équipollence qui suit :

- Critère (fonction) d'équipollence :

$$\mathrm{Eq}(F,F') \Leftrightarrow \mathrm{Simil}(F,F') \geq -e$$

- $\Delta m = 0$ (fonction nulle)
- Recoup = FAUX (fonction constante)

5.6.2.4. Modèle de similitude exigeant l'égalité des facteurs

Ce dernier cas, extrême, correspond à la recherche de blocs de facteurs répétés exactement. Cette recherche, pouvant naturellement être vue comme un cas particulier des deux qui précèdent avec e=0, peut être effectuée avec FlExPat en adoptant par exemple la configuration suivante :

- ensemble des types d'appariements autorisés, fonction de contribution, fonction Δm, fonction Recoup : identiques à celles du 5.6.2.2.
- Critère (fonction) d'équipollence :

$$\mathrm{Eq}(F,F') \Leftrightarrow \mathrm{Simil}(F,F') = 0$$

5.6.2.5. Discussion

Bien entendu, il ne s'agit nullement là des conditions normales d'utilisation de FlExPat. Aussi, en ce qui concerne la recherche de blocs de facteurs répétés exactement, la complexité en temps théorique, grossièrement majorée par $O(L^2 \times m_{max}^2)$ — cf. 5.3.8.1 — est-elle moins satisfaisante que celle atteinte par des algorithmes dédiés à ces conditions particulières (cf. notamment 3.4.1.2).

Précisons d'ailleurs que, pour ce qui concerne l'extraction de patterns basés sur des répétitions exactes (blocs de mots répétés), nous avons implémenté dans Imprology une version de KMR (cf. 3.4.1.1), algorithme que nous avons généralisé pour pouvoir travailler sur des corpus non unitaires, de manière analogue à ce qui a été proposé par Landraud, Avril & Chretienne (1989).

En revanche, avec des modèles de comparaison plus riches — en particulier la distance de Levenshtein avec substitutions — la complexité de FlExPat peut s'avérer significativement meilleure que celle des algorithmes cités, comme nous le verrons au 5.6.4.

5.6.3. Efficacité combinatoire

Dans l'analyse de l'efficacité combinatoire de la phase d'appariement de facteurs de FlExPat, deux aspects interdépendants peuvent être pris en compte :

- Celui lié à l'ordre dans lequel les paires de facteurs sont prises en compte en vue de leur comparaison
- Celui lié à l'utilisation de techniques basées sur la programmation dynamique pour la comparaison de facteurs

Ici encore, pour simplifier l'exposition plaçons-nous dans le cadre du modèle d'édition standard et supposons que le corpus est unitaire : C={S}. Comme nous l'avons vu, l'ordonnancement appliqué à la prise en compte des paires de facteurs de S (cf. 5.3.4.2) permet que la comparaison entre toute paires de facteurs (S[[i,m]],S[[i',m']]) se fasse en temps constant. On montre aisément que le mécanisme mis en œuvre à ce niveau dans FlExPat-AF est exactement équivalent à celui mis en œuvre dans la famille d'algorithmes d'alignement global basés sur la programmation dynamique (AG2SPD, cf. 3.2.6.1). Or il est possible de quantifier le gain combinatoire (en termes de complexité en temps) que ces algorithmes permettent par rapport à une approche "en force-brute". Celle-ci est l'approche où l'on on parcourrait explicitement l'espace de tous les alignements possibles entre les deux séquences comparées.

Il s'agit d'estimer le nombre d'alignements possibles entre deux séquences X et Y de longueur donnée (avec $\mathrm{ETAA} = \mathrm{ETAA}_{\mathrm{standard}}$). Si *X* et *Y* sont de même longueur L, le nombre d'alignements possibles entre *X* et *Y*, noté *Nal*(*L*,*L*), vérifie [r213][Laquer 1981; Waterman 1995] :

Equation 5-7

La preuve s'obtient à partir de la formule de récurrence générale [Waterman 1995] *:*

$$Nal(n,m)=Nal(n-1,m)+Nal(n-1,m-1)+Nal(n,m-1)$$

Moyennant l'approximation de l'Equation 5-7, nous obtenons par exemple :

$$Nal(100,100) \approx 10^{76} \text{ et } \mathrm{Nal}(1000,1000) \approx 10^{767,4}$$

Sur une machine dont la fréquence d'horloge serait de 1 GHz, et dont nous ferions l'hypothèse (irréaliste) qu'elle est capable d'évaluer un alignement en un seul cycle d'horloge, le temps requis pour comparer deux séquences de longueur 1000 dépasserait les 10^{750} années…

Par contraste, trouver l'alignement optimal entre X et Y en utilisant un algorithme de type AG2SPD prend un temps $O(L^2)$ — cf. 3.2.6.1. (Ainsi, le temps mis par la même machine pour trouver l'alignement optimal entre X et Y en utilisant un algorithme de type AG2SPD, serait de l'ordre de : $10^6 \times 10^{-9}$ soit 1 ms…)

Pour **chaque** paire de facteurs comparés dans FlExPat-AF, un gain en complexité très important intervient. Même si un calcul exact du gain total serait très complexe voire impossible, on peut donc affirmer que les mécanismes spécifiques mis en œuvre dans FlExPat-AF permettent, par rapport à une approche naïve, un facteur de gain combinatoire considérable. Ce facteur est croissant avec en fonction de la taille du corpus, et on peut supposer que cette croissance est très rapide, peut-être exponentielle.

5.6.4. Comparaison avec les algorithmes généraux ou biologiques

Du point de vue du modèle de comparaison qu'il peut employer, FlExPat est plus général que tous les autres algorithmes détaillés plus haut. A l'inverse, dans le cas où la phase de catégorisation suit l'approche « Centre d'Etoile » telle que nous l'avons présentée au 5.4.2, une limitation peut être vue quant à forme des patterns que FlExPat extrait. En effet, dans ce cas, ceux-ci sont des mots alors que certains autres algorithmes sont capables d'extraire des structures plus générales, par exemple des expressions régulières [Wang et al. 1994] ou des produits de sous-alphabets dans le cadre d'une couverture combinatoire pondérée [Sagot & Viari 1996]

Une comparaison pure et simple de la complexité algorithmique de FlExPat avec celle des différents algorithmes passés en revue au 3.4 n'aurait pas grand sens, les objectifs poursuivis n'étant pas les mêmes (définitions distinctes de l'équipollence et de la similitude). En revanche, il semble intéressant de comparer la complexité en temps de FlExPat avec celle de l'algorithme le plus semblable du point de vue de ces définitions. Cet algorithme est LeMoivre (3.4.1.7.1),extrayant également des patterns qui sont des mots sur le même alphabet que les séquences sources, et utilisant un modèle d'édition.

Nous avons vu que sa complexité en temps, notée ici CplxLeMoivre, est en $O(L.K^{2e+1}.|\Sigma|^{e+1})$ où :

- L est la longueur cumulée du corpus
- K est la longueur maximale des modèles (patterns) recherchés — ou celle du plus long modèle trouvé
- e est la limite supérieure autorisée de la distance de Levenshtein avec substitutions entre un pattern et une occurrence (nombre maximum de substitutions ou indels).
- Σ est l'alphabet sur lequel on se place

Supposons K petit devant L ; si |Σ| est également petit devant L, en particulier si c'est une constante faible, CplxLeMoivre est extrêmement satisfaisante puisqu'on peut l'assimiler à une complexité linéaire par rapport à la taille du corpus. Ainsi sur des séquences d'acides nucléiques (alphabet à 4 nucléotides) l'efficacité est grande. En revanche, dès lors que |Σ| est du même ordre de grandeur de L (cf. 4.7.9) la complexité devient **$O(L^{e+2}. K^{2e+1})$** (ou pire, suivant la taille de Σ).

Cela doit être comparé à la valeur CplxFlExPat = $O(L^2 \times K^2 \times \|ETAA\|)$ majorant la complexité de FlExPat (cf. 5.3.8 et 5.4.2.5), si l'on désigne la longueur maximale des patterns recherchés par K au lieu de m_{max}. Nous supposons que la phase de catégorisation emploie l'algorithme Centre d'Etoile explicité au 5.4.2. Pour que la comparaison soit la plus valide possible, il faut supposer que $ETAA=ETAA_{standard}$, d'où $\|ETAA\|=|ETAA|=3$, ce qui donne **$O(L^2 \times K^2)$**. Cela signifie que même avec une seule erreur permise, l'ordre de grandeur CplxLeMoivre est supérieur d'une unité à celui de CplxFlExPat : cubique en L vs. quadratique, bien que ce dernier utilise un modèle de comparaison plus général (et plus riche) et ne fixe **pas** de limite au nombre d'indels ou de substitutions. Le contraste est spectaculaire si l'on passe par exemple à 10 erreurs, un genre de valeur fréquemment rencontrée dans les applications musicales de FlExPat — par exemple 3 indels et 7 substitutions (« modérées ») entre deux facteurs de longueur 15. Dans ce cas-là en effet, CplxLeMoivre devient $O(L^{12}. K^{21})$ — ce qui est bien sûr totalement impraticable même pour des valeurs modérées de L — CplxFlExPat restant pour sa part en $O(L^2 \times K^2)$.

Si K est du même ordre de grandeur que L, l'écart de complexité entre FlExPat et LeMoivre n'en est que plus fort — $O(L^4)$ contre $O(L^{33})$ dans le cas qui vient d'être envisagé !… Plaçons-nous à présent du point de vue de la **complexité en espace**. Cette complexité est en $O(L.K^{2e+1})$ ou $O(L.K^{2e}.|\Sigma|^e)$ suivant la version de l'algorithme utilisée, à confronter avec celle de FlExPat dont nous avons vu (5.3.8.4) qu'elle est majorée par **$O(L^2 \times m_{max}^2)$,** indépendamment de la taille de l'alphabet. Comme pour la complexité en temps, la mémoire requise par LeMoivre reste tout à fait raisonnable tant que K reste petit devant L, tout comme la taille de Σ pour la 2^e version de l'algorithme. En revanche dès que K et/ou |Σ| atteignent l'ordre de grandeur de L, l'algorithme est limité à un très petit nombre d'erreurs — indépendamment d'ailleurs de la taille du modèle en cours de traitement. Ici encore, l'écart de complexité entre FlExPat et LeMoivre devient spectaculaire dès que l'on accepte, dans celui-ci, plus d'une ou deux erreurs.

5.6.5. Comparaison avec les algorithmes musicaux

Par rapport aux algorithmes statiques d'extraction de patterns (section 3.4.2.1), une des caractéristiques distinctives de FlExPat est d'éviter l'inefficacité combinatoire considérable qui entache un système comme EMI. En particulier, d'après les informations fournies par Cope, la recherche des blocs de segments répétés de longueur k se fait de manière indépendante des recherches de blocs répétés de longueur k' où k'<k. Remarquons que, compte tenu de la forme du modèle de comparaison de segments utilisé, cette redondance combinatoire pourrait être évité au travers de la mise en œuvre d'un algorithme d'extraction à base d'arbres des suffixes ou de DAWG, voire un algorithme de type KMR (cf. 3.4).

Par rapport aux algorithmes en-ligne (section 3.4.2.2), l'un des avantages importants d' Imprology/flExPat est de ne pas exiger une [pré-]segmentation des séquences sources, encore moins une segmentation **unique**, comme c'est le cas tant pour Cypher que pour l'algorithme de Pennycook, Stammen & Reynolds (1993) que nous avons baptisé PSR93. Or, même si un certain nombre de résultats ont été obtenus (cf. par exemple [Stammen & Pennycook 1994]), on ne dispose pas à ce jour d'algorithmes entièrement satisfaisants de segmentation musicale, même en temps différé. L'une des raisons majeures à cela est que, même d'un point de vue cognitif, il existe en général une multitude de segmentations possibles de la même séquence musicale. Par suite la seule solution viable pour un algorithme nécessitant de segmenter serait de gérer simultanément plusieurs schémas alternatifs de segmentation, ce que ne permet pas l'algorithme tel qu'il est décrit par Pennycook et al. dans les communications citées. Une conséquence supplémentaire est une limitation du champ d'application de ces algorithmes à des séquences mélodiques ou à des séquences issues d'un domaine d’application ou une telle segmentation est possible à la fois d’un point de vue théorique[59] et d’un point de vue pratique

En revanche, une limitation par rapport à PSR93 et Cypher est que FlExPat ne permet pas un fonctionnement en-ligne (temps-réel), du moins dans sa forme et son implémentation actuelles.

Comparons à présent FlExPat aux algorithmes des deux catégories réunies.

Un avantage spécifique majeur de FlExPat et de son implémentation dans Imprology est de permettre, au travers le modèle d'édition valué multi-descriptions *:*

- la prise en compte simultanée et pondérée de multiples descriptions musicales (MEVM), et :
- l’utilisation d’un modèle riche de comparaison de séquences, à savoir un modèle d'édition à contributions variables.

Certes l’algorithme de Pennycook, Stammen & Reynolds (1993) utilise également un tel modèle d’édition (qui a d’ailleurs influencé la conception du MEVM). Cependant, ce modèle ne fait intervenir que deux dimensions fixées au départ, et le paradigme de combinaison de ces deux dimensions est également fixe (distance euclidienne). Il n'est donc possible, ni de choisir un autre jeu de descriptions, ni même de pondérer différemment les deux dimensions imposées.

Par ailleurs, au travers de son articulation en une phase de construction de graphe de similitude et une phase de catégorisation à partir de ce graphe, FlExPat permet un contrôle fin sur le processus d’extraction à proprement parler. A l’inverse, à titre d’exemple dans un système comme PSR93 un [représentant de] pattern est toujours, schématiquement, la **première** occurrence du pattern, ce qui introduit un biais[60].

En outre, par comparaison avec d’autres algorithmes musicaux nous avons, d’une manière générale, assoupli et paramétré dans FlExPat différents critères — cf. par exemple le critère d’équipollence fixe, « 4 notes concordantes », employé dans Cypher…

---

[59] Et aussi **sémantique**, autrement dit la structure de groupement induite par la segmentation doit avoir un sens, de même que la *structure de phrase* en a un en musique.

[60] Ce biais est à mettre en parallèle avec la critique faite à l’analyse paradigmatique et rapportée dans [Anagnostopoulou 1997].

Chapitre 6. CONCLUSION ET PERSPECTIVES

Les apports et contributions de cette thèse s'articulent suivant plusieurs axes interdépendants. Nous choisissons de grouper dans une sous-section spécifique les contributions liées aux aspects musicaux de nos travaux, même si cette séparation est quelque peu artificielle.

Nos travaux peuvent être situés dans de la partie de l'Extraction de Connaissances à partir de Données qui s'applique à des données présentées sous forme de **séquences**, les patterns que l'on cherche à localiser ou à extraire étant eux-mêmes des entités séquentielles. Les concepts et algorithmes généraux que nous avons proposés et implémentés pour la découverte de régularités dans les séquences, s'inscrivent dans la lignée d'un ensemble de travaux, à composante combinatoire plus ou moins forte, que nous avons répartis en deux groupes : ceux — nombreux — relevant (principalement) de l'informatique théorique et de la bioinformatique, et ceux relevant directement de l'informatique musicale. Cela a notamment permis de rapprocher certaines techniques ad hoc proposées en informatique musicale, de techniques générales d'algorithmique sur les séquences. Pour les deux groupes de travaux, l'unification de notre présentation, au-delà d'une unification de la terminologie, a été effectuée autour de la notion centrale de **modèle de comparaison entre entités séquentielles.** Nous avons donc passé en revue un certain nombre de modèles de comparaison proposés dans la littérature tant musicale qu'extra-musicale, allant des modèles les plus simples (structure de correspondance fixe) aux plus riches (structure de correspondance variable, valuation).

Une contribution principale de nos travaux concerne le **modèle général de comparaison entre entités séquentielles** appelé ***modèle d'édition valué multi-***

descriptions (MEVM), que nous avons proposé et intégré à des algorithmes effectuant la localisation et surtout l'extraction de patterns [Rolland & Ganascia 1998a]. Ce modèle est notamment conforme au modèle général proposé en psychologie par Tversky (*modèle des contrastes*). La prise en compte simultanée de multiples descriptions au sein du MEVM est rendue nécessaire par une phase d'**enrichissement (ou de changement) de la représentation** des séquences sources et de leurs éléments ; nous insérons cette phase, dont nous avons justifié théoriquement et expérimentalement la pertinence voire la nécessité, au sein du processus de découverte de régularités dans les séquences, de manière soit préalable (prétraitement), soit dynamique. Dans le MEVM, la **pondération** des différentes descriptions au sein des fonctions de contributions associées à chaque type d'appariement autorisé (modèle d'édition ; cf. sections 3.2.4.2 et 4.7.3), offre la possibilité d'adopter différents **points de vue** dans le cadre du calcul de similitude entre entités séquentielles et, par suite, dans celui de la localisation et de l'extraction de patterns. La prise en compte de descriptions multiples et pondérées matérialise, dans un contexte de découverte de régularités tel que celui de l'analyse musicale, une **simulation** ou une **modélisation cognitive** du jugement de similitude humain, en l'occurrence le jugement de similitude entre passages musicaux du musicologue [Rolland & Ganascia 1996b]. Nous avons mené un travail de réflexion général sur la notion de description dans le contexte de données (connaissances) organisées en séquences ; cette réflexion a débouché sur la formalisation d'une **typologie** de descriptions et de catégories de descriptions distinguant, notamment, descriptions *globales*, *locales*, et *individuelles* ; descriptions *de base*, *primitives*, et *dérivées* ; enfin, pour les descriptions individuelles : descriptions *intrinsèques* et descriptions *relationnelles*.

Une autre contribution principale se rapporte à l'**algorithme général d'extraction de patterns** que nous avons conçu et implémenté [Rolland 1998] (il s'agit en fait, en toute rigueur, d'une *famille* d'algorithmes) : ***FlExPat*** est à notre connaissance le seul algorithme d'extraction général permettant d'utiliser un modèle d'édition à contributions (ou coûts) variables comme le MEVM. Un autre algorithme basé sur un modèle particulier de cette catégorie est celui proposé dans le domaine musical par Pennycook, Stammen & Reynolds (1993). Plusieurs limitations-clés rattachées à cet algorithme ont été mises en évidence, que FlExPat permet d'outrepasser. A l'inverse, par rapport à l'algorithme de Pennycook, Stammen & Reynolds, FlExPat possède le désavantage de ne pas permettre un fonctionnement en temps-réel, du moins au stade actuel. Nous avons vu que la complexité de FlExPat — quadratique, au pire et en théorie, par rapport à la longueur cumulée du corpus et celle des plus longs patterns extraits — peut être très significativement meilleure que celle d'algorithmes offrant pourtant des possibilités nettement plus restreintes quant au modèle de comparaison utilisable : cela est vrai dès que l'on a affaire à des « gros » alphabets, ce qui, comme nous l'avons explicité, se produit usuellement dès lors que l'on enrichit la représentation.

Un certain nombre de résultats ont été présentées essentiellement dans le domaine musical (cf. ci-après), fournissant une validation expérimentale des différentes caractéristiques du MEVM et de FlExPat.

Aspects musicaux

Au travers les applications que nous avons faites de nos concepts, modèles et algorithmes dans le domaine musical, notre recherche apporte une contribution au champ de recherche désigné par 'Intelligence Artificielle et Musique' (cf. chapitre d'introduction) et plus généralement à celui de l'informatique musicale. Nos travaux s'inscrivent dans un courant de recherche qui confère une place centrale à la notion de 'sens commun musical', et à la modélisation (ou la simulation) de la perception et de la cognition musicales intervenant dans les activités musicales, au sens large, que l'on souhaite faire accomplir aux systèmes : écoute, analyse, composition, improvisation, accompagnement, interprétation, etc. (cf. par exemple [Widmer 1994 ; 1992 ; Ramalho & Pachet 1994 ; Ramalho, Rolland & Ganascia 1998 ; Horowitz 1995]).

L'instanciation de nos concepts, techniques et algorithmes dans le cadre de problèmes musicaux prennent la forme, principalement :

- d'un modèle de comparaison entre entités séquentielles musicales (passages mélodiques particulièrement), et :
- d'algorithmes permettant de localiser les occurrences de patterns musicaux (passages, phrases) donnés au départ, et surtout, d'extraire de nouveaux patterns, mélodiques ou par exemple harmoniques (séquences d'accords) ;

L'implémentation de l'ensemble de ces concepts, techniques et algorithmes, sous forme d'un système baptisé ***Imprology*** [Rolland & Ganascia 1996a ; 1998a], a notamment permis d'obtenir un certain nombre de résultats expérimentaux dont une partie est présentée dans le présent mémoire. La conclusion centrale tirée de ces expérimentations est qu'**un enrichissement (ou un changement) pertinent de la représentation conduit à une amélioration très significative des résultats** en matière de découverte de régularités dans les séquences, amélioration tant qualitative que quantitative. Plus spécifiquement, ces expérimentations ont permis de valider expérimentalement les aspects essentiels du MEVM :

- Structure de correspondance variable (modèle d'édition).
- Valuation (fonctions de contributions, non constantes).
- Prise en compte simultanée de multiples descriptions (primitives / dérivées ; globales / locales / individuelles ; intrinsèques / relationnelles).
- Pondération ajustable des différentes descriptions, et notion associée de point de vue.

Les résultats ont également fourni une validation expérimentale des caractéristiques principales de FlExPat :

- Possibilité d'utiliser un modèle d'édition valué (MEVM).
- Séparation en deux phases algorithmiques distinctes : phase d'appariement de facteurs puis phase de catégorisation.
- Approche « Centre d'Etoile » de catégorisation que nous avons proposée et implémentée.
- Gestion des recoupements entre facteurs et/ou représentants de patterns, tout au long des deux phases algorithmiques.

Au chapitre expérimental, soulignons l'une des caractéristiques importantes de nos travaux, du point de vue de leur contribution au domaine de « l'analyse musicale assistée par ordinateur » [Pachet 1998] : nous avons choisi de conduire des expérimentations sur des problèmes (corpus) en « vraie grandeur », à la différence de nombre de travaux dans ce domaine. Cette vision quantitative, et non simplement qualitative, de la « musicologie assistée par ordinateur » [Bel & Vecchione 1993] est celle que soutiennent avec énergie des institutions tels que le *Center for Computer Assisted Research in the Humanities* (CCARH, Menlo Park, Californie) et qui sous-tend, entre autres, des projets comme *Humdrum* [Huron 1994] ou encore *the Music Library of the Future* [Pennycook 1995].

A la croisée de plusieurs chemins...

Nos travaux se situent donc à l'intersection de plusieurs disciplines ou sous-disciplines :

- ECD et fouille de données

- Informatique théorique
- Intelligence artificielle et reconnaissance des formes
- Psychologie cognitive — en particulier, pour les applications musicales : la psychologie de la perception, de l'écoute, de la composition, de l'improvisation, et plus généralement de la cognition musicales.
- Informatique musicale
- « Théorie musicale » au sens large

Perspectives

Un certain nombre de perspectives peuvent être dessinées au-delà des travaux présentés. Nous les articulons selon les thématiques suivantes :

Indexation

L'indexation de bases de données organisées en séquences fait depuis plusieurs années l'objet de nombreux travaux de recherche et de développement, notamment pour la mémorisation et l'accès aux données multimédia (vidéo, son etc.). L'indexation et l'organisation des séquences dans une base de séquences, et la récupération dans une telle base (requête), sont des champs de recherche auxquels nous souhaitons contribuer, sur la base des travaux que nous avons effectués dans le domaine, connexe, de la découverte de régularités dans les séquences. En particulier, les concepts et algorithmes que nous avons proposés pour la comparaison de séquences offrent la possibilité de développer des paradigmes novateurs de récupération basée sur le contenu (*content-based retrieval*). Dans ce type de récupération, la requête (ou une partie de la requête) est constituée d'une « sonde » de même nature que le contenu de la base, par exemple un son pour une base de données sonore [Hashimoto, Qi & Chang 1996], ou un fragment d'image, par exemple un visage, pour une base de données graphique, et la récupération se fonde typiquement sur la similitude (*loco-globale*, dans la formalisation que nous avons proposée) déterminée entre la sonde et les différents enregistrements de la base. L'idée centrale est que non seulement les descriptions sur lesquelles s'appuie le traitement de la requête pourraient être choisies et pondérées par l'utilisateur, mais que, de plus, ces choix et ces pondérations pourraient être effectuées automatiquement et dynamiquement par le système en fonction de critères ou de stratégies donnés [Rolland & Ganascia 1996a].

Prédiction, anticipation

Un autre prolongement important du travail accompli jusqu’à présent porte sur les problématiques, omniprésentes lorsqu’on a affaire à des séquences temporelles, de la prédiction et l’anticipation. A côté des applications classiques de la prédiction (anticipation boursière ou économique par exemple) existent des champs d’application prometteurs, dont notamment :

- Les agents d’interface (cf. 2.1.2.2), qui peuvent à partir de la prédiction des prochaines actions de l’utilisateur, soit accomplir celles-ci à sa place, soit lui fournir une aide contextuelle tenant compte de ses actions passées et futures (et des buts supposés que le système y associe)

- Les jeux interactifs (cf. 2.1.2.3), dans lesquels la stratégie du joueur informatique peut utilement s’appuyer sur l’anticipation des actions du ou des joueur(s) humain(s).

- Les systèmes musicaux interactifs (cf. 2.1.3.3.1), sur lesquels nous reviendrons plus bas.

De très nombreux travaux portent sur le développement de tels agents ou systèmes, ce qui atteste de la pertinence de travaux de recherche visant à améliorer les techniques existantes en matière de prédiction et d’anticipation dans les séquences. Nous avons d’ailleurs mentionné l'application de techniques d’extraction de patterns — de façon certes simpliste — pour la prédiction et d’anticipation dans les séquences temporelles : Minasi (1991) dans le domaine des jeux (cf. 2.1.2.3) et Rowe (1993) dans le domaine musical.

Les travaux projetés pourront se baser non seulement sur nos acquis (Imprology/FlExPat), mais aussi sur des algorithmiques spécialisées (par exemple celle des arbres de suffixes ou des DAWG, cf. section 3.4.1.2) conçues pour des sous-problèmes plus simples.

Applications

Un certain nombre d’applications directes des techniques développées dans le cadre de nos travaux ont été présentées. Outre celles-ci, nous souhaitons effectuer ou accompagner d’autres expérimentations sur la découverte de régularités dans des séquences :

D'un côté, nous aimerions continuer les expérimentations sur des données musicales en faisant appel à d'autres corpora et à d'autres genres musicaux. Effectuer ce type de découverte représente une tâche si immense pour un musicologue humain qu'une faible proportion des nombreux corpora musicaux disponibles (par exemple, [White 1978] dans le genre jazz) ont fait à ce jour l'objet d'une telle analyse. Ainsi une réelle demande existe qu'Imprology pourrait contribuer à satisfaire. Les possibilités très prometteuses d'application, esquissées plus haut (section 5.5.2), en matière de classification automatique de morceaux — en fonction, par exemple de leur auteur ou de leur genre — pourront aussi avantageusement être explorées. Mongeau & Sankoff (1990) suggèrent que cela pourrait même permettre de traiter des questions de Copyright musical. Parallèlement, la comparaison des résultats de la découverte de régularités dans des séquences musicales issues de musiciens ou genres différents, peut fournir des enseignements très intéressants relativement à la notion de style.

De l'autre, il s'agirait de transposer le travail théorique, expérimental et de programmation informatique effectué sur le domaine musical, aux autres domaines où les données se présentent sous une forme séquentielle, et en particulier à la biologie. Une telle transposition suppose de :

- répertorier les descriptions importantes devant intervenir dans la détermination de la similitude entre séquences / facteurs ;
- concevoir puis implémenter les algorithmes effectuant automatiquement, le cas échéant, le calcul des descriptions dérivées (supplémentaires) à partir des descriptions de base ;
- définir les fonctions de contribution associées à chacun des types d'édition devant être utilisés (modèle d'édition valué multi-descriptions). Implémenter ensuite ces fonctions ;

Ce sont là autant de tâches que nous avons effectuées pour l'application musicale. Bien entendu, comme nous l'avons indiqué, un certain nombre de fonctionnalités existant déjà dans Imprology offrent une assistance efficace dans ces tâches. C'est le cas par exemple du calcul automatique du **graphe de dépendance** entre descriptions, lequel permet entre autres de détecter automatiquement des incohérences (exemple : une description *d* calculée à partir d'une description *d'*, elle-même calculée à partir de *d*).

---

[61] Nous avons vu plus haut, notamment au chapitre d'introduction, que cette automatisation, partielle ou totale, n'était pas forcément indispensable.

Autres

« Last but not least », indiquons d'autres perspectives n'entrant pas dans la classification ci-dessus. La première est générale, les autres étant, au contraire, rattachées spécifiquement au domaine musical.

Une fois les patterns extraits, il peut être intéressant d'étudier **l'agencement** de leurs occurrences au sein des séquences ; en particulier, nous chercherons à détecter les régularités apparaissant dans l'ordre d'enchaînement des occurrences respectives des différents patterns, problématique ayant un lien certain avec celle de l'**inférence grammaticale.** Mentionnons que l'inférence grammaticale a notamment fait l'objet d'applications en musique [Cruz & Vidal 1997], où les résultats de ce type d'investigations fournissent des informations musicologiques importantes sur le style ou encore sur les techniques de composition ou d'improvisation mises en œuvre.

Les concepts et techniques que nous avons proposés pour la comparaison de séquences musicales pourraient avantageusement être appliqués dans le cadre du **suivi automatique de partitions** (cf. par exemple [Dannenberg 1984 ; Bloch & Dannenberg 1985]). Par ailleurs, nos travaux ouvrent manifestement des possibilités d'application en matière pédagogique (cf. par exemple [Smith, Smaill & Wiggins 1994]). Enfin, une perspective majeure, par laquelle nous terminerons ce chapitre, concerne la fusion d'Imprology avec un système musical interactif, en l'occurrence le système **ImPact** mentionné à diverses reprises dans les chapitres précédents. En premier lieu, la constitution de la *mémoire musicale* de patterns d'ImPact (cf. 3.2.4.3) — actuellement effectuée manuellement — pourra l'être automatiquement grâce à Imprology-FlExPat. Ainsi, en « écoutant » des corpora de lignes de basse et même d'autres instruments[62], le bassiste informatique extraira et mémorisera des passages récurrents, simulant ainsi le processus « d'imitation-assimilation » typique rencontré chez les musiciens de jazz [Sabatella, 1996 ; Coker, 1970]. En second lieu, moyennant la mise en œuvre de mécanismes de prédiction et d'anticipation mentionnés plus haut, ImPact pourrait, dans le cadre d'une performance en groupe (petit orchestre de jazz), anticiper sur le jeu improvisé des autres musiciens. On simulerait ainsi la capacité – reconnue indispensable chez les musiciens humains – de faire en permanence des hypothèses sur ce que vont jouer les autres musiciens pour élaborer à chaque instant un jeu adéquat [Ramalho, Rolland & Ganascia 1998].

---

[62] « L'emprunt » à d'autres, par un musicien de jazz, de patterns et formules, ne se fait pas nécessairement à partir du jeu improvisé de musiciens jouant du même instrument. Ainsi un tromboniste peut « emprunter » des patterns à un saxophoniste, un pianiste à un trompettiste, etc.

BIBLIOGRAPHIE

[Agrawal et al. 1996] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, I. 1996. Fast Discovery of Association Rules. in Fayyad U. M., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press / The MIT Press. ISBN 0-262-56097-6

[Aho & Corasick 1975] Aho, A.V. & Corasick, M. 1975 Efficient String Matching: an aid to bibliographic search. Comm. ACM 18, pp. 333-340.

[Aho, Hopcroft & Ullman 1974] Aho, A.V., Hopcroft, J.E., Ullman, J.D. The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, MA. 1974.

[Aho, Hopcroft & Ullman 1983] Aho, A.V., Hopcroft, J.E., Ullman, J.D. 1983. Data Structures and Algorithms. Addison-Wesley, Reading, MA. 1974. Reading, MA : Addison-Wesley.

[Akoulchina & Ganascia 1997] Akoulchina, I. & Ganascia, J.G. 1997. SATELIT-AGENT: an Adaptive Interface Based on Learning Interface Agents Technology. In Proceedings Sixth International Conference User Modeling (UM'97). Springer Wien New York. pp. 21-32.

[Altschul & Lipman 1989] Altschul, S.F., & Lipman, D.J. 1989. Tree, stars, and multiple biological sequence alignment. SIAM J. Appl. Math., 49:197-209.

[Anagnostopoulou 1997] Anagnostopoulou, C. 1997. Classification in Music: A Computational Model for Paradigmatic Analysis. Proceedings of the 21st

International Computer Music Conference. San Francisco: International Computer Music Association.

[Babel 1991] Babel, L. Finding maximum cliques in arbitrary and in special graphs. Computing 46, pp. 321-341.

[Baggi 1989] Baggi, D. 1989. NeurSwing: An Intelligent Workbench for the Investigation of Swing in Jazz. In D. Baggi (Eds.), *Computer-Generated Music* (pp. 79-93). IEEE Computer Society Press.

[Baker 1980] Baker, D. 1980. *Miles Davis Trumpet.* Giants of Jazz Series. Lebanon: Studio 224 Ed.

[Bakhmutova, Gusev & Titkova 1997] Bakhmutova I., Gusev V., Titkova T. 1997. The Search for Adaptations in Song Melodies. Computer Music Journal 21:1 (Spring 97) p. 58-67.

[Balaban, Ebcioglu & Laske 1992] Balaban M., Ebcioglu, Laske O., eds. 1992. Understanding Music with AI : perspectives on Music Cognition. MIT Press.

[Balas & Yu 1986] Balas, E. & Yu, C.S. 1986. Finding a maximum clique in an arbitrary graph. SIAM Journal on Computing 15(4), pp. 1054-1068.

[Baudoin 1992a] Baudoin, P. 1992. Jazz mode d'emploi. Vol. 1. Editions Outre Mesure.

[Baudoin 1992b] Baudoin, P. 1992. Jazz mode d'emploi. Vol. 2. Editions Outre Mesure.

[Bel & Vecchione 1993] Bel, B. & Vecchione, B. (guest eds.). 1993. Computer and the Humanities 27(1): Special Issue on Computational Musicology.

[Bellman 1957] Bellman, R. 1957. Dynamic Programming. Princeton University Press.

[Berndt & Clifford 1996] Berndt, D. J., and J. Clifford. 1996. Finding Patterns in Time Series: A Dynamic Programming Approach. In Fayyad U. M., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press / The MIT Press. ISBN 0-262-56097-6

[Bloch and Dannenberg 1985] Bloch, J. and Dannenberg, R. 1985. "Real-Time Accompaniment of Polyphonic Keyboard Performance", in Proceedings of the 1985 International Computer Music Conference, pp. 279-290

[Blumer et al. 1983] Blumer, A., Blumer, J. Ehrenfeucht, A., Haussler, D., McConnell, R. 1983. Linear Size Finite Automata for the Set of all Subwords of a Text. Bull. Europ. Assoc. Theoret. Comput. Sci., 21. pp. 12-20.

[Bouchon-Meunier 1993] Bouchon-Meunier, B. 1993. La Logique Floue. Que-sais-je ?, N° 2702. P.U.F.

[Bouchon-Meunier 1995] Bouchon-Meunier, B. 1995. La logique floue et ses applications. Vie artificielle. Addison Wesley.

[Boyer & Moore 1977] Boyer, R.S. & Moore J.S. A Fast String Searching Algorithm. Comm. ACM. 20: 762-772.

[Chan, Wong & Chiu 1992] Chang, S.C., Wong, A.K. & Chiu, D.K. A Survey of Multiple Sequence Comparison Methods. Bull. Math. Biol., 54. pp. 563-598.

[Chemiller 1994] Chemiller, M. 1994. Analysis and Computer Reconstruction of a Musical Fragment of György Ligeti's *Melodien*. Symposium Music and Mathematics, Bucarest, 1994.

[Clift et al. 1986] Clift, B., Haussler, D., McConnell, R., Schneider, T.D., Stormo, G.D. 1986. Sequence Landscapes. Nucleic Acids Research 14. pp. 141-158.

[Coker 1964] Coker, Jerry. 1964. Improvising Jazz. Englewood Cliffs, NJ: Prentice-Hall.

[Coker 1970] Coker, Jerry. 1970. Patterns for Jazz. Studio publications/recordings. Lebanon, IN.

[Cope 1990] Cope, D. 1990 ICMC 1990 Pattern Matching Code. Online document available at http://arts.ucsc.edu/faculty/cope/glasgow.html.

[Cope 1991a] Cope, D. 1991. *Computers and Musical Style.* Oxford: Oxford University Press.

[Cope 1991b] Cope, D. 1991. *Recombinant Music.* Computer, July 1991

[Cope 1992] Cope, D. 1992. A Computer Model of Music Composition." In Machine Models of Music, Stephan Schwanauer and David Levitt, eds. Cambridge, Mass.: MIT Press. pp. 403-425

[Coüasnon & Retif 1998] Coüasnon, B. & Rétif, B. 1998. Utilisation d'une grammaire dans la reconnaissance de partitions d'orchestre. In Pachet, F. & Chemiller, M. (coordinateurs). Recherches et applications en informatique musicale. Hermès.

[Crochemore 1985] Crochemore, M. 1985. Optimal factor transducers. In Apostolico, A. & Galil, Z., eds., Combinatorial Algorithms on Words, volume 12 - F. pp. 31-43. NATO Advanced Science Institutes.

[Crochemore 1986] Crochemore, M. 1986. Transducers and repetitions. Theoretical computer science, 45. 63-86.

[Crochemore et al. 1992] Crochemore, M., Czumaj, A. Gasieniec, L., Jarominek, S., Lecroq, T., Plandowski, Rytter, W. 1992. Speeding Up Two String Matching Algorithms. In Finkel, A., & Jantzen, M., eds., 9th Annual Symposium on Theoretical Aspects of Computer Science, pp. 589-600. Berlin : Springer-Verlag.

[Crochemore et al. 1993] Crochemore, M., Czumaj, A. Gasieniec, L., Jarominek, S., Lecroq, T., Plandowski, Rytter, W. 1993. Fast Multi-Pattern Matching. Rapport I.G.M. 93-3. Univ. Marne-la-Vallée.

[Crochemore & Perrin 1991] Crochemore, M. & Perrin, D. 1991. Two-way string matching. Journal of the ACM 38, 3. pp. 651-675

[Cruz & Vidal 1997] Cruz, P., and E. Vidal. 1997. A study of Grammatical Inference Algorithms in Automatic Music Composition and Musical Styles Recognition. Workshop on Automata Induction, Grammatical Inference, and Language Acquisition. Fourteenth International Conference on Machine Learning (ICML-97), July 12, 1997, Nashville, Tennessee.

[Crochemore & Rytter 1994] Crochemore, M. & Rytter, W. 1994. Text Algorithms. Oxford University Press.

[Dannenberg 1984] Dannenberg, R. 1984. An on-line algorithm for real-time accompaniment. Procceedings of the 1984 International Computer Music Conference (ICMC'84).

[Dayhoff, Barker & Hunt 1983] Dayhoff, M.O., Barker, W.C., and Hunt, L.T. 1983. Establishing Holomogies in Protein Sequences. Methods in Enzymology 91, p. 524-544. Academic Press.

[Dayhoff, Schwartz & Orcutt 1978] Dayhoff, M.O., Schwartz, R.M. & Orcutt, B.C. 1978. A Model of Evolutionary Change in Proteins. In Atlas of Protein Sequence and Structure. National Biomed. Res. Found.

[Diday 1980] Diday, E. 1980. Introduction à la méthode des nuées dynamiques. in Analyse des données, vol. 1. A.P.M.E.P. pp. 121-132.

[Diday et al. 1982] Diday, E., Lemaire, J., Pouget, J., Testu, F. 1982. Eléments d'analyse de données. Dunod.

[Dowling 1994] Dowling, W.J. 1994. Melodic contour in Hearing and Remembering Melodies. in Musical Perceptions, Oxford University Press. New York.

[Dowling & Fujitani 1971] Dowling, W.J. & Fujitani, D. 1971. Contour, Interval and Pitch Recognition in Memory for Melodies. Psychological Review, 85(4), PP. 341-354.

[Drake 1990]. Drake, C. 1990. Processus cognitifs impliqués dans l'organisation du rythme musical. Thèse de Doctorat, Université Paris V.

[Edwards et al. 1996] Edwards, P., Bayer, D., Green, C.L., Payne, T.R. 1996. Experience with Learning Agents which Manage Internet-Based Information. Proceedings AAAI Spring Symposium on Machine Learning in Information Access. pp. 31-40.

[Erickson & Sellers 1983] Erickson, B. & Sellers, P. 1983. Recognition of Patterns in Genetic Sequences. In Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley.

[Favory 1995] Favory, J. 1995. Les Unités Sémiotiques Temporelles : simulation et production par ordinateur. Journées d'Informatique Musicales (JIM'95), Paris.

[Fayyad, Piatetsky-Shapiro & Smyth 1996a] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. 1996. From Data Mining to Knowledge Discovery in Databases. AI Magazine 17(3), Fall 1996.

[Fayyad, Piatetsky-Shapiro & Smyth 1996b] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. 1996. From Data Mining to Knowledge Discovery : an Overview. in Fayyad U. M., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. 1996. Advances in

Knowledge Discovery and Data Mining. AAAI Press / The MIT Press. ISBN 0-262-56097-6

[Fayyad, Haussler & Storloz 1996] Fayyad, U.M., Haussler, D. & Stolorz, Z. 1996. KDD for Science Data Analysis: Issues and Examples. in Proceedings Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 50-56. Menlo Park : AAAI.

[Flann & Dietterich 1986] Flann, N. & Dietterich, T. 1986. Selecting Appropriate Representations for Learning from Examples. In Proceedings Fifth National Conference on Artificial Intelligence (AAAI-86).

[Frawley, Piatetsky-Shapiro & Matheus 1991] Frawley, W.J., Piatetsky-Shapiro, G. and Matheus, J. 1991. Knowledge Discovery in Databases: an Overview. in [Piatetsky-Shapiro & Frawley 1991], p. 1-27.

[Fry 1991] Fry, C. 1991. Flavors Band: A Language for Specifying Musical Style. In S. Pope (Eds.), *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology* (pp. 49-63). Massachusetts: MIT Press.

[Fujinaga et al. 1989] Fujinaga, I., Alphonce, B., Pennycook, B. & Boisvert, N. 1989. Optical Recognition of Music Notation by Computer. Computers in Music Research 1, Fall 1989.

[Galil 1979] Galil, Z. 1979. On improving the worst-case running time of the boyer-moore string-searching algorithm. Comm. ACM 22, pp. 505-508.

[Galil 1994] Galil, Z. 1994. Préface à l'ouvrage de Crochemore, M. & Rytter, W. 1994. Text Algorithms. Oxford University Press.

[Galil & Seiferas 1983] Galil, Z. & Seiferas, J. 1983. Time-Space Optimal String Matching. J. Comput. Syst. Sci. 26, pp. 280-294.

[Ganascia 1990] Ganascia, J.-G. 1990. L’âme-machine: les enjeux de l’intelligence artificielle. Science Ouverte. Paris: Seuil.

[Ganascia 1993] Ganascia, J.-G. 1993. L’intelligence artificielle. Dominos. Paris: Flammarion.

[Ganascia 1995] Ganascia, J.-G. 1995. Les sciences cognitives. Dominos. Paris: Flammarion.

[Ganascia, Ramalho & Rolland 1995] Ganascia, J.G., Ramalho, G. and Rolland, P.Y., 1995. An Artificially Intelligent Jazz Performer. In Proceedings of InterSymp'95: 1995 International Symposium on Cybernetic Paradigms of Musical and Theatrical Performance. Baden-Baden, Germany, Août 1995. (Voir aussi [Ramalho, Rolland & Ganascia 1998])

[Garey & Johnson 1979] Garey, M.R. & Johnson, D.S. 1979. Computers and intractability : a guide to the theory of NP-completeness. New York : W.H. Freeman & Co.

[Gerhards & Lindenberg 1979] Gerhards, L. & Lindenberg, W. 1979. Clique detection for non-directed graphs : two new algorithms.

[Goldberg & Robson 1983] Goldberg, A., & Robson, D. 1983. *Smalltalk-80: The Language and its Implementation.* London: Addison-Wesley.

[Gondran & Minoux 1979] Gondran, M. & Minoux, M. 1979. Graphes et algorithmes. Paris: Eyrolles.

[Gridley 1985] Gridley, M., 1985. Jazz styles. Englewood Cliffs: Prentice-Hall.

[Gusfield 1993] Gusfield, D. 1993. Efficient Methods for Multiple Sequence Alignment with Guaranteed Error Bounds. Bull. Math. Biol., 55:141-154.

[Hashimoto, Qi & Chang 1996] Hashimoto, S., Qi, H., & Chang D., 1996. Sound Database Retrieved by Sound. Poster at 20th International Computer Music Conference (ICMC'96, Honk Kong, Août 1996).

[Henikoff & Henikoff 1992] Henikoff, S. & Henikoff, J.G. 1992. Amino acid substitution matrices for protein blocks. Proceedings National Academy of Sciences of the USA 89, pp. 10915-10919.

[Hirata 1997] Hirata, K. (Ed.) 1997. Proceedings of Workshop on Artificial Intelligence and Music, 15th International Joint Conference on Artificial Intelligence (IJCAI’97), Nagoya, Japan.

[Hodgson 1996] Hodgson, P. 1996. Modeling Cognition in Creative Musical Improvisation. Poster, 4th International Conference on Music Perception and Cognition (ICMPC’96), Montreal, Août 1996.

[Hörnel & Ragg 1996] Hörnel D., R. Thomas. 1996. Learning Musical Structure and Style by Recognition, Prediction and Evolution. Poster at 20th International

Computer Music Conference (ICMC '96). San Francisco: Intenational Computer Music Association.

[Horowitz 1995a] Horowitz, D. 1995. Representing Musical Knowledge: Processing Melodic Lines in a Jazz Improvization System. International Conference on Music and Artificial Intelligence (ICMAI'95)

[Horowitz 1995b] Horowitz, D. 1995. Representing Musical Knowledge in a Jazz Improvization System. In G. Widmer (Ed.), Proceedings 4th Workshop on Artificial Intelligence and Music, 14th International Joint Conference on Artificial Intelligence (IJCAI'95) Montreal, Canada, 21-25 Août, 1995.

[Hui 1992] Hui, L.C.K. 1992. Color Set Size Problem with Applications to String Matching. In Apostolico, A., Crochemore, M., Galil, Z. & Manber, U. (eds.) Combinatorial Pattern Matching. Lecture Notes in Computer Science. Springer Verlag. pp. 230-243.

[Huron 1994] Huron, D. (1994) *Humdrum Toolkit Manual*. Menlo Park: Center for Computer Assisted Research in the Humanities.

[Karlin et al. 1988] Karlin, S., Morris, M., Gyandour, G. & Leung, M.Y. 1988. Efficient algorithms for molecular sequence analysis. Proceedings National Academy of Sciences of the USA 85. pp. 841-845.

[Karp, Miller & Rosenberg 1972] Karp, R.M., Miller, R.E. & Rosenberg, A.L. 1972. Rapid Identification of Repeated Patterns in Strings, Arrays and Trees. Proc. 4th ACM Symposium on Theory of Computing. ACM, New York. 125-136.

[Kernfeld 1983] Kernfeld, B. (1983) Two Coltranes. *Annual Review of Jazz Studies*, 2, 7-66.

[Kernfeld 1989] Kernfeld, B. (ed.) 1989. Improvisation. In Kernfeld, B. (ed.) 1989. *The New Grove Dictionary of Jazz.* Grove, USA. pp. 554-562.

[Knuth, Morris & Pratt 1977] Knuth, D.E., Morris Jr., J.H. & Pratt, V.R. Fast Pattern Matching in Strings. SIAM J. Comput. 6. 323-350.

[Kolodner 1993] Kolodner, J. 1993. Case Based Reasoning. Morgan-Koffman Pub.

[Korth & Silberschatz 1991] Korth, H.F., Silberschatz, A. 1991. Database system concepts. Mac Graw-Hill, Computer Science Series

[Krumhansl 1990] Krumhansl, C.L. 1990. Cognitive foundations of musical pitch. Oxford: Oxford University Press

[Kruskal 1983] Kruskal, J.B. 1983. An Overview of Sequence Comparison. in Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley. p. 1-44.

[Kruskal & Liberman 1983] Kruskal, J. & Liberman, M. The Symmetric Time-Warping Problem: from Continuous to Discrete. in Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley.

[Kruskal & Sankoff 1983] Kruskal, J. & Sankoff, D. 1983. An Anthology of Algorithms and Concepts for Sequence Comparison. 293-296. In Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley.

[Landraud, Avril & Chretienne 1989] Landraud A.M., J.F. Avril and P. Chretienne. 1989. An Algorithm for Finding a Common Structure Shared by a Family of Strings. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(8): 890-895.

[Laquer 1981] Laquer, H.T. 1981. Asymptotic limits for a two-dimensional recursion. Stud. Appl. Math., 64, 271-277.

[Lawrence et al. 1993] Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F. & Wooton, J.C. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262, pp. 208-214.

[Lawrence & Reilly 1990] Lawrence, C.E. & Reilly, A.A. 1990. An expected maximisation (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. Proteins: structure, function and genetics 7. pp. 41-51

[Lenat 1990] Lenat, D. 1990. L'intelligence artificielle. Pour la Science, 217. pp. 54-57.

[Lenat & Feigenbaum 1991] Lenat, D., & Feigenbaum, E. (1991). On the Thresholds of Knowledge. Artificial Intelligence, 47(1-3), 185-250.

[Lerdahl & Jackendoff 1983] Lerdahl, F. & Jackendoff, R. 1983. A Generative Theory of Tonal Music. MIT Press.

[Leung et al. 1991] Leung, M.Y., Blaisdell, B.E., Burge, C. & Karlin, S. 1991. An efficient algorithm for identifying matches with errors in multiple long molecular sequences. Journal of Molecular Biology 221. pp. 1367-1378.

[Levenshtein 1965] Levenshtein, V.I. 1965. Binary codes capable of correcting deletions, insertions and reversals. Cybernetics and Control Theory 10(8): 707-710.

[Longuet-Higgins & Lee 1982] Longuet-Higgins, H.C., Lee, C.S. 1982. The perception of musical rythms. Perception 11, 115-156.

[Maglio & Barett 1997] Maglio, P.P., Barett, R. 1997. How to Build Modeling Agents to Support Web Searchers. Proceedings International Conference User Modeling (UM'97). Springer Wien New York. pp. 5-16.

[Malson 1993] Malson, L. 1993. Les maîtres du jazz—d'Oliver à Coltrane. Que sais-je, PUF. 10e édition mise à jour.

[Mannila, Toivonen & Verkamo 1997] Mannila, H., Toivonen, H. and Verkamo, A.I. 1997. Discovery of Frequent Episodes in Event Sequences. Report C-1997-15, Department of Computer Science, University of Helsinki, Finland.

[Martinez 1983] Martinez, H.M. 1983. An efficient method for finding repearts in molecular sequences. Nucleic Acid Research, 11. pp. 4629-4634.

[Martinez 1988] Martinez, H.M. 1988. A flexible multiple sequence alignment program. Nucleic Acid Research, 16. pp. 1683-1691.

[Masek & Paterson 1983] Masek, W.J., & Paterson, M.S. 1983. How to Compute String-Edit Distances Quickly. In Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley. pp. 337-349.

[Massaro, Kallman & Kelly 1980] Massaro, D.W., Kallman, H.J., Kelly, J.L. 1980. The Role of Tone Height, Melodic Contour, and Tone Chroma in Melody Recognition. Journal of Experimental Psychology: Human Learning and Memory, 6(1) 77-90

[McAdams & Bigand 1993] McAdams, S., Bigand, E. Thinking in Sound : the Cognitive Psychology of Human Audition. Trad. fr. : penser les sons : la psychologie cognitive de l'audition humaine, PUF, Paris

[McCreight 1976] McCreight, E.M. 1976. A space-economical suffix tree construction algorithm. Journal of the ACM, 23. pp. 262-272.

[Mehegan] Mehegan, J. 1962. Jazz improvization. Vol. 1: Rythmic and Tonal Principles. Amsco Pub. Co; New York.

[Meyer 1956] Meyer, L.B. 1956. Emotion and meaning in music. Chicago: University of Chicago Press.

[Meyer 1973] Meyer, L.B. 1973. Explaining Music. Berkeley: University of California Press.

[Meyer, Ganascia & Zucker 1997] Meyer C., Ganascia J.-G., Zucker J.-D. 1997. Learning Strategies in Games by Anticipation. Proceedings of IJCAI, Morgan Kaufman, Japan, August, 1997

[Milgram 1993] Milgram, M. 1993. Reconnaissance des formes: méthodes numériques et connexionistes. Armand Collin

[Minasi 1991] Minasi, M. Recognizing Patterns. AI Expert, February 1991. pp. 15-17.

[Minsky 1986] Minsky, M. 1986. The Society of Mind. London: Heinemann.

[Mongeau & Sankoff 1990] Mongeau, M. and Sankoff, D. 1990. Comparison of musical sequences. Computer and the humanities, 24. 161-175

[Moog 1986] Moog, B. 1986. MIDI: Musical Instrument Digital Interface. *Journal of the Audio Engineering Society*, 34(5), 394-404.

[Morris & Pratt 1970] Morris Jr., J.H. & Pratt, V.R. 1970. A Linear Pattern-Matching Algorithm. Report #40, Univ. California at Berkeley.

[Musitek 1995] Musitek Music Recognition Technologies, Ojai, CA. MIDISCAN for Windows Manual.

[Myers & Miller 1988] Myers, E.W., & Miller, W. 1988. Optimal Alignment in Linear Space. CABIOS 4, pp. 11-17.

[Narmour 1989] Narmour, E. 1989. The analysis and cognition of basic melodic structures: the implication-realization model. University of Chicago Press.

[Narmour 1992] Narmour, E. 1992. The analysis and cognition of melodic complexity. U. of Chicago Press.

[Needleman & Wunsch 1970] Needleman, S., Wunsch, C.D. 1970. A general Method Applicable to the Search for Similarities in the Amino-Acid Sequence of Two Proteins. J. Mol. Bio. 48:443-453

[Osherson & Smith 1981] Osherson, D.N. & Smith, E.E. 1981. On The Adequacy of Prototype Theory as a Theory of Concepts. Cognition, 9. pp. 35-58.

[Owens 1974] Owens, T. 1974. Charlie Parker: Techniques of Improvisation. Ph.D. Thesis, Dept. of Music, University of California at Los Angeles (UCLA), 2 vols., 900 pages.

[Owens 1996] Owens, T. Communication personnelle, mai 1996.

[Pachet 1991a] Pachet, F. 1991. Representing Knowledge Used by Jazz Musicians. In Alphonce, B. and B. Pennycook (eds.), Proceedings of the 1991 International Computer Music Conference. San Francisco: Computer Music Association.

[Pachet 1991b] Pachet, F. 1991. A meta-level architecture for analysing jazz chord sequences. In Alphonce, B. and B. Pennycook (eds.), Proceedings of the 1991 International Computer Music Conference, pp. 266- 269. San Francisco: Computer Music Association.

[Pachet 1992] Pachet, F. 1992. Representation de connaissances par objets et règles: le système NéOpus. Thèse de doctorat, Université Paris VI.

[Pachet 1994] Pachet, F. 1994. The MusES system: an environment for experimenting with knowledge representation techniques in tonal harmony. In *First Brazilian Symposium on Computer Music - SBC&M '94*, (pp. 195-201). Caxambu: Computer Science Brazilian Society (SBC).

[Pachet 1998] Pachet, F. 1998. Computer Analysis : a Synthesis. Contemporary Music Review. Special Issue on Music and Artificial Intelligence (Miranda, E., ed.). (à paraître).

[Pachet & Assayag 1995] Pachet, F., & Assayag, G. 1995. Dossier IA et Musique. Bulletin de l'AFIA, 23 (octobre), pp. 30-48.

[Pachet & Chemiller 1998] Pachet, F. & Chemiller, M. (coordinateurs). 1998. Recherches et applications en informatique musicale. Ouvrage collectif basé sur une sélection de communications aux Journées d'Informatique Musicale (JIM) 1995, 1996 et 1997. Paris : Hermès.

[Pachet, Ramalho & Carrive 1996] Pachet, F., Ramalho, G., & Carrive, J. (1996). Representing Temporal Musical Objects and Reasoning in the MusES System. *Journal of New Music Research*, 3.

[Pennycook 1990] Pennycook, Bruce. 1990. "Towards Advanced Optical Music Recognition". Advanced Imaging, April 1990.

[Pennycook 1995] Pennycook, Bruce. 1995. Music Library of the Future: A Pilot Project. In *Proceedings of the 1995 International Computer Music Conference*, pp. 195-198. San Francisco: International Computer Music Association.

[Pennycook & Costa-Giomi 1996] Pennycook, B., & Costa-Giomi, E. (Chairs). 1996. Proceedings of the 4th International Conference on Music Perception and Cognition (ICMPC'96), Montreal. (Proceedings also available on CD-ROM)

[Pennycook & Simoni 1995] Pennycook, B. & Simoni, M. (eds.). 1995. ICMA/ICMC Proceedings Index Ver. II (1987-95). Designed, compiled and edited by Bruce Pennycook and Mary Simoni. International Computer Music Association. Egalement disponibles sur le world wide web à l’URL : http://music.dartmouth.edu/~icma/ProceedingsIndex.html

[Pennycook, Stammen & Reynolds 1993] Pennycook, B., D.R. Stammen, & D. Reynolds (1993) Toward a computer model of a jazz improviser. In *Proceedings of the 1993 International Computer Music Conference*, pp. 228-231. San Francisco: International Computer Music Association.

[Piatetsky-Shapiro & Frawley 1991] Piatetsky-Shapiro, G.Frawley, W. 1991. Knowledge discovery in databases. Menlo Park, Calif. : Cambridge, Mass. : AAAI Press ; MIT Press. 525 p.

[Povel & Essens 1985] Povel, D.J. & Essens, P. 1985. Perception of temporal patterns. Music perception, 2 (4), 411-440

[Pressing 1988] Pressing, J. 1988. Improvisation: methods and models. in Sloboda, J.A. 1988 (ed.). Generative processes in music: the psychology of performance, improvisation and composition. Oxford: Oxford Sciences Publications.

[Ramachandran, Ramakrishnan & Sesisakharan 1963] Ramachandran, G.N., Ramakrishnan, C. & Sesisakharan, V. 1963. Stereochemistry of polypeptide chain configurations. J. Mol. Biol. 7, pp. 95-99.

[Ramalho 1997a] Ramalho, G. 1997. Construction d'un agent rationnel jouant du jazz. Thèse de doctorat de l'Université Pierre et Marie Curie (Paris VI).

[Ramalho 1997b] Ramalho, G. Pattern Reuse in Tonal Music Improvisation and Accompaniment Systems. Fourth Brasilian Symposium on Computer Music - SBC&M '97, Brasilia. pp 47-58

[Ramalho & Ganascia 1994] Ramalho, G., & Ganascia, J.-G. 1994. Simulating Creativity in Jazz Performance. In *Twelfth National Conference on Artificial Intelligence*, (pp. 108-13). Seattle: The AAAI Press.

[Ramalho & Pachet 1994] Ramalho, G., & Pachet, F. 1994. What is Needed to Bridge the Gap Between Real Book and Real Jazz Performance? In Fourth International Conference on Music Perception and Cognition, (pp. 349-50). Liège: Escom.

[Ramalho, Rolland & Ganascia 1998] Ramalho, G., Rolland, P.Y., and Ganascia, J.G., 1998. An Artificially Intelligent Jazz Performer. Journal of New Music Research. (à paraître; article basé sur [Ganascia, Ramalho & Rolland 1995])

[Rifqi 1995] Rifqi, M. 1995. Le raisonnement à partir de cas dans un environnement incertain et imprécis : à la recherche d'une formalisation des mesures de similarité. 4e séminaire français sur le raisonnement à partir de cas. AFCET, Paris.

[Rifqi 1996] Rifqi, M. 1996. Mesures de comparaison, typicalité et classification d'objets flous : théorie et pratique. Thèse de doctorat d'Informatique de l'Université Paris 6.

[Rolland 1998] Rolland, P.Y. 1998. FlExPat : a Novel Algorithm for Musical Pattern Discovery. 12th Colloquium on Musical Informatics (XII CIM). Gorizia, Italy.

[Rolland & Ganascia 1996a] Rolland, P.Y., Ganascia, J.G. 1996a. Automated Motive Oriented Analysis of Musical Corpuses: a Jazz Case Study. In Proceedings of the 20th International Computer Music Conference (ICMC'96, Honk Kong, Août 1996) pp. 240-243.

[Rolland & Ganascia 1996b] Rolland, P.Y., Ganascia, J.G. Automated Identification of Prominent Motives in Jazz Solo Corpuses. In Pennycook, B., & Costa-Giomi, E. (Chairs.). Proceedings of the 4th International Conference on Music Perception and Cognition (ICMPC'96, Montreal, Août 1996) pp. 491-495.

[Rolland & Ganascia 1998a] Rolland, P.Y., Ganascia, J.G. 1998. Musical Pattern Extraction and Similarity Assessment. Contemporary Music Review. Special Issue on Music and Artificial Intelligence (Miranda, E., ed.). (à paraître).

[Rolland & Pachet 1996] Rolland, P.Y., F. Pachet, A Framework for Representing Knowledge about Synthesizer Programming. Computer Music Journal 20:3, Fall 1996. Cambridge, Massachusetts: MIT Press.

[Rolland & Pachet 1998] Rolland, P.Y. and Pachet, F. 1998. Représentation de connaissances sur la programmation de synthétiseurs. In Pachet, F. & Chemiller, M. (coordinateurs). Recherches et applications en informatique musicale. Hermès. *Basé sur une communication de même titre aux Secondes Journées d'Informatique Musicale (JIM'95), Paris, France, Avril 1995.*

[Rosch 1978] Rosch, E. 1978. Principles of Categorization. in Rosch, E. & Lloyd, B.B., eds., Cognition and Categorization, Laurence Erlbaum Associates. pp. 27-48.

[Rowe 1993] Rowe, R. 1993. Interactive Music Systems. 1993. MIT Press.

[Rowe 1998] Rowe, R. 1998. (Titre à préciser). Contemporary Music Review. Special Issue on Music and Artificial Intelligence. (Miranda, E., ed.) (à paraître)

[Ruwet 1972a] Ruwet, N. 1972. Note sur les duplications dans l'oeuvre de Claude Debussy. In Ruwet, N. 1972. Langage, musique, poésie. Editions Le Seuil.

[Ruwet 1972b] Ruwet, N. 1972. Quelques remarques sur le rôle de la répétition dans la syntaxe musicale. In Ruwet, N. 1972. Langage, musique poésie. Editions Le Seuil.

[Sabatella 1996] Sabatella, M. 1996. A Jazz Improvisation Primer. Document disponible sur le web à l'URL suivante : http://www.fornet.org/~marc/primer.

[Sagot 1996] Sagot, M.F. 1996. Ressemblance lexicale et structurale entre macromolécules–formalisation et approches combinatoires. Thèse, Université de Marne-La-Vallée.

[Sagot et al. 1995a] Sagot, M.F., Escalier, V.; Viari, A., Soldano, H. 1995. Searching for repeated words in a text allowing for mismatches and gaps. Vinas del Mar, Chili. Second South American Workshop on String Processing, pp. 87-100.

[Sagot et al. 1995a] Sagot, M.F., Viari, A., Pothier, J., Soldano, H. 1995. Finding flexible patterns in a text–an application to 3D molecular matching. CABIOS 11, pp. 59-70.

[Sagot & Viari 1996] Sagot, M.F., Viari, A. 1996. A double combinatorial approach to discovering patterns in biological sequences. In Combinatorial Pattern Matching, Springer Verlag, LNCS 1075, 168-208.

[Sagot, Viari & Soldano 1995] Sagot, M.F., Viari, A., Soldano, H. 1995. Multiple comparison: a peptide matching approach. In Galil, Z. & Ukkonen, E. (eds.), Combinatorial Pattern Matching, Springer Verlag, LNCS 937, 366-385.

[Sakoe & Chiba 1990] Sakoe, H. & Chiba, S. 1990. Dynamic Programming Optimization for Spoken Word Recognition. in Waibel, A. & Lee, K., eds. Readings in Speech Recognition. Morgan Kaufman Pub. pp. 159-165.

[Sankoff 1972] Sankoff, D. 1972. Matching Sequences under deletion-insertion constraints. Proc. Natl. Acad. Sci. USA, 68, 4-6.

[Sankoff & Kruskal 1983] Sankoff, D. & Kruskal, J. (eds.). 1983. Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison. Addison Wesley.

[Santibanez & Rohde 1987] Santibanez, M. & Rohde, K. 1987. A multiple alignement program for protein sequences. Computer Applications in the Biosciences, 3, pp. 111-114.

[Saporta 1990] Saporta, G. 1990. Probabilités, analyse des données et statistiques. Technip.

[Saxena 1991] Saxena, S. 1991. Evaluating Alternative Instance Representations. In Proceedings of the European Working Session on Learning. pp. 465-467. Springer-Verlag.

[Schenker 1954] Schenker, H. 1954. Harmony. Translated by Elisabeth Mann Borgese. Chicago: University of Chicago Press.

[Schenker 1956] Schenker, H. 1956. Der Freie Satz. Vienna: Universal.

[Schuler, Altschul & Lipman 1991] Schuler, G.D., Altschul, S.F. & Lipman, D.J. 1991. A workbench for multiple alignment construction and analysis. Proteins, 9, pp. 180-190.

[Sellers 1974] Sellers, P. 1974. On the theory and computation of evolutionary distances. SIAM J. Appl. Math., 26, 787-793.

[Sellers 1980] Sellers, P.H. 1980. The theory and Computation of Evolutionary distance: Pattern Recognition. Journal of Algorithms 1: 359-373.

[Shapiro 1992] Shapiro, S. C. (editor in chief). 1992. Encyclopedia of Artificial Intelligence. Second edition. 2 Volumes. Wiley-Interscience.

[Simon & Sumner 1968] Simon, H and Sumner, R.K. 1968. Patterns in Music. In Formal Representations of Human Judgment. New York, John Wiley Sons.

[Smaill, Wiggins & Harris 1993] Smaill, A., Wiggins, G. & Harris, M. (1993) Hierarchical Music Representation for Composition and Analysis. (1993). In *Computational Musicology,* edited by Bel, B. Vecchione, B. Special Issue of Computer and the Humanities 27(1). North Holland Pub. Co.

[SMF 1988] Standard MIDI Files specification 0.06/1.0. (Based on D. Oppenheim's spec.). Los Angeles, CA: International Midi Association. 1988.

[Sloboda 1985] Sloboda, J.A. 1985 (ed.). The Musical Mind—The Cognitive Psychology of Music, Oxford Science Publications. Oxford: Clarendon Press.

[Sloboda 1988] Sloboda, J.A. 1988 (ed.). Generative processes in music: the psychology of performance, improvisation and composition. Oxford: Oxford Sciences Publications.

[Smith, Smaill & Wiggins 1994] Smith, A. Smaill, & G. Wiggins (Eds.). 1994. Music Education: an Artificial Intelligence Perspective. London: Springer-Verlag.

[Smith 1983] Smith, G. (1983) Homer, Gregory, and Bill Evans? The Theory of Formulaic Composition in the Context of Jazz Piano Improvisation. Ph.D. Thesis, Harvard University.

[Smith & Waterman 1981] Smith, T.F. & Waterman M.S. 1981. Identification of Common Molecular Subsequences. Journal of Molecular Biology. 147, 195-97.

[Soldano, Viari & Champesme 1995] Soldano, H., Viari, A., Champesme, M. Searching for flexible repeated patterns using a non transitive similarity relation. Pattern Recognition Letters, 16:233-246, 1995.

[Spector & Alpern 1995] Spector, L., & Alpern, A. (1995). Induction and Recapitulation of Deep Musical Structure. In G. Widmer (Ed.), *Fourth Workshop on Artificial Intelligence and Music, IJCAI-95*, (pp. 41-8). Montreal:

[Spouge 1989] Spouge, J.L. 1989. Speeding up Dynamic Programming Algorithms for Finding Optimal Lattice Graphs. SIAM J. Appl. Math., 5. pp. 1552-1566.

[Stammen & Pennycook 1993] Stammen, D. and B. Pennycook. (1993) Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the 1985 International Computer Music Conference*. San Francisco: International Computer Music Association.

[Stammen & Pennycook 1994] Stammen, D. and B. Pennycook. 1994 Real time segmentation of music using an adaptation of Lerdahl and Jackendoff's grouping principles. ICMPC 94.

[Stech 1981] Stech, D. A. 1981. A computer assisted approach to micro analysis of melodic lines. Computer and the humanities 15:211-221. North Holland Pub. Co.

[Suppes et al. 1990] Suppes, P., Krantz, D.H., Luce, R.D., & Tversky, A. 1990. Foundations of Measurement, Vol. 2. New York : Academic Press.

[Todd & Loy 1991] Todd, P.M. and D.G. Loy (eds.). 1991. Music and Connexionism. Cambridge: MIT Press.

[Tversky 1977] Tversky, A. 1977. Features of Similarity. Psychological Review, 84. pp. 327-352.

[Ukkonen 1985] Ukkonen, E. 1985. Algorithms for Approximate String Matching. Information and Control 64: 100-118.

[Ukkonen 1992] Ukkonen, E. 1992. Constructing suffix trees on-line in linear time. IFIP'92, pp. 484-492.

[Ullman 1988] Ullman, J.D. Principles of Database and Knowledge-Base Systems - Volume 1: classical Database systems - Computer Science Press, 1988.

[Ulrich 1977] Ulrich, W. (1977). The Analysis and Synthesis of Jazz by Computer. In *Fifth International Joint Conference on Artificial Intelligence*, (pp. 865-72).

[Vishkin 1985] Vishkin, U. 1985. Optimal Parallel Pattern Matching in Strings. Information and Control, 67. pp. 91-113.

[Vishkin 1991] Vishkin, U. 1991. Deterministic Sampling — a New Technique for Fast Pattern Matching. Siam J. Comput. 20:1 pp. 22-40.

[Wagner & Fisher 1974] Wagner, R.A., and Fischer, M.J. 1974. The string to string correction problem. Journal of the ACM 21(1):168-173.

[Wang et al. 1994] Wang, J., Chirn, G.-W., Marr, T.G. Shapiro, B. Shasha, D., Zhang, K. 1994. Combinatorial Pattern Discovery for Scientific Data: Some Preliminary Results. Proceedings of ACM SIGMOD '94.

[Waterman 1995] Waterman, M. S. 1995. Introduction to Computational Biology. Maps, sequences and genomes. London: Chapman & Hall.

[Waterman & Eggert; 1987] Waterman, M.S. & M. Eggert; 1987. A New Algorithm for Best Subsequence Alignments with Applications to tRNA-rRNA Comparisons. J. Mol. Biol. 197:723-728.

[White 1978] White, A. (ed.) 1978. 308 Transcribed Alto Saxophone and Tenor Saxophone solos. Washington.

[Widmer 1992] Widmer, G. 1992. Qualitative perception modeling and intelligent musical learning. Computer Music Journal 16(2).

[Widmer 1993] Widmer, G. 1993. Understanding and learning musical expression. International Computer Music Conference 1993. International Computer Music Association.

[Widmer 1994] Widmer, G. 1994. The Synergy of Music Theory and AI: Learning Multi-Level Expressive Interpretation. Extended version of a paper appearing in Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-94), Seattle, WA, USA.

[Widmer 1995] Widmer, G. (Ed.), 1995. Proceedings of Workshop on Artificial Intelligence and Music, 14th International Joint Conference on Artificial Intelligence (IJCAI’95) Montreal, Canada, 21-25 Août, 1995.

[Won Yi 1998] Won Yi, Suk (Chair). 1998. Proceedings of the 5th International Conference on Music Perception and Cognition (ICMPC'98, Séoul)

[Zaragoza & Gallinari 1998] Zaragoza H., Gallinari P. (1998), Modèle Hiérarchique de Recherche et d'Extraction de l'Information Textuelle de Surface. Journées Francophones d'Apprentissage (JFA'98). (à paraître)

[Zucker 1996] Zucker, J.D. 1996. Appariements et changements de représentation pour l’apprentissage symbolique. Thèse de doctorat de l’Université Paris VI.

[Zwick, Carlstein & Budescu 1987] Zwick, R., Carlstein, E., Budescu, D.V. 1987. Measures of Similarity among Fuzzy Concepts: a Comparative Analysis. International Journal of Approximate Reasoning, 1:2. pp. 221-242.

ANNEXE A. PATTERNS D'OWENS [1974]

Une partie du lexique de patterns (« motives ») extrait du jeu improvisé de C. Parker par T. Owens, est reproduite ici, avec la permission de l'auteur, sous la forme manuscrite originale [Owens 1974, pp. 1, 5 et 10]. Les nombres approximatifs d'occurrences sont indiqués pour les différentes patterns ou groupes de patterns. Certains des identificateurs employés dans le présent mémoire ne figurent pas sur les pages reproduites ici : nous avons en effet précisé certains identificateurs d'Owens qui ne permettaient pas de distinguer entre plusieurs patterns. Par exemple, nous avons précisé 1Aa en 1Aau, 1Aav, et 1Aaw.

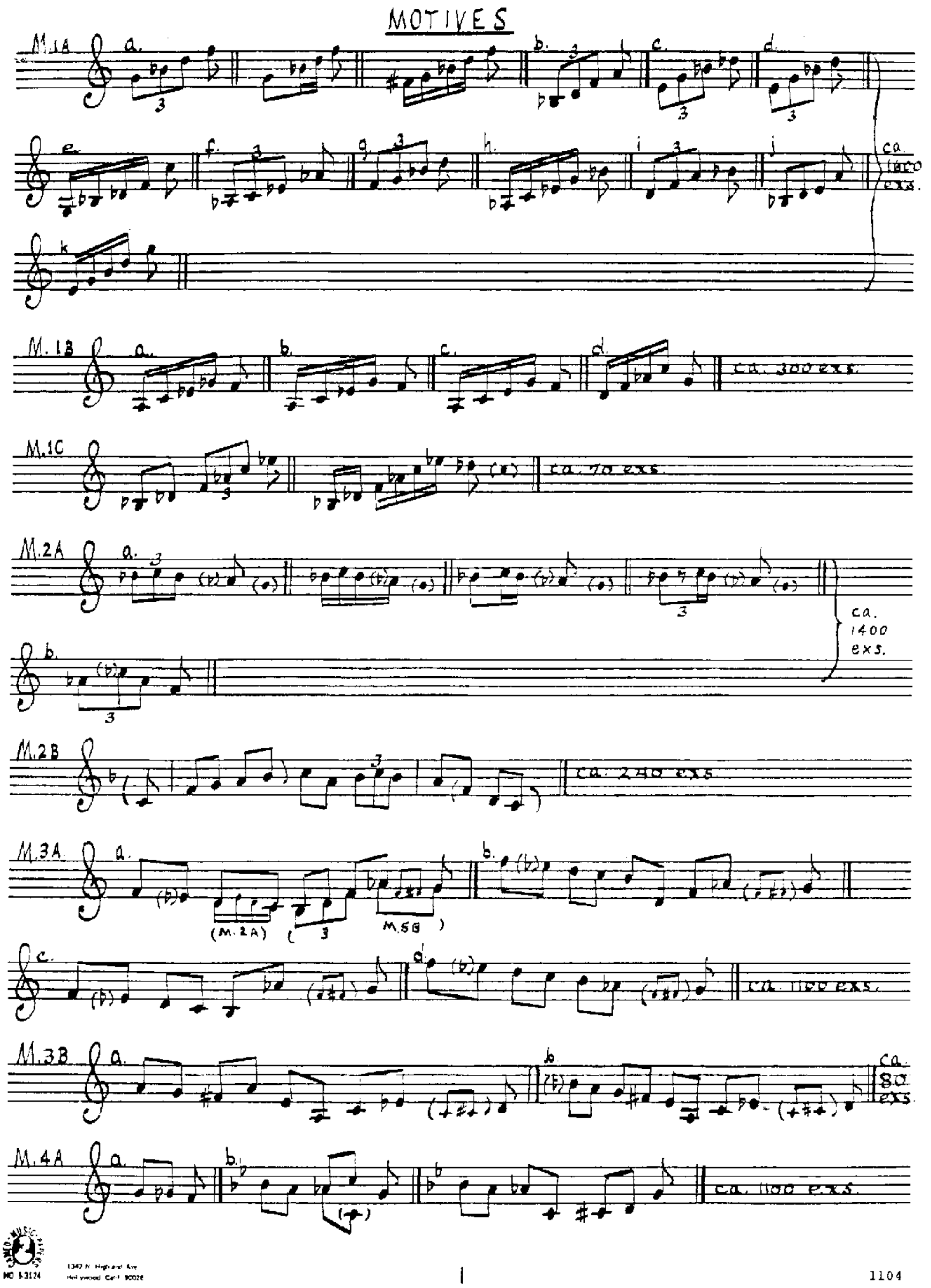



Figure 7-1. Owens 1974, Vol. II, p. 1

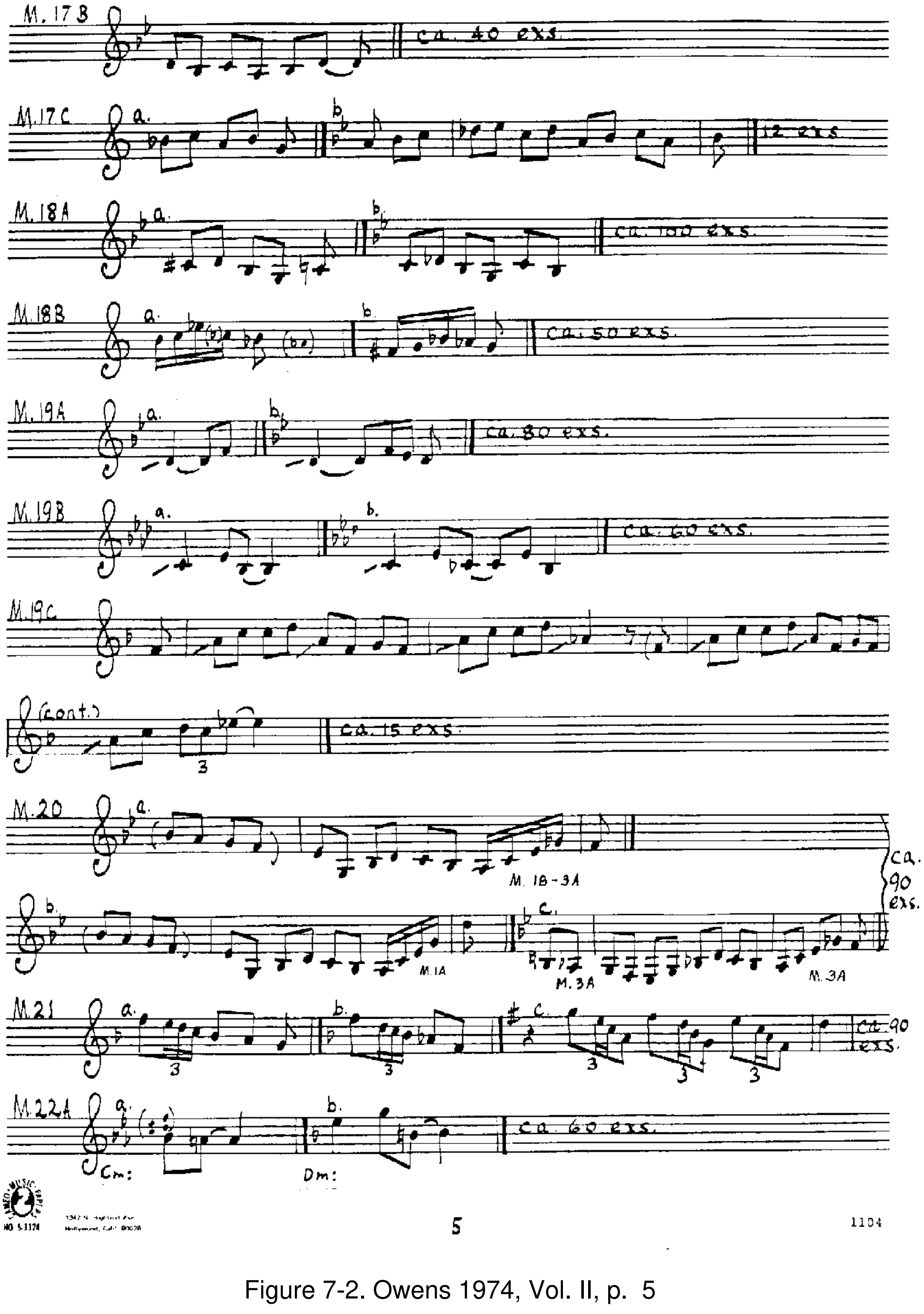



Figure 7-2. Owens 1974, Vol. II, p. 5

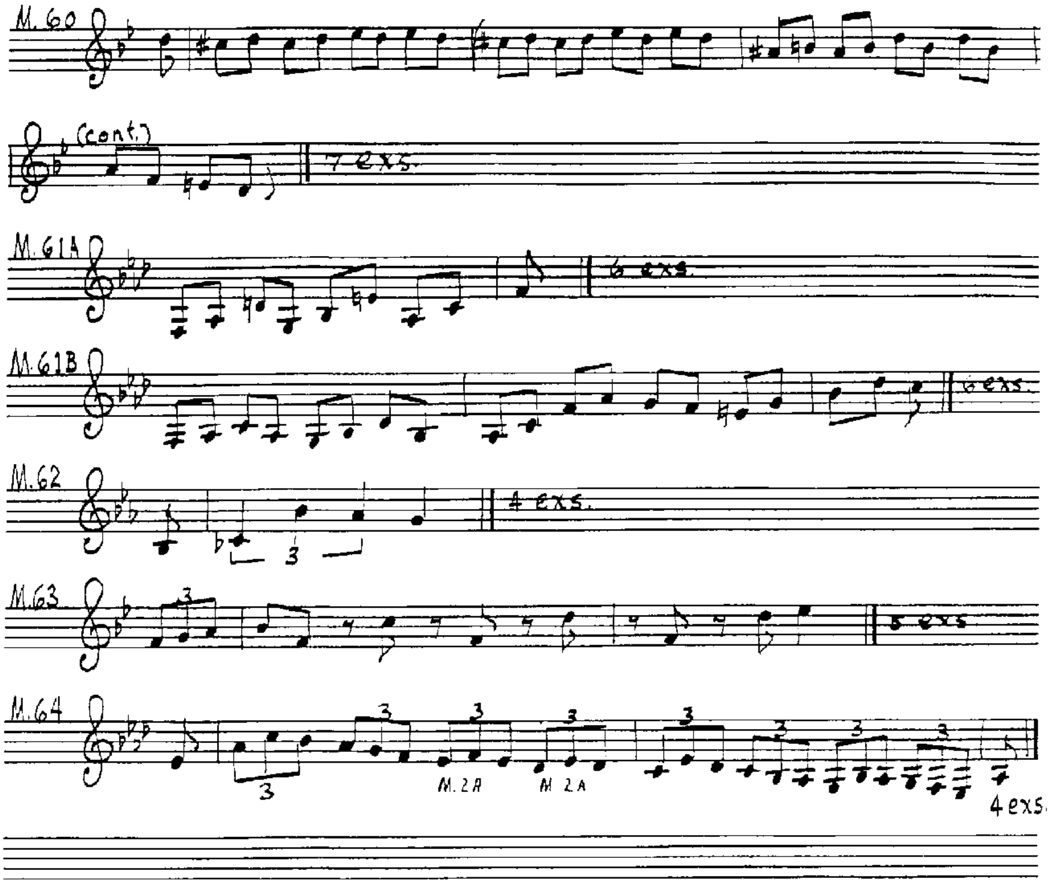



Figure 7-3. Owens 1974, Vol. II, p. 10

ANNEXE B. TRANSCRIPTIONS DE SOLOS DE C. PARKER

Les transcriptions, faites par T. Owens, sont reproduites ici avec la permission de l'auteur sous forme manuscrite, telles qu'elles apparaîssent dans le volume II de sa thèse [Owens 1974].

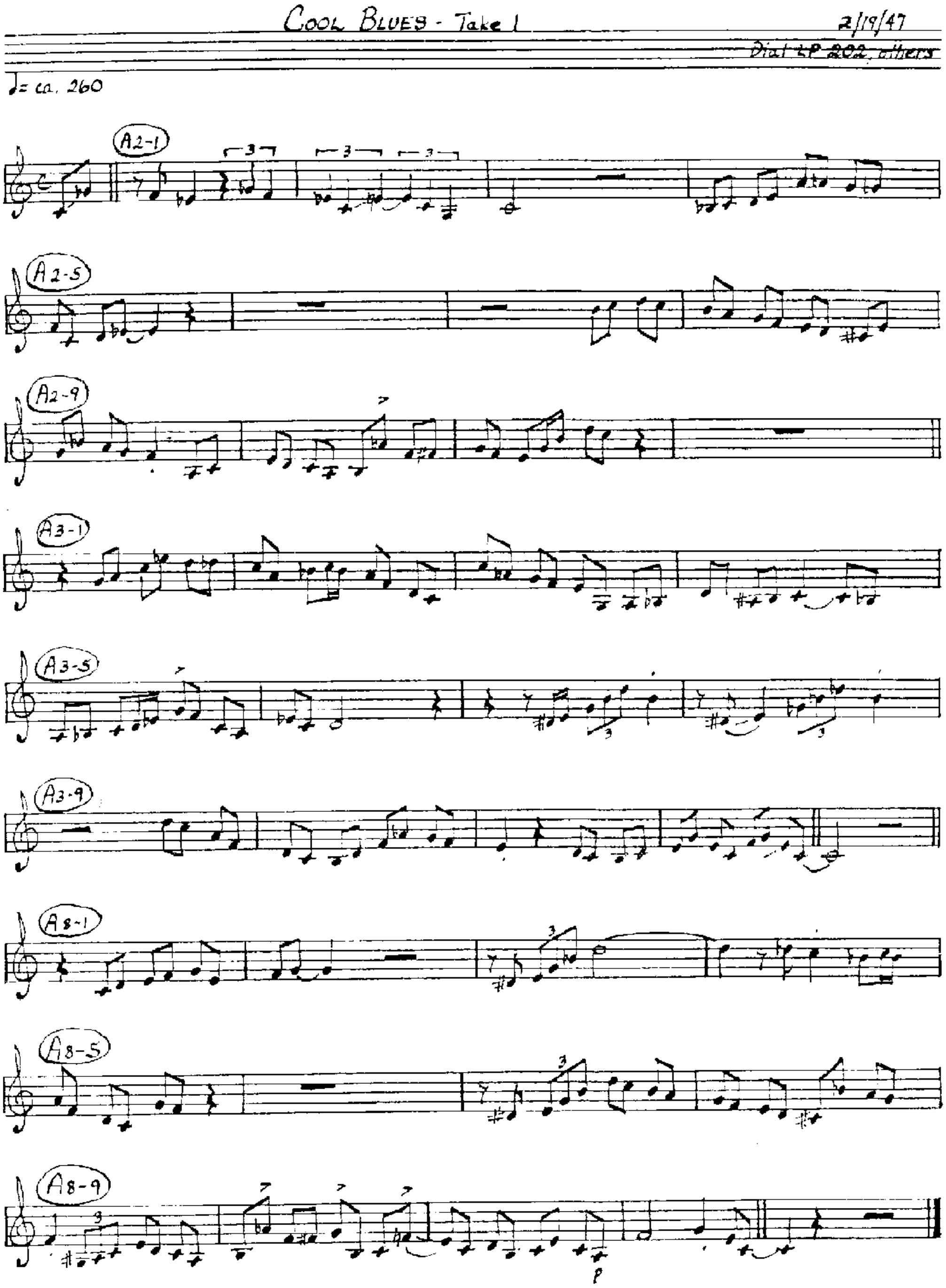



Figure 8-1. [Owens 1974, Vol. I, p. 342]

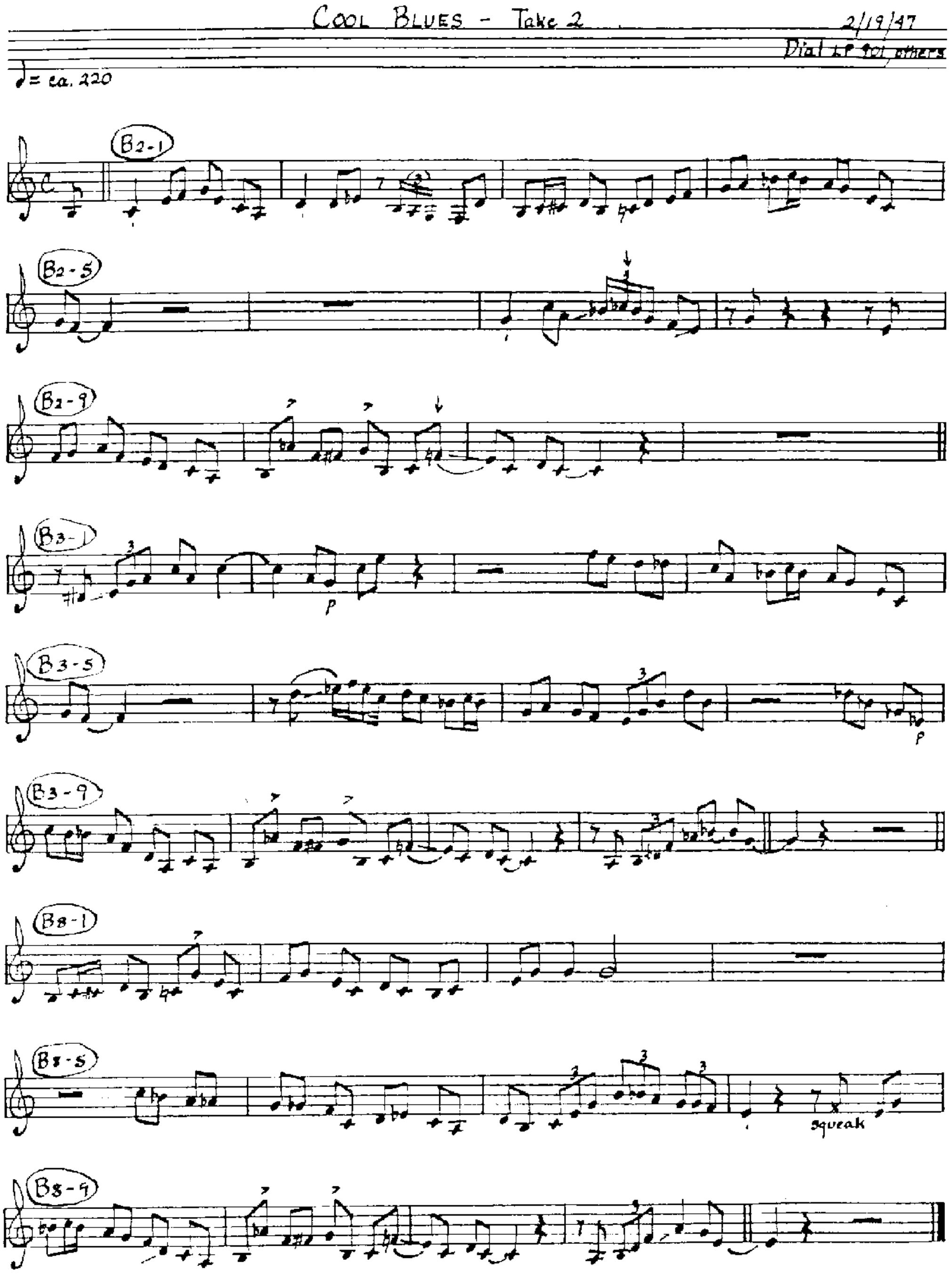



Figure 8-2. [Owens 1974, Vol. I, p. 343]

TABLE DES FIGURES

TABLE DES MATIÈRES